Synthese Library 375
Studies in Epistemology, Logic, Methodology,
and Philosophy of Science

Vincent C. Müller *Editor*

Computing and Philosophy

Selected Papers from IACAP 2014

Springer

Synthese Library

Studies in Epistemology, Logic, Methodology, and Philosophy of Science

Volume 375

More information about this series at http://www.springer.com/series/6607

Vincent C. Müller
Editor

Computing and Philosophy

Selected Papers from IACAP 2014

Springer

Editor
Vincent C. Müller
Anatolia College/ACT
Thessaloniki, Greece
http://orcid.org/0000-0002-4144-4957
http://www.sophia.de

Synthese Library
ISBN 978-3-319-23290-4 ISBN 978-3-319-23291-1 (eBook)
DOI 10.1007/978-3-319-23291-1

Library of Congress Control Number: 2015959017

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Editorial

IACAP 2014

The conferences on ‘Computing and Philosophy’ (CAP) have a long tradition of 28 years, and they are now organised annually by the International Association for Computing and Philosophy (IACAP, http://www.iacap.org/), alternating between Europe and North America. The meeting took place in Thessaloniki, July 2–4, 2014, at the suggestion of the IACAP leadership, in particular of Mariarosaria Taddeo (president) and Marcello Guarini (executive director). The academic and organisational responsibility was given to this editor, who was supported by our team in Thessaloniki, especially Theo Gantinas.

Details of the meeting, including a programme, videos and slides of the invited papers, some photos and a list of participants, are available on the site of the conference, which will remain on http://www.pt-ai.org/iacap. There was general agreement that the conference ran smoothly and showed significantly higher academic level than in some previous years.

Review Process

We sent out a call for papers saying, ‘Computing technologies both raise philosophical questions and shed light on traditional philosophical problems; it is this two-way relation that is the focus of IACAP meetings since 1986’. In total we had 78 submissions by the deadline – which was note extended. Of these 34 (43 %) were accepted for the main track and 5 for the ‘young researchers’ track. We also accepted 14 papers as poster presentations.

We are very grateful to the 42 members of our programme committee who did all the hard reviewing work, double blind. Together with the authors, they are to be thanked for the academic quality of the meeting:

Akman, Varol – Bilkent University
Beavers, Anthony – The University of Evansville

Our invited speakers were Judith Simon (ITU Kopenhagen), ‘The challenge of the computational: towards a socio-technical epistemology’; Hector Zenil (Karolinska Institute, Stockholm), ‘Information and Computation in Synthetic Biology’; Gregory Chaitin (HCTE/Federal University of Rio de Janeiro), ‘Conceptual Complexity and Algorithmic Information’; Selmer Bringsjord (Rensselaer Polytechnic Institute, Troy, NY), Covey Award Winner 2014, ‘Two Refutations of Hegemonic Bayesianism’; and Gualterio Piccinini (University of Missouri-St. Louis), Simon Award Winner 2014, ‘Computation and the Metaphysics of Mind’.

Apart from papers, we also called for symposia and accepted five to be run at IACAP – the organisers of these symposia were responsible for the presentations there, and the symposium papers are not published here in the proceedings. We are very grateful that some high-level symposia were run, namely:

- “Anti-reductionist computational metaphors in evolution, metamathematics and the contemporary human self-image” (organiser: Gordana Dodig-Crnkovic)
- “Robotics: From Science Fiction to Ethical and Legal Issues” (organisers: Sabine Thuermel, Fiorella Battaglia, Barbara Henry)
- “History and philosophy of computing” (organisers: Giuseppe Primiero and Liesbeth De Mol)
- “SuchThatCast” (organiser: Johnny Søraker)
- “Lightning Rounds” (organiser: Don Berkich)

After the conference, the authors of accepted papers were invited to submit full papers. We then ran a second round of online reviews between authors, non-blind this time, which resulted in fruitful and serious exchanges. In the light of these exchanges and comments from the editor, all full papers were revised, mostly several times, and significantly improved – or so we like to think. The revision process ended early November 2014, when each paper had been reviewed at least four times and checked by the editor. As a result of the reviews of the full papers, two submissions were withdrawn and three rejected, resulting in a total of 29 papers. The next review round was from an anonymous reviewer for Springer, who recommended further significant cuts, so after some negotiation we whittled this down to 18 papers, in the end (23 % of the 78 original submissions).

Of course, it is somewhat artificial to sort the conference papers into categories, but a few areas of research can be discerned that form the chapters of this volume: (1) philosophy of computing, (2) philosophy of computer science and discovery, (3) philosophy of cognition and intelligence and (4) computing and society. To this editor, it looks like IACAP is relocating itself, now that ‘computing’ has become a nearly transparent technology and I tend to think that the reflection on the philosophical basics of computing and computer science (Chaps. 1 and 2) can make a useful core for the meetings, while cognition and ‘intelligence’ are quite separate concerns that call for different methods. This core can perhaps be supplemented by societal and ethical concerns (Chap. 4) – even though these tend to be oversubscribed already, with specialist conferences and associations. We tried to make IACAP a high-level specialist meeting in 2014 and offer you the fruits of this hard work. Where the association will go from here, the future will tell.

Struggle and learn!

Anatolia College/ACT, Pylaia-Thessaloniki, Greece Vincent C. Müller

Contents

Part I Philosophy of Computing

Part II Philosophy of Computer Science & Discovery

Part I
Philosophy of Computing

Chapter 1
What Is a Computational Constraint?

Cem Bozşahin

Abstract The paper argues that a computational constraint is one that appeals to control of computational resources in a computationalist explanation. Such constraints may arise in a theory and in its models. Instrumental use of the same concept is trivial because the constraining behavior of any function eventually reduces to its computation. Computationalism is not instrumentalism. Born-again computationalism, which is an ardent form of pancomputationalism, may need some soul searching about whether a genuinely computational explanation is necessary or needed in every domain, because the resources in a computationalist explanation are limited.

Computational resources are the potential targets of computational constraints. They are representability, time, space, and, possibly, randomness, assuming that '**BPP** = **BQP**?' question remains open. The first three are epitomized by the Turing machine, and manifest themselves for example in complexity theories. Randomness may be a genuine resource in quantum computing.

From this perspective, some purported computational constraints may be instrumental, and some supposedly noncomputational or cognitivist constraints may be computational. Examples for both cases are provided. If pancomputationalism has instrumentalism in mind, then it may be a truism, therefore not very interesting, but born-again cannot be computationalism as conceived here.

1.1 Introduction

If I am asked to explain starvation in Korea or Hume's billiard balls, I would not say they are because of computation, and, I suspect, neither would an expert. But why not? After all, all the participants in either story are physical entities or their properties, and any physical property is computational if Deutsch (1985, 1989) thesis is right (but read on).

C. Bozşahin (✉)
Cognitive Science Department, The Informatics Institute, Middle East Technical University (ODTÜ), Ankara, Turkey
e-mail: bozsahin@metu.edu.tr

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_1

Deutsch reformulates Church-Turing thesis using quantum computers, and the examples in the beginning only relate to the revised version of the thesis, that all finitely realizable things are finitely computable, and universal computation is physically realizable, sometimes misleadingly called Church-Turing-Deutsch thesis. Turing and Church talked about effective computation only. Turing did mention physical realizability of effective procedures, but that is far from saying anything physical is computable, as noted by Copeland (2008). (There is further controversy here, one that involves whether Turing addressed humanly effective computations—Copeland interpretation—or effective computations including machines. Hodges (2013) takes the more general view, with which I agree. Otherwise, why would Turing extend his formal model to his imitation game?)

If anything, Turing computability entails physical realizability, but physical realizability does not entail Turing computability. If we have a perfect circle, we have the number π. Unit right triangle gives us $\sqrt{2}$. But there is no effective calculation of $\sqrt[\pi]{2}$. We can question whether there is such a thing as perfect circle or unit right triangle, or whether we should turn to computing over reals and complex numbers. As Cockshott et al. (2012), Deutsch (2011) argue, the question boils down to impossibility of perfect error correction in nondigital computing.

In the world of explanations, for example if DNAs really compute, failure of error correction leads to cancer, and the quest for cure of cancer becomes a search for perfect error correction. The hope very much depends on whether DNAs compute analog way—futile case—or digital, for no measurement is possible without error. If *we* formulate the behavior of DNA as a computational problem, by encoding/decoding hence by having a representation, it is a different enterprise, and some exploits of inherent parallelism is expected in an instrumental way, and, who knows, we might explain cancer if we understand its constraints.

The logic of the argument in the current paper is as follows: we need to ascertain whether we are looking at computation in a process. The intuitions of many computer scientists and physicists about this aspect are nicely put in a framework by Horsman et al. (2013). Once we have that, we need to find out whether a computational explanation is worth considering. If it is, then the degrees of freedom in the data must be formulable in computationalist terms, i.e. its computational constraints must be discernible from functional, cognitive, and other kinds of constraints. This way we can distinguish fundamental constraints, the ones that do the explaining, from auxiliary ones.

Therefore it is important to realize just what is mechanical in computing, and what are its resources. I will cover these aspects first, then define the resources of a computational system as possible targets of constraints, both in a theory and in a model. We then look at purportedly computational constraints, and to noncomputationalist explanations that might benefit from having another look at their domain from this perspective.

1.2 Turing-Deutsch Resources

For reference throughout the paper, let us define a Turing Machine M as TM $M{=}(Q, \Sigma{=}\{0,1\}, \Gamma{=}\{0,1\} \cup \{B\}, \delta, s, h)$, where Q is a finite set of states, $s \in Q$ is the start state, $h \in Q$ is the halt state, δ is the transition function of a single-tape standard TM (i.e. infinite on one side, just like natural numbers, with a designated start cell), 0s and 1s are input (Σ) and output (Γ) encodings, and B is blank. The starting configuration of a TM for input ax is $(s, \hat{a}x)$, and the halting configuration of a TM is (h, y), where $x \in \Sigma^*, y \in (\Sigma \cup \Gamma)^*$, and $\hat{a}$ designates current tape head looking at symbol a. Using 0s and 1s for any kind of computation is no loss of generality because any string can be converted to its unique Gödel number, and that number has a unique bit string representation with an inverse (i.e. input and output can be translated back to the source vocabulary).

M computes a function $f(x)$ iff $M(x){=}f(x)$ for all x, by which, using eta equivalence of lambda calculus, $M{=}f$. (Read 'computes' as 'decides' if f is a decision procedure.) The function f as a decision process is semi-decidable if there is a TM $M(x)$ for it (it can be represented), which only halts if $M(x)$ is in the language of f ending in h, in which case we interpret h as "yes" state. The Halting Problem is semi-decidable; we can write a TM for it but we cannot decide with it. Some problems are such that there is no TM representation for them, for example '$f{=}\pi$?' is uncomputable. There is no TM for it with full precision, therefore we cannot even write purported decision TMs such as 'is x the same as π?' let alone semi-decide with them. Therefore *uncomputable* means two things: either we have no way to represent the process as a TM, or, if we do, it is not decidable, but may be semi-decidable. This exhausts the set of undecidable problems.

It is clear that uncomputable problems are not resource-sensitive, but computable ones show gradient use of resources, as reflected in complexity theory.[1] The standard Turing Machine has been extended in many ways, from multiple tapes, two-way infinite tapes, multiple tape heads, to random-access memory. So far no one has managed to extend them by having infinite states in finite computations, but research on related area is ongoing in the form of *hybrid automata*, where a finite-state discrete system is augmented with an analog system of continuous variables, as a proxy for infinity. The impossibility of perfect error correction, like in all analog components, makes them suspect as universal models of computation (unreprogrammability). In the current paper, I suggest that there is another philosophical problem with this route: the auxiliary component would make the whole

[1] Interaction of computational complexity, which looks at resource *behavior* of functions both at the level of theory and the model, and algorithmic information theory, which looks at resource usage at program sizes, not behaviors, in the form of Chaitin-Kolmogorov-Solomonoff complexity, is an open research area; see e.g. Chaitin (1987). There is also the notion of logical resources, i.e. number of variables and quantifiers, and size of a formula, which leads to descriptive complexity. I leave the latter complexity measures to another paper.

system instrumental rather than explanatory. Constraints on it can be of arbitrary nature. Computational constraints, I suggest, are not like that.

One example of transcending the TM is Abramson's (1971) Extended Turing Machine (ETM) that can store a real number in a single cell. We might try to hypothesize that the tape can hold holistic units such as perfect circle (for π), or right unit triangle, to compute $\sqrt{2}$. Abramson machine has two tapes, one for reals, and one for discrete elements. Members of first tape are fetched somehow. But 'how' question is difficult to answer. (If we could be sure of being able to represent the input in full precision in an enumerable way, we would be ensured of finite computation by ETM if the function is calculable as a countable polynomial.) Blum et al. (1989) machines turn the table around by not feeding the real input bit by bit but as a mathematical object. Finite computation is ensured not by step count but by imposing a built-in constant on the result. If certain operations such as finding the greatest integer in a real can be done in constant time, such machines can be more powerful than TMs. If this is realizable, then we have a nontrivial extension of TMs, and its resources would be of great interest.[2]

We can take the Abramson route in such problems, and try to guarantee finite computation by measuring the distance from a desired output of finite precision. (He proves that this is the only way an ETM can tell whether two analytic functions are equal.) This is also tantamount to using the computer as an instrument rather than an explanatory mechanism. Or, to stay within bounds of a computational explanation, we could say that the question relates to difference between complexity, which presumes decidability, and computability, which does not. Then we can question the need for representation for certain parts of the problem, and we can entertain the possibility of having an instrumental mechanism working in tandem with a computational one, precisely because that mechanism is not computational although it may be physically realizable. Language would not be such a problem, but vision might.

Let us consider now the most natural extension of TMs besides multiple tapes and heads: probabilistic TM (PTM). We know that a PTM can be reduced in terms to a nondeterministic TM, which in turn can be reduced to a standard TM using multiple tapes. (These tapes make it convenient to find out that every nondeterministic computation of say $f(x)$ ends with the same value in $M(x)$, i.e. it is a function.) PTM's complexity class **BPP** has the property **BPP** $\subseteq$ **BQP**, but

[2] In this sense, the notion of representation is not vague in a computationalist explanation, as it is sometimes claimed in dynamical approaches to cognition, e.g. Beer (2003). It is something that makes other resources of a TM accessible. Its ontology in a model is a related but different practice. Reference by an internal representation can be wrong at times, as in early child language acquisition, but representation itself is robust. Of course, not all dynamicists would agree on uselessness of a representation, but complexity results from dynamical systems, such as the one currently discussed in Blum et al. (1989), must already alert the dynamicist and embodied cognitivist that computation in cognition was never parodied as rule-following only; it's about management of computational resources; see Mirolli (2012), Valiant (2013) and Aaronson (2013b) for some discussion.

the reverse containment is not known, therefore randomness might play different roles in PTMs and Quantum Turing Machines (QTMs). It might be a resource in a QTM, whereas it can be translated to other resources in a TM. These are complexity classes, and they do not add new computable functions to our repertoire. From a resource perspective, however, they might make a difference by adding randomness such as in quantum computing.

1.3 Trans-Turing Resources

Consider another extension in this regard, the Putnam-Gold (1965) machine, PGTM. Such machines record their output on a separate tape, whenever they can. (It is easier to think of them as a 3-tape TM, one holding the input TM, the other the input to input TM, and the last one the output.) For example, to attempt the Halting Problem of TM M, a PGTM writes ‘no’ on the output tape before it executes M, and changes it to ‘yes’ if M halts. If it does not, we still have an answer for the object machine M, but the PGTM might run forever. This manouvre appears to add more problems to the class of computable functions, but it only delays the infinite regress by one more step: the halting problem of PGTM is not formulable. If the current answer is ‘no’, it only means it has not halted yet, which is not an answer to the halting problem of PGTM. This way of thinking is not going to add more resources to computing, as the reduction of multiple tape TMs to standard TMs already implicated.

Looking at the output tape without worrying about the status of the object machine on the other tape is same as having a finite but indefinite precision. Assume that we ask the question ‘is $x=\pi$?’ to a PGTM. The answer can be ‘yes’, which means the indices are the same *so far*. ‘No’ as a tentative answer might mean the indices examined are not the same, and we are done. As a *final* answer it would mean PGTM knows the value of π, which is not possible. Uncomputable problems in kind remain uncomputable in PGTM.

Appeals to infinite resources to transcend Turing computability are not facing merely a technological problem; they seem to lack a physical substrate on which to execute an algorithm. Take for example hypercomputation by speed doubling at every step (or halving the amount it takes to reach the next step; Copeland 2002). We either exceed the speed of light in some problems, such as computing $x \overset{?}{=} \sqrt{2}$, or require infinite energy to keep it going (Cockshott et al. 2012). Special Theory of Relativity and thermodynamics have shown that these are not possible outcomes of physical events. The notional extension has no realizability.

There are also examples where rethinking purportedly computational constraints as something else, and vice versa, might lead to a new understanding of the phenomenon under study. In the end the question arises about the choice of automata being the right level to call a constraint computational.

Membrane computing is one such area where the primitives of the executive substrate, biology, seems so remote compared to a TM. (This did not stop Turing from inaugurating morphogenesis.) Such systems perform synchronous transitions from states to states, therefore they have configurations, i.e. they are computing mechanisms. Păun and Rozenberg (2002) show that halting configurations of membranes exist, and the system has complexity classes. Inherent parallelism is kept under control by permeability, which is control of trading time for space to solve exponential-time problems in polynomial time but in exponential space. Unsurprisingly, it can be translated in terms to space (hence resource) control. This can be explained by Turing computation's properties $\mathbf{NP} \subseteq \mathbf{EXP} \subseteq \mathbf{EXPSPACE}$.

In summary, the Turing machine seems to rise above normativity of definitions; it seems to manifest a natural boundary for computability. Even Gödel, who was deeply suspicious of taking formal definitions too seriously, has acknowledged this fact: "with this concept [TM] one has for the first time succeeded in giving an absolute definition of an interesting epistemological notion, i.e. one not depending on the formalism chosen." (Gödel 1946).[3]

This result, and Gödel's point about the epistemological aspects, bring computational explanation as a genuine device in science making. A natural consequence is to look at what these explanations can bring in as a resource, to posit a computational constraint on a phenomenon as an epitome of this way of thinking.

Ever since Turing, these resources have been thought to be mechanical, and it is important to be explicit about what 'mechanical' means. Human computers of Turing did their work "mindlessly," to the point of full substitutability by another human. His imitation game extends the experiment of same kind of computation to artificial computers, and it is clear that the whole process depends crucially on one thing: having a representation. Without representation, there is no need to encode a process in the beginning, to start the physical substrate to compute, and no need to decode at the end, to report it back. These representations live in their own abstract world, in the theory and in the models predicted by the theory. For example the notion of making more effort in a computation translates to taking more steps in a Turing model, with measures in abstract time and space, as we do in complexity classes $\mathbf{P}, \mathbf{NP}, \mathbf{SPACE}$ and others.

Trans-Turing alternatives to computation carry the implication that physical processes might compute, but not necessarily the Turing way. We expect some measurement in such machines, with error control, but that also applies to standard TM, as finite precision of representation. Measurement error cannot be zero. In digital computing it is *made* zero after every sampling, and errors never accumulate. It leads to robust representability. Therefore error correction is not an engineering kludge, it is the fundamental principle of digital processes.[4]

[3] Also note that this comment comes from a man who had considered the mind connection of Turing unthinkable, assuming that Turing flawed by considering possible states of the mind to continue to be finite when the mind changes (Wang 1974: 325, quoting Gödel). $\mathbf{P} \neq \mathbf{NP}$ conjecture brought another perspective to the mind problem; see Aaronson (2013a,b).

[4] As an example from cognitive science of language, consider the word *sleep*. In the spectrum of sleep-like thoughts and behaviors, one ceases to use the word 'sleep' if it strays too far from someone's understanding, and from desires of communication. This is self-imposed error

Once represented, the number π can be calculated to any index by a standard TM, but we still cannot entertain questions such as 'is π same as what is measured on the nondiscrete tape?' The notional extensions must show they are physically (but not necessarily technologically) realizable somehow, when they claim to compute trans-Turing functions. This is because a computational constraint, whatever it is, can be imposed *during* the execution of the function, i.e. in a model, not in the abstract formalism, but also *before* the execution of the function, in the abstract theory, which then leads to models which are by definition equipped with the constraint. For example choosing finite-automata in theory affords only certain kinds of computation in its models.

1.4 Computationalist Explanation

Ascribing computational powers to everything has been called pancomputationalism (Floridi 2004; Piccinini 2003, 2007, and early Putnam 1967; see Müller 2009). For what I have in mind, 'born-again computationalism' is probably a better term, which is the doctrine which says belief in computation brings out an explanation for everything. Not quite, as the previous argument shows (see also Horsman et al. 2013).[5] Pancomputationalism may be a truism if Deutsch is right, but born-again computationalism cannot be. It also follows that avoiding a computational explanation when there is one can be equally blindfolding. That relates to the nature of computational constraints, the main subject of this paper.

Apparent degrees of freedom in so-called new kinds of computation seem to proliferate, which makes it harder to see the following: if we stop at some level which is considered computational in some conception, say computations of

correction, which avoids uncountably many sleep-related words. In that sense language is a digital process, i.e. it is discrete and robustly representational (Haugeland 1997), and words are its proofs.

[5] A cogent warning about why pancomputationalism may be a truism is given by Horsman et al. (2013: 20): "A common, and unfortunate, method of ascribing computational ability to a nonstandard system is as follows. A novel computing substrate is proposed (a stone, a soap bubble, a large interacting condensed-matter system, etc.). The physical substrate is set going and an evolution occurs. At a certain point, the end of the process is declared and measurements taken. The initial and final states of the system are compared, and then a computation and a representation picked such that if the initial state, and final states are represented in such a way then such a computation would abstractly connect them. The system is then declared to have performed such a computation: the stone has evaluated the gravitational constant, the soap bubble has solved a complex optimization problem and so on.

Such arguments, without any further testing or evidence, are rightly treated with suspicion. Given the vast range of representation for physical systems available, almost any computation can be made to fit the difference of initial and final physical states of a system. If such arguments really were correct, we would not only have to conclude that everything in the universe computes, but that everything computes every possible computation all of the time. Such extreme pancomputationalism is even less useful than the usual kind."

individuals in social computing, are we looking at a computational explanation? In cases such as Simon (1996) where power laws and resource boundedness of individuals explain what is socioeconomically organizable, we probably do. I know of no other way of narrowing down the degrees of freedom except to appeal to theoretical limits of what is being claimed and its physical possibility, and to their differences from the basic definition we accept in computer science, namely the Turing model of computation[6]:

> In general, a demonstration that a new system is more powerful than a Church-Turing system involves showing that while all terms of some Church-Turing system can be reduced to terms of the new system, there are terms of the new system that cannot be reduced to terms of that Church-Turing system; in other words, the new system has greater expressive power than that Church-Turing system and hence of any Church-Turing system. Incidentally, it would be truly astonishing if a new system were demonstrated whose terms could not be reduced to those of a Church-Turing system; that is, the new system and Church-Turing systems had incomparable expressive power. Cockshott et al. (2012: 176)

The next section enumerates the terms of a Turing system.

1.5 Computational Constraint

We can distinguish computation as an auxiliary mechanism in an explanation, and computation as the explanatory mechanism. In the first case we need not talk about computational constraints because lifting or imposing the constraint would not make a difference in the explanation. It might, however, constrain an auxiliary function. Conceived that way, any function f that we can execute in a universal TM acts as a constraint: if the problem requires f to be computed, then computing some $g \neq f$ would not serve the purpose. We must have f. This is instrumental use of a computer, i.e. experimentation by programming, and f as a constraint would not sieve out impossible outcomes of the theory.

As an explanatory mechanism, a computational constraint can appeal to resources of computation. As discussed before, these resources can reveal themselves in the theory, in which case the constraint would be embedded in every model arising from the theory, to the extent that, looking from the model's side, it might be difficult to see the constraint. One such example is finite-state machines, which all carry the TM constraint that the machine makes no turns in the tape. We can execute finer resource control in theory than Chomsky hierarchy, by for example employing

[6] Physical possibility is crucial because every computation requires an executive medium. This is one difference between functionalism and computationalism, where in the latter the primitives of the executive substrate may shape the form of the computed function. For example, membrane computing's *in* command (to permeate a local membrane) has a Turing correlate (Păun 2000), but we would expect membrane models to be programmed more naturally with *in*. Making a difference in complexity classes makes a difference in computationalism, but not necessarily in functionalism.

one-turn PDAs to get linear languages, and finite-turn PDAs to move to ultralinear languages (Harrison 1978), before we reach context-freeness. Once realized in a model, these constraints are no longer explicit but their effects are built in.[7]

Other computational resources are revealed on the model side: time, space, and randomness (because whether **BPP** = **BQP** is not known). The first two are exploited in complexity theory. The last one is not a resource per se in classical computing, as we have seen. It might be a resource in quantum computing and oracles, because if we are satisfied with a truly random answer, then such machines can provide one without effort. This is not the role of randomness in quantum computing, because if it were, we would have to take a wrong answer which might be spit out randomly as the result of a quantum computation. QTMs do compute functions; they can be programmed in principle. Randomness has unsettled consequences in computer science; for example it is tempting to think of **P** to be a proper subset of **BPP**, but it may in fact be the case that **P** = **BPP** (see Arora and Barak 2009 for discussion).

If a machine is not required to give a causal explanation of how the answer was constructed, it can reduce the amount of work to nil in principle. This is what oracles do. Quantum machines, however, must be able to provide an explanation because of superposition and unitary transformations (reversible computing).[8] By playing with randomness that we can tolerate, we can control the amount of work a computational system is expected to do. That makes randomness a resource, and a potential target for a computational constraint. This is true of standard TM computation too, but not as much as the quantum variety: interactive proofs use randomness of the verifier as a resource in finite number of interactions between the prover and the verifier (the prover can be probabilistic too, without change of results). The class **IP** which characterizes such proofs is in **PSPACE**, i.e. we do not need more than efficiently space-bounded computations to deal with such kind of randomness, hence its constraints may have already been spoken for in terms of space.

1.6 Computational Constraint Versus Instrumental and Cognitivist Constraints

Ecorithm is a term Valiant (2013) uses for nature's algorithms for learning and prospering in complex situations, and it might at first sight appear to suggest some form of born-again computationalism. This would be an incorrect interpretation.

[7] Notice that turn itself is not a resource. It translates to bounded use of space, which is a resource.

[8] Roughly speaking, here is how we can get the answer algorithmically. Assume that we expect QTM M to compute $M(x)=f(x)$, given x. Input is prepared in a superposition by for example a series of Hadamard transformations, from $|0\rangle$ to $|x0\cdots0\rangle$. M computes and ends in a state $|xf(x)0\cdots0r\rangle$, where r is the residue of quantum gates. Copy—not clone—the result to unused bits to get $|xf(x)f(x)r\rangle$ by a series of quantum operations $|b_1b_2\rangle \mapsto |b_1(b_1 \oplus b_2)\rangle$. Then run the gates in reverse to get $|x0\cdots0f(x)0\cdots0\rangle$, and Hadamard to get $|0\cdots0f(x)0\cdots0\rangle$, viz. $|f(x)\rangle$.

These algorithms are not computational because they are natural, they are natural because they are resource-sensitive in a computational way, in this particular case by complexity measures and sampling. These constraints are explanatory, and not pancomputationalist. They do assume that the problem becomes formulable in computationalist terms if there is an encoding/decoding mechanism of its instances, much like in the sense described by Horsman et al. (2013).

Some cognitive constraints may turn out to be computational as well. (What I am suggesting is that, a computationalist reformulation that can posit computational constraints only, to stay at this level of explanation, may be an alternative explanation to cognitivism. In the instrumental sense, any constraint eventually becomes a computer-implemented function, as argued earlier.)

Consider one example. Gentner (1982) claimed that children acquire nouns first universally, because nouns are names for things, and there are things around when a child grows. Framed as such, this is a conceptual (cognitive) constraint or bias, toward the objects and their perception, called Natural Partition Hypothesis. A computationalist alternative is to suggest that maybe short, unambiguous and frequent things are learned first, whatever they are, because these aspects can be shown to reduce computational effort (see Bozşahin 2012 for more discussion). For example, contra natural partition hypothesis, Mandarin children seem to show no cognitive bias toward nouns (Tardif 1996). It may be because Mandarin words are uniformly short. Tzeltal children use verbs very early (Brown 1998), which would not be surprising from a computationalist perspective because these verbs are argument-specific therefore unambiguous (e.g. Tzeltal has a generic word for 'eat' and a specific word for eating tortillas, and children acquire the specific ones first). Turkish children seem to use parts of speech equally frequently (Avcu 2014). It might be simply because they are equally frequently exposed to them through morphology (e.g. clauses can be nominalized to bear nominal inflection, and nominals can be predicates, requiring verbal inflection). Of course this simplistic scenario cannot be the whole story extending to adult life, but keep in mind that this is just one computational constraint. Expressivity and the need to communicate will steer the child in somewhat contrary directions. (We can be very creative and use personal words all the time. That would be very expressive but highly uncommunicative. We can be very communicative by using no novel word forms at all, but that would not be very expressive, etc.) This too can be shown to be a computational constraint: in artificial simulations of lexicon development, finite convergence to common vocabulary requires control of synonymy, and indirectly, expressivity and communicative success (Eryılmaz and Bozşahin 2012). In particular, full communicative success implies very early convergence, and full expressivity implies no convergence.

Consider another example. Tomasello's gradual transition to a position where chimpanzees are considered to have a relatively simple mind (from the earlier position of having no mind at all), suggests a potential for computational explanation as well (Tomasello and Call 1997; Tomasello et al. 2003). Not surprisingly, he and his colleagues formulated the constraints in terms of cognitive capacities, joint attention, communicative intent and others. But, there seems to be a continuum along the computational resources. Chimpanzees are known to make plans (see Steedman 2002 for a review, and Jaynes 1976 for more examples). But they seem to have difficulty going from “I intentions” to “we intentions,” to use Searle’s (1990) terminology (a reliable source of mind anti-computationalism). We can conceive the instrumental plans of chimpanzees as stack of actions, and collaborative we-intentions of humans as stack of stacks (Bozşahin 2014). We can prove that they make fundamentally different uses of computational resources. We can then go back and try to explain chimpanzee’s awareness of other minds in computationalist terms, perhaps with more experiments to tell computational constraints apart from cognitive ones.

There is an area where we know that computationalist explanations can predict an upper bound on the nature of complexities: dependency structures in language are known to be over and above context-freeness (Shieber 1985). And how much above has a computational answer: linear-indexed dependencies, therefore linear-indexed languages. This computational constraint in theory translates itself to embedded PDA in models (Vijay-Shanker 1987) (a stack of stacks, hence the appeal to computational continuity above), which applies to all the models without the need of an explicit mention. CCG and TAG are two such theories.

In contrast, we can have another look at explanations that seem to use computational terminology but not computational resources. Take for example the so-called functional constraints on language, the kind Hawkins (1994) identifies as some word orders causing “processing difficulties” because they purportedly make it difficult to connect earlier constituents in parsing, which is supposedly a function, to mother nodes in a tree of constituents. Judging from the terminology, it appears to be a computational constraint. But no automata class is singled out by this constraint, or a resource. The theory itself is not constructed in a grammar formalism that spells a certain class of automata either (finite-state, push-down, embedded push-down etc.). We have no grounds to see what kind of computations is left out by the constraint, and we are left with an impression that parsing is a performance function. However, children parse to learn, rather than learn to parse (Fodor 1998), and there are computational constraints on the process. For example, Joshi’s (1990) constraints are computational in this sense. A truly functional constraint would not be computational. Hawkins (1994) appears to posit a functional constraint, and instrumental, rather than computational.

Automatic computations and interactive ones can be compared from this perspective as well. Turing (1936) called the first kind a-machines and the second c-machines (c for ‘choice’). Oracles (o-machines; Turing 1939) differ from both. They need not specify an internal i.e. computational mechanism although they might have one.

Consider the difference between calculating an integral within an interval and playing backgammon. There is no external element in the first one that would force the symbols of its “computations” to communicate with the computer and/or the outside world. For backgammon there is an external element, the dice. The dice throw, however, can be incorporated into a-machine configurations, say by having another tape to enumerate sequence of dice throws long enough to serve any finite game. If this a-machine must decide whether the same configuration reached several times in a game warrants continuation, asking this question puts a computational constraint on the game, unlike asking for a dice throw, because the automata class changes from a-machines to c-machines, with concomitant results such as potential undecidability. As an open system in this sense, it is constrained descriptively by whatever is delivered by the dice, and computationally by an interaction. For integral calculation, if e.g. switching from integers to reals could change the complexity of the calculation, then we would consider the integer/real constraint a computational one. Blum et al. (1989) suggest that rethinking computation with reals reveals new complexity classes such as **NP**-R of decision problems for R, where R=$\mathbb{R}$ or R=$\mathbb{Z}$. **NP** is same as **NP**-$\mathbb{Z}$, but not **NP**-$\mathbb{R}$.

1.7 Conclusion

If a constraint is truly computational, we can see a computationalist explanation behind it. If not, the constraint may be instrumental. If some phenomenon begs for a computationalist look, there will be telltale signs, most notably parsimonious use of a computational resource, i.e. a Turing-reducible resource: representation, time, space, and randomness (assuming **BPP** $\supseteq$ **BQP** is unresolved, and perhaps unlikely to be true; whether **BPP** $\subseteq$ **BQP** containment is proper is not known).

I have given examples of two kinds: purportedly computational constraints which are not translatable to these terms, and supposedly noncomputational or cognitivist constraints, which might. Different vocabulary does not seem to change this aspect (as we observed in membrane computing), but may make it more opaque or indirect to see the potentially computational nature of a problem.

Computational explanations face the same risk, not least of which ranges from reading Church-Turing thesis stronger than necessary, to pancomputationalism and born-again computationalism.

Acknowledgements Thanks to Mark Addis, Tarek Besold, Mark Bishop, Robin Hill, Doukas Kapantaïs, Stasinos Konstantopoulos, Giacomo Lini, Halit Oğuztüzün, Deniz Şener, Mario Verdicchio, Ioannis Votsis, Aziz Zambak for discussion and advice on the paper and related topics. I am to blame for all errors and misunderstanding.

References

Aaronson, S. (2013a). *Quantum computing since Demokritus*. Cambridge: Cambridge University Press.

Aaronson, S. (2013b). Why philosophers should care about computational complexity. In B. J. Copeland, C. J. Posy, & O. Shagrir (Eds.), *Computability: Turing, Gödel, Church, and beyond*. Cambridge: MIT Press.

Abramson, F. G. (1971). Effective computation over the real numbers. In *IEEE 54th Annual Symposium on Foundations of Computer Science*, Los Alamitos (pp. 33–37).

Arora, S., & Barak, B. (2009). *Computational complexity: A modern approach*. Cambridge: Cambridge University Press.

Avcu, E. (2014). *Nouns-first, verbs-first and computationally easier first: A preliminary design to test the order of acquisition*. Unpublished master's thesis, Cognitive Science department, Middle East Technical University (ODTÜ), Ankara.

Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior, 11*(4), 209–243.

Blum, L., Shub, M., & Smale, S. (1989). On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin (New Series) of the American Mathematical Society, 21*(1), 1–46.

Bozşahin, C. (2012). *Combinatory linguistics*. Berlin/Boston: De Gruyter Mouton.

Bozşahin, C. (2014). Natural recursion doesn't work that way: Automata in planning and syntax. In V. C. Müller (Ed.), *Fundamental issues of artificial intelligence (synthese library)*. Berlin: Springer.

Brown, P. (1998). Children's first verbs in Tzeltal: Evidence for an early verb category. *Linguistics, 36*(4), 713–753.

Chaitin, G. J. (1987). *Algorithmic information theory*. Cambridge/New York: Cambridge University Press.

Cockshott, P., Mackenzie, L. M., & Michaelson, G. (2012). *Computation and its limits*. Oxford/New York: Oxford University Press.

Copeland, B. J. (2002). Hypercomputation. *Minds and Machines, 12*(4), 461–502.

Copeland, B. J. (2008). The Church-Turing thesis. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy*. Stanford: Stanford University.

Deutsch, D. (1985). Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 400*(1818), 97–117.

Deutsch, D. (1989). Quantum computational networks. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 425*(1868), 73–90.

Deutsch, D. (2011). *The beginning of infinity: Explanations that transform the world*. New York: Penguin.

Eryılmaz, K., & Bozşahin, C. (2012). Lexical redundancy, naming game and self-constrained synonymy. In *Proceedings of the 34th Annual Meeting of the Cognitive Science Society*. Sapporo.

Floridi, L. (2004). Open problems in the philosophy of information. *Metaphilosophy, 35*(4), 554–582.

Fodor, J. D. (1998). Parsing to learn. *Journal of Psycholinguistic research, 27*(3), 339–374.

Gentner, D. (1982). Why nouns are learned before verbs: Linguistic relativity versus natural partitioning. In S. A. Kuczaj II (Ed.), *Language development, vol.2: Language, thought and culture* (pp. 301–334). Hillsdale: Lawrence Erlbaum.

Gödel, K. (1946). Remarks before the Princeton bicentennial conference on problems in mathematics. In M. Davis (Ed.), *Undecidable*. New York: Raven Press. (1965, p. 84)

Gold, E. M. (1965). Limiting recursion. *Journal Symbolic Logic, 30*, 28–48.

Harrison, M. (1978). *Introduction to formal language theory*. Reading: Addison-Wesley.

Haugeland, J. (1997). What is mind design? In J. Haugeland (Ed.), *Mind design II* (pp. 1–28). Cambridge: MIT Press.

Hawkins, J. A. (1994). *A performance theory of order and constituency*. Cambridge: Cambridge University Press.

Hodges, A. (2013). Alan Turing. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy*. Stanford: Stanford University.

Horsman, C., Stepney, S., Wagner, R. C., & Kendon, V. (2013). When does a physical system compute? *Proceedings of the Royal Society A, 470*(20140182).

Jaynes, J. (1976). *The origin of consciousness in the breakdown of the bicameral mind*. New York: Houghton Mifflin Harcourt.

Joshi, A. (1990). Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes, 5*, 1–27.

Mirolli, M. (2012). Representations in dynamical embodied agents: Re-analyzing a minimally cognitive model agent. *Cognitive Science, 36*(5), 870–895.

Müller, V. C. (2009). Pancomputationalism: Theory or metaphor? In *The relevance of philosophy for information science*. Berlin: Springer.

Păun, G. (2000). Computing with membranes. *Journal of Computer and System Sciences, 61*(1), 108–143.

Păun, G., & Rozenberg, G. (2002). A guide to membrane computing. *Theoretical Computer Science, 287*(1), 73–100.

Piccinini, G. (2003). *Computations and computers in the sciences of mind and brain*. Unpublished doctoral dissertation, University of Pittsburgh.

Piccinini, G. (2007). Computational modelling vs. computational explanation: Is everything a Turing Machine, and does it matter to the philosophy of mind? *Australasian Journal of Philosophy, 85*(1), 93–115.

Putnam, H. (1965). Trial and error predicates and the solution of a problem of Mostowski. *Journal Symbolic Logic, 30*, 49–57.

Putnam, H. (1967). Psychological predicates. In *Art, mind, and religion* (pp. 37–48). Pittsburgh: University of Pittsburgh Press.

Searle, J. R. (1990). Collective intentions and actions. In M. E. P. Philip, R. Cohen Jerry, & L. Morgan (Ed.), *Intentions in communication*. Cambridge: MIT Press.

Shieber, S. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy, 8*, 333–343.

Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge: MIT press.

Steedman, M. (2002). Plans, affordances, and combinatory grammar. *Linguistics and Philosophy, 25,* 723–753.

Tardif, T. (1996). Nouns are not always learned before verbs: Evidence from Mandarin speakers' early vocabularies. *Developmental Psychology, 32*(3), 497–504.

Tomasello, M., & Call, J.(1997). *Primate cognition*. New York: Oxford University Press.

Tomasello, M., Call, J., & Hare, B. (2003). Chimpanzees understand psychological states—the question is which ones and to what extent. *Trends in Cognitive Sciences, 7*(4), 153–156.

Turing, A. M. (1936). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society, 42*(series 2), 230–265.

Turing, A. M. (1939). Systems of logic based on ordinals. *Proceedings of the London Mathematical Society, 2*(1), 161–228.

Valiant, L. (2013). *Probably approximately correct: Nature's algorithms for learning and prospering in a complex world*. New York: Basic Books.

Vijay-Shanker, K. (1987). *A study of tree adjoining grammars*. Unpublished doctoral dissertation, University of Pennsylvania.

Wang, H. (1974). *From mathematics to philosophy*. London: Routledge & Kegan Paul.

Chapter 2
Computing Mechanisms and Autopoietic Systems

Joe Dewhurst

Abstract This chapter draws an analogy between computing mechanisms and autopoietic systems, focusing on the non-representational status of both kinds of system (computational and autopoietic). It will be argued that the role played by input and output components in a computing mechanism closely resembles the relationship between an autopoietic system and its environment, and in this sense differs from the classical understanding of inputs and outputs. The analogy helps to make sense of why we should think of computing mechanisms as non-representational, and might also facilitate reconciliation between computational and autopoietic/enactive approaches to the study of cognition.

2.1 Introduction

Computational and autopoietic (or enactive[1]) approaches to cognition have traditionally been opposed to one another, primarily due to a disagreement about whether or not representation is necessary for cognition. On the one hand, computation has classically been understood as inherently representational (Sprevak 2010: 260), leading to the conclusion that if cognition is computational then it must also be representational (Fodor 1981: 180). On the other hand, autopoietic theory and the enactive approach that it inspired have both argued that cognition does not require representation (see e.g. Varela et al. 1991: 8; Thompson 2007: 52–3). This has led to a situation in which we have two divergent approaches to the study of cognition that appear to be fundamentally irreconcilable.

[1] Here I have in mind the "biological enactivism" of Varela (e.g. 1991), Thompson (e.g. 2007), and di Paolo (e.g. 2005), as opposed to the "sensori-motor" enactivism of Hurley (e.g. 1998), Noë (e.g. 2004), and Hutto and Myin (2013). See Villalobos and Ward (2014, fn 1) for a discussion of this distinction. All further references to enactivism should be understood as referring to biological enactivism.

J. Dewhurst (✉)
Department of Philosophy, School of Philosophy, Psychology and Language Sciences,
University of Edinburgh, Edinburgh, UK
e-mail: joseph.e.dewhurst@gmail.com

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_2

Recently, however, there have been several attempts to give non-representational accounts of computation (see e.g. Egan 1995; Piccinini 2008), which could in theory lead to reconciliation with autopoietic/enactive approaches. One such attempt is Gualtiero Piccinini's mechanistic account of computation, which characterises computational states as components in a mechanism (see his 2007). According to this account it is not necessary that computational states represent, although it does not rule out this possibility either – rather the question of what it means to compute becomes separated from that of what it means to represent (Piccinini 2004: 404). Similarly, whilst an autopoietic system might be interpreted as representing, this is neither essential to its identity nor constitutive of its operation (Maturana and Varela 1980: 78). Here I draw an analogy between the two kinds of system, based on this point of similarity. My primary aim is to elucidate the nature of inputs and outputs in the mechanistic account, but I also hope that this comparison might facilitate reconciliation between computational and autopoietic/enactive approaches to the study of cognition. Given the current dominance of computationalism in cognitive neuroscience, this would allow for autopoietic and enactive approaches to make a more meaningful contribution to practical research.

The first two sections will introduce computing mechanisms and autopoietic systems, respectively. The third and fourth sections will expand on the analogy between the two, focusing first on the role of representations and then turning to inputs, outputs, and perturbations. Finally there will be a brief discussion of how this analogy might help reconcile the two approaches, and how this might be of benefit to the study of cognition.

2.2 Computing Mechanisms

Classical accounts of computation have tended to invoke representation in order to distinguish computational states from mere physical states (Ramsey 2007: 43), and also to individuate those states (Sprevak 2010: 24–6). This is problematic if you have prior theoretical objections to representation; such as thinking that representation introduces a vicious circularity into computational explanation (Piccinini 2004: 377). It is concerns of this kind that have prompted Piccinini to develop a non-representational account of computation (see his 2008), although it should be noted that Piccinini is not committed to a totally non-representational account of cognition outwith its computational elements.

Piccinini's account is inspired by recent work on mechanistic explanation in cognitive science (2007: 501). Mechanistic explanation focuses on describing a target phenomenon in terms of the structured interaction of physical components (Craver and Bechtel 2006), where these components are understood as fulfilling certain functional roles. It is claimed that mechanistic explanation is especially suited to the special sciences, such as cognitive science and computer science, where the traditional deductive-nomological model might be less relevant, as we are unlikely to discover broadly applicable natural laws (Bechtel 2005: 208; Craver and Darden 2005: 234).

A computing mechanism[2] is defined as a physical system that carries out systematic transformations on strings of digits (this does not rule out its also having some other function, such as controlling a system as a result of these transformations). By 'systematic transformation' I simply mean any physical interaction that will transform strings of digits in a way that would be replicated if the same kind of interaction were to take place under relevantly similar circumstances. This is distinct from the view that computation is simply an implementation of a systematic transformation from input to output. In contrast, the mechanistic account defines computation as the systematic transformation of digits, and does not in fact require an input or output of any kind (see below).

This structure requires a minimum of two components: a string of digits and a processor to transform those digits. It may also include an input device (for transforming external stimuli into strings of digits), an output device (for reversing this process), and a memory component (which may just be a looped string of digits). Whilst many computers will include these additional components, the simplest forms of computation can proceed without them (Piccinini 2007: 514). Each component is individuated functionally, such that every digit of the same type is treated in the same way by the mechanism, and every processor of the same type performs the same transformation on any given string (Piccinini 2007: 508–20). Whilst this processing of digits can be given a representational interpretation, it is not necessary to do so in order to explain how the mechanism functions (see Piccinini 2008).

This can be contrasted with what Sprevak calls "the received view", which is that computation must necessarily invoke representation in order to individuate digits and processors (2010: 260). Typically this is cashed out in one of two ways: representation can be required either for the individuation of computational states and processes (this is Sprevak's preferred interpretation), or it can be required in order to explain the transformation from input to output. In both cases the vehicles are understood as being physical states of some kind (such as holes in a punch-card or variations in voltage level), whilst the content can be either features of the world or abstract entities such as numbers (again, Sprevak prefers the latter, more minimalist interpretation). Under the mechanistic account there is no need for any appeal to representational content for either individuation or causal explanation.

I will not provide any further defense of the mechanistic account at this time, although I hope that the analogy with autopoietic systems will help elucidate precisely why computing mechanisms are not intrinsically representational. For a more detailed defense of the mechanistic account see Dewhurst (2014).

[2] Strictly speaking, a digital computing mechanism, although similar accounts can be given for analog or generic computing mechanisms (see Piccinini and Bahar 2013).

2.3 Autopoietic Systems

Maturana & Varela's theory of autopoiesis was developed in response to a perceived need for a better definition of living systems, which they take to encompass cognitive systems (i.e. all cognitive systems will be living systems, even if not all living systems are cognitive systems). It does this by focusing on the homeostatic nature of living organisms, which seem to be uniquely capable of preserving their structural integrity in response to environmental interference (Maturana and Varela 1980: 78ff). This criterion seems to capture everything that we might instinctively think of as living, and has the added benefit of not automatically excluding non-organic (or even non-physical) systems.

The neologism *autopoiesis*, formed from the Greek *auto* ("self") and *poiesis* ("production") is intended to capture the essence of this criterion. An autopoietic system is one that is focused towards continual self-production, as opposed to an allopoietic system, which produces something other than itself (Maturana and Varela 1980: 80–1). As a natural, deterministic entity an autopoietic system is non-teleological and therefore non-representational. It also exhibits conceptual and physical unity (*ibid*: 96): conceptual unity because self-production grants it an identity independent of any external observer, and physical unity because of its ability to maintain a coherent structure over time (*ibid*: 97). This reinforces its non-teleological status, as its behaviour emerges out of homeostatic regulation rather than being directed towards any external goal. Maturana & Varela acknowledge that we will often describe an autopoietic system in teleological terms, but descriptions of this kind are to be understood as strictly observer-relative, rather than reflecting anything intrinsic to the system itself.[3]

The archetypal example of a physical autopoietic system is a living cell, which whilst being "a thermodynamically open system" is nonetheless a closed unity in the sense that it "produces its own components [...] in an ongoing circular process" (Thompson 2007: 98). The body, as a collection of autopoietic cells, is granted second-order autopoietic status, and cognition is seen as a function of the living organism as a whole (Maturana and Varela 1980: 13). One could even imagine autopoietic systems made up of collections of organisms, although Maturana & Varela disagreed about whether this would be possible, and so did not comment on it (1980: 118). Cognition, for Maturana & Varela, is not intrinsic to an autopoietic system, but is rather an observer-relative categorization of the kind of behaviour that it exhibits.

This is by necessity a very brief outline of Maturana & Varela's theory of autopoiesis, but it should be sufficient for current purposes. For an accessible

[3] Varela later turned away from the idea that autopoietic systems are non-teleological (see Weber and Varela 2002), but this position is maintained in Maturana's later work. I focus here on the formulation of autopoiesis given in Maturana and Varela (1980).

overview see Mingers (1989); for the original formulation see Maturana and Varela (1980). In the following sections I will provide additional clarification when it is required in order to make sense of the analogy with computing mechanisms.

2.4 Representation

Representations are commonly seen as an essential component of cognitive scientific explanation (at least on the classical view), and thus naturalising representation is seen as an essential task for the philosophy of cognitive science (cf. Ramsey 2007). This makes it all the more interesting that both the mechanistic account and autopoietic theory are claimed to be non-representational (Piccinini 2008; Maturana and Varela 1980: 91), whilst at the same time serving as the basis for theories of cognition.

Computing mechanisms are non-representational because their components (digits and processors) can be individuated without specifying representational content. This is done functionally, by describing how a given digit-type will behave when it encounters a given processor-type. For instance, imagine a system with digits of two types (call them 0 and 1), and a certain processor (call it an x-gate). This processor takes pairs of digits and produces a single digit, based on what the pair consists of. If the pair is 0–0, 0–1, or 1–0, it produces a 0, and if the pair is 1–1, it produces a 1. Note that whilst this appears to correspond precisely with the logical function AND, we do not need to know this in order to individuate the digits, meaning that it is unnecessary for us to say that it represents this function. Additionally, our choice to label the first kind of digit 0 and the second 1 was completely arbitrary, and we could just as easily have reversed it, in which case our processor would appear to correspond with the OR function. The physical process is not sufficient to determine what logical function is taking place. All that is required for the computing mechanism to operate correctly is a physical difference between the two kinds of digit that are recognised by the processor (this could be done in a variety of ways, such as voltage levels or the presence/absence of a hole on a tape). Whilst we might choose to attribute representational content to computational states or processes, this attribution is not what makes the system computational, nor is it essential to our understanding of what it is to compute (see Piccinini 2008 for more detail).

Autopoietic systems are non-representational because, as Maturana & Varela put it, “there is no specification in the cell of what it is not” (1980: 91). This is a consequence of what Maturana calls “structural determinism”, which is true of any physical system: the result of any interaction with a physical system is fully determined by the intrinsic structure of that system, meaning that anything external to the system is merely a trigger, rather than a determiner, of that system’s structural dynamics (see Maturana 2003: 61).

Representation is further ruled out by the lack of teleology described earlier – to represent requires being able to misrepresent, which implies some purpose or intention by which to judge failure or success (Millikan 1995: 186). Whilst it "may be metaphorically useful" to attribute representational content to an autopoietic system, this is ultimately "inadequate and misleading" (Maturana and Varela 1980: 99). Representation, just like teleology, should be treated as an epistemic tool for the benefit of an observer, rather than as a genuine aspect of an autopoietic system (*ibid*: 85–6).

2.5 Inputs, Outputs, and Perturbations

Both computing mechanisms and autopoietic systems turn out to be non-representational for much the same reason. This is essentially because of the way that they interact with the external world. Each kind of system is fully specified without any mention of its environment, in the sense that we can give a complete definition of a computing mechanism or autopoietic system that does not mention anything external to the system. This pre-empts the need for representational states, which typically represent something external to the system. Real-world systems do exist in an environment though, and so we must consider how they interact with that environment, and what makes this interaction non-representational.

As mentioned above, most actual computing mechanisms will include input and output components, without which they would be unable to interact with the external world. These components typically act as transducers, converting external stimuli into a format that is compatible with the mechanism's processors (i.e. strings of digits), and vice versa. To reiterate, though, the mechanism would still be computing in the absence of inputs or outputs, it would just not be of much use or interest to us, its users. Even if we interpret these inputs and outputs as representational, this need not carry over into the mechanism itself, which will continue to function regardless of our interpretation (see Dewhurst 2014: sec. 3).

Autopoietic systems are comparable in the sense that whilst they might not require anything external, they do nonetheless exist in an environment and will be influenced by that environment. Maturana & Varela call these environmental influences "deformations" and "perturbations", and note that they are indistinguishable from internal influences, at least insofar as the dynamics of the system are concerned (1980: 98). What happens is that the system's homeostasis is interrupted by an event (either external or internal), and the system then responds to the event, either by compensating in some way that returns it to homeostasis, or undergoing more radical change that constitutes the creation of a new autopoietic unity (*ibid*: 99). At no point does the system treat an influence as being external rather than internal, thus preserving the status of the system as non-representational. Whilst for practical purposes the system might sometimes require a way of distinguishing internal from external stimuli, this could take the form of a purely functional marker, and would not constitute the introduction of representation into the system.[4]

[4] I thank Paul Bello for bringing this last point to my attention.

Maturana & Varela also specify, "in terms of their functional organisation living organisms do not have inputs and outputs" (1980: 51). Here I take them to be referring to inputs and outputs in a classical sense, i.e. as processes that transfer representational content (see Piccinini 2012: sec. 2.3). The input and output components of a computing mechanism must differ from this classical sense if they are going to remain non-representational. Maturana acknowledges that an autopoietic system has "sensory and effector surfaces", which we could think of as comparable with inputs and outputs, but these do not preclude functional closure, because "the environment [...] acts only as an intervening element through which the effector and sensory [surfaces] interact completing the closure of the system" (Maturana 1975: 318). An autopoietic system is functionally closed because whilst it might respond to an external stimulus (i.e. a perturbation or deformation), the fact that the stimulus is external does not differentiate it in any way from an internal stimulus located at the sensory surface. It is, in effect, treated as a spontaneous internal event.

We should think of the input and output components of a computing mechanism in much the same way. From our perspective, looking in to the mechanism from the outside, they appear to operate in the classical sense that Maturana & Varela rule out. However, from an imagined internal perspective things look quite different. An input component simply produces strings of digits, whilst an output component consumes them. Taken together as a pair, they bear a strong resemblance to any other single processor, which consumes and produces strings of digits in much the same way. The only difference is that the processor is in this case constituted by the external world, which mediates the transformation from output string to input string. Functional closure is preserved, as the computing mechanism, like the autopoietic system, is functionally isolated from the external world. Thus, just the same as in an autopoietic or enactive system, input and output are "co-dependent aspects of a single circular process" (Villalobos and Ward 2014: 5), and there is no point at which representation can enter the picture.

2.6 Thinking Outside the Box

It is perhaps an unfortunate historical accident that computational and autopoietic/enactive approaches to the study of cognition came to be so diametrically opposed to one another. Both had foundations in early cybernetics, and only really began to drift apart after WWII. By the time Maturana & Varela published *Autopoiesis and Cognition* in 1980[5] the two traditions had been separated for several decades, and the computational approach to cognitive science had become dominant. It is in this context that autopoietic theory, and the enactivist tradition

[5] Maturana had published the first half of the book separately in 1972, under the title *Autopoiesis: The Organization of the Living*, and the second half was published a year later.

that it contributed to, is seen as a radical alternative to mainstream representational theories of mind. However, if the mechanistic account is correct, then there is nothing essentially representational about computation, and this divide between the traditions becomes somewhat weaker, perhaps even allowing for complete reconciliation.

This reconciliation would force the computational approach to reconsider the relationship between the brain and its environment. Whilst I have focused on the unity and closure of autopoietic systems, they are also fundamentally involved in their environment, as the later development of enactivism makes clear. Maturana describes this as a consequence of an autopoietic system being unable to distinguish between internal and external perturbations. For the system "there is no inside or outside", meaning that autopoiesis becomes inherently world involving (Maturana 2003: 99).

Here there is an important lesson for the computational theorist – computing mechanisms cannot be all that there is to cognitive science. The nervous system might well be computational, but it is also part of a situated organism, and a theory of cognition that ignores this will fail to capture the full complexity of its target domain. There is something missing from the picture – the world itself – that is restored when we take into account the role of worldly interaction in cognitive activity, understood as a mediating factor between output and input components.

On the other hand, enactivism has sometimes been criticised for failing to acknowledge the important role that the brain plays in cognition, or else failing to give a full account of what it is that the brain contributes to cognitive activity. For better or for worse, contemporary neuroscience is primarily computational, and accepting a non-representational account of computation could allow enactivism to become better integrated with mainstream empirical research. There seems to be no fundamental reason why a synthesis of the mechanistic account with autopoietic/enactive theory could not contribute to a fuller explanation of cognitive phenomena.

2.7 Conclusion

I have shown how an analogy can be drawn between computing mechanisms and autopoietic systems, focusing on the status of representations in both kinds of system. This analogy helps clarify the mechanistic account of computation by demonstrating that a computing mechanism treats paired input/output components as equivalent to processing components, thus preserving functional closure and pre-empting the need for representation. I have also suggested that the analogy might facilitate reconciliation between computationalism and enactivism, and that this would be beneficial to the study of cognition.

What I have not done is make any serious attempt to motivate why I think such reconciliation would be beneficial to both parties. My full thoughts on this must be saved for another occasion, but in brief I think that computation can offer enactivism and autopoietic theory a convincing mechanistic base, and that in return enactivism and autopoietic theory can help computation escape from the metaphysical baggage that representationalism has burdened the received view with. In addition, paying attention to enactivism and autopoietic theory might help the development of non-traditional models of "sui generis neural computation", as hinted at by Piccinini and Bahar (2013), and considered more explicitly by Friston (2013).

I have also not considered the many dis-analogies between computing mechanisms and autopoietic systems, partly in the interest of space but mostly because they are not relevant to the claims that I am making. Computing mechanisms may not be identical with autopoietic systems, but proving that was never my intention. Rather I think it is unlikely that either approach will by itself fully explain cognition, and by comparing the two I hope to have gestured toward a synthesis that might further our understanding.

Acknowledgements I would like to thank Dave Ward, Paul Bello, Stefano Franchi, Mario Villalobos, four anonymous reviewers, and several members of the audience at IACAP 2014 for their helpful comments and suggestions.

References

Bechtel, W. (2005). Mental mechanisms: What are the operations? In *Proceedings of the 27th annual meeting of the Cognitive Science Society* (pp. 208–213). New Jersey: Lawrence Erlbaum Associates.

Craver, C., & Bechtel, W. (2006). Mechanism. In N. Sarkar & N. Pfeifer (Eds.), *Philosophy of science: An enyclopedia* (pp. 469–478). New York: Routledge.

Craver, C., & Darden, L. (2005). Introduction. *Studies in History and Philosophy of Biological and Biomedical Science, 36*, 233–244.

Dewhurst, J. (2014). Rejecting the received view: Representation, computation, and observer relativity. In *Proceedings of AISB 50*. http://www.doc.gold.ac.uk/aisb50/AISB50-S03/AISB50-S3-Dewhurst-paper.pdf

Di Paolo, E. (2005). Autopoiesis, adaptivity, teleology, agency. *Phenomenology and the Cognitive Sciences, 4*(4), 429–452.

Egan, F. (1995). Computation and content. *Philosophical Review, 104*, 181–204.

Fodor, J. (1981). *Representations*. Cambridge: MIT Press.

Friston, K. (2013). Life as we know it. *Journal of the Royal Society Interface, 10*, 20130475.

Hurley, S. (1998). *Consciousness in action*. Cambridge: Harvard University Press.

Hutto, D., & Myin, E. (2013). *Radicalizing enactivism*. Cambridge: MIT Press.

Maturana, H. (1975). The organization of the living: A theory of the living organization. *International Journal of Man-Machine Studies, 7*(3), 313–332.

Maturana, H. (2003). The biological foundations of self-consciousness and the physical domain of existence. In N. Luhmann, H. Maturana, M. Namiki, V. Redder, & F. Varela (Eds.), *Beobachter: Convergenz der Erkenntnistheorien?* (pp. 47–117). München: Wilhelm Fink Verlag.

Maturana, H., & Varela, F. (1980). *Autopoiesis and cognition*. London: Reidel.

Millikan, R. (1995). Pushmi-pullyu representations. In J. Tomberlin (Ed.), *Philosophical perspectives 9: AI, connectionism, and philosophical psychology* (pp. 185–200). Atascadero: Ridgeview Publishig Company.

Mingers, J. (1989). An introduction to autopoiesis. *Systems Practice, 2*(2), 159–180.

Noë, A. (2004). *Action in perception*. Cambridge: MIT Press.

Piccinini, G. (2004). Functionalism, computationalism, and mental contents. *Canadian Journal of Philosophy, 34*(3), 375–410.

Piccinini, G. (2007). Computing mechanisms. *Philosophy of Science, 74*, 501–526.

Piccinini, G. (2008). Computation without representation. *Philosophical Studies, 137*, 205–241.

Piccinini, G. (2012). Computation in physical systems. In E. Zalta (Ed.), *The Stanford encyclopedia of philosophy (Fall 2012 Edition).* http://plato.stanford.edu/archives/fall2012/entries/computation-physicalsystems/

Piccinini, G., & Bahar, S. (2013). Neural computation and the computational theory of cognition. *Cognitive Science, 34*, 453–488.

Ramsey, W. (2007). *Representation reconsidered*. Cambridge: Cambridge Universtiy Press.

Sprevak, M. (2010). Computation, individuation, and the received view on representation. *Studies in History and Philosophy of Science, 41*, 260–270.

Thompson, E. (2007). *Mind in life*. Cambridge: MIT Press.

Varela, F. (1991). Organism: A meshwork of selfless selves. In A. I. Tauber (Ed.), *Organism and the origin of self* (pp. 79–107). Dordrecht: Kluwer Academic Publishers.

Varela, F., Thompson, E., & Rosch, E. (1991). *The embodied mind*. Cambridge: MIT Press.

Villalobos, M., & Ward, D. (2014). Living systems: Autonomy, autopoiesis and enaction. *Philosophy of Technology*, online first. doi:10.1007/s13347-014-0154-y.

Weber, A., & Varela, F. (2002). Life after Kant: Natural purposes and the autopoietic foundations of biological individuality. *Phenomenology and the Cognitive Sciences, 1*, 97–125.

Chapter 3
Are Gandy Machines Really Local?

Vincenzo Fano, Pierluigi Graziani, Roberto Macrelli, and Gino Tarozzi

Abstract This paper discusses the empirical question concerning the physical *realization* (or *implementation*) of a computation. We give a precise definition of the realization of a Turing-computable algorithm into a physical situation. This definition is not based, as usual, on an interpretation function of physical states, but on an implementation function from machine states to physical states (as suggested by Piccinini G, Computation in physical systems. The Stanford encyclopedia of philosophy. http://plato.stanford.edu/archives/fall2012/entries/computation-physicalsystems. Accessed 5 Dec 2013, 2012). We show that our definition avoids difficulties posed by Putnam's theorem (Putnam H, Representation and reality. MIT Press, Cambridge, 1988) and Kripke's objections (Stabler EP Jr, Kripke on functionalism and automata. Synthese 70(1):1–22, 1987; Scheutz M, What is not to implement a computation: a critical analysis of Chalmers' notion of implementation. http://hrilab.tufts.edu/publications/scheutzcogsci12chalmers.pdf. Accessed 5 Dec 2013, 2001). Using our notion of representation, we analyse Gandy machines, *intended in a physical sense*, as a case study and show an inaccuracy in Gandy's analysis with respect to the locality notion. This shows the epistemological relevance of our realization concept. We also discuss Gandy machines in quantum context. In fact, it is well known that in quantum mechanics, locality is seriously questioned, therefore it is worthwhile to analyse briefly, whether quantum machines are Gandy machines.

V. Fano (✉) • R. Macrelli • G. Tarozzi
Department of Basic Sciences and Foundations, University of Urbino, Urbino, Italy
e-mail: vincenzo.fano@uniurb.it; roberto.macrelli@uniurb.it; gino.tarozzi@uniurb.it

P. Graziani
Department of Philosophical, Pedagogical and Economic-Quantitative Sciences,
University of Chieti-Pescara, Chieti, Italy
e-mail: pierluigi.graziani@unich.it

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_3

3.1 Church, Turing and Gandy

When we say that a function[1] is ‘effectively calculable’ we mean, roughly speaking, that there is a procedure that, starting from any argument, always produces the correct value. The intuitive notion of ‘procedure’ we are introducing is purely epistemic: it has nothing to do with the physical world. A ‘procedure’ in this sense is a *finite* set of *distinct* operations that lead without *creativity* from the argument to the value of a function. Thus, intuitively, we define ‘calculable’ in terms of ‘procedure’ and this in terms of ‘operation’. But at this point we do not know precisely what ‘operations’ are. The fact that we get involved in a hopelessly inaccurate regression may suggest a change of strategy.

Church-Turing’s Thesis, as is known, states that the set of effective calculable functions (intuitive notion) and the set of Turing-computable[2] functions (formal notion) coincide.

Church-Turing Thesis: a function is effectively calculable if and only if it is computable by a Turing machine.

According to Alan Turing (1936, p. 249), every argument in favour of the thesis that everything calculable is computable appeals to intuition, and for this reason it is rather unsatisfactory from a mathematical point of view. Consequently, we prefer to consider the so-called *Church-Turing Thesis* as an *explication* (in Carnap’s sense[3]) of the calculability notion[4] which makes use of the logical notion of a Turing machine and with respect to which you can pose interesting questions.[5] In this paper we will address the empirical[6] question concerning the *physical realization* (*implementation*) of computation representable by a Turing Machine (TM).

[1] Let us consider only those functions whose arguments and values are natural numbers.

[2] From now on we will use the term “computable”.

[3] The term explication appears for the first time in *The Two Concepts of Probability*, but a more complete discussion of this concept can be found in the first chapter of *Logical Foundations of Probability* in which Carnap writes: “By an explication we understand the transformation of an inexact, prescientific concept, the explicandum, into an exact concept, the explicatum” (cap.1, p.1). [...] “The term ‘explicatum’ has been suggested by the following two usages. Kant calls a judgment explicative if the predicate is obtained by analysis of the subject. Husserl, in speaking about the synthesis of identification between a confused, nonarticulated sense and a subsequently intended distinct, articulated sense, calls the latter an ‘Explicat’ of the former. What I mean by ‘explicandum’ and ‘explicatum’ is to some extent similar to what C.H. Langford calls ‘analysandum’ and ‘analysans’: ‘the analysis that states an appropriate relation between the analysandum and the analysans’; he says that the motive of an analysis ‘is usually that of supplanting a relative vague idea by a more precise one (cap. 1, § 2)’”. See Carnap (1945, 1950).

[4] For a discussion see, for example, Klenee (1952, pp. 317–319), Mendelson (1990, p. 229), Soare (1996), Herken (1995), Olszewski et al. (2007), Kripke (2013).

[5] See Herken (1995), Olszewski et al. (2007).

[6] The term ‘empirical’ is to be understood in a purely epistemological way, that is, as a comparison with experimental data. In spite of this no applications are involved.

In 1980 Robin Gandy described and analysed what today are called *Gandy Machines*. This notion is a very interesting contribution to the computability theory, because it is a mathematical concept that, on the one hand, is equivalent to another mathematical notion, namely Turing computability, and on the other hand, it is a very interesting formal model of some physical situation concerning computation. As is well known, the concept of Turing computability is modelled on the operations a human computor performs. In contrast, Gandy machines are models of *discrete* physical devices. Therefore, it is possible to discuss Gandy machines *from a physical point of view* and to analyse some questions concerning the physical realization of computation. In particular this paper is divided into two parts: the first (Sect. 3.2) attempts to explicate the notion of realization on the basis of the existent scientific literature and proposes a new definition; the second applies our definition of realization to the case study of Gandy machines, showing that if they are interpreted in a physical sense, then in Gandy analysis (1980) there is an *inaccuracy* with respect to the locality property (Sects. 3.3 and 3.4); finally, we discuss Gandy machines and quantum non-locality (Sect. 3.5).

3.2 The Realization

To explain the notion of realization (implementation) of computation we would like to start with a very simple example.

Balls that move on a pool table can be represented mathematically. Let us call b_i the pool table with all the balls at time i. Consider b_i as a sort of *rigid designator*,[7] i.e. b_i refers to the pool table at time i as a proper name refers to the named person, without the use of any of its properties, and then referring to it in all its individuality. Let us imagine that the pool table is completely isolated, that is, without exchanges of matter and energy with its environment.[8] So b_i refers rigidly to the time steps of this small universe.

[7] The identification of *rigid designators* in natural languages is a very complex issue and it is still a matter of controversy. In our paper, the term 'rigid designator' is to be understood in the sense that we refer to parts of the world in a manner as neutral as possible. The use of rigid designators is not essential for the purposes of our argument, it helps us to simplify our discourse, and it is a useful expedient to avoid a conceptual connotation of physical objects of which we are speaking.

[8] It is impossible to stick to an excessive rigor in defining the delicate and difficult concept of realization. So we have chosen the practical way of modern natural science, which consists of bringing the object of our investigation to a partially ideal situation that can find approximate concrete examples in the world. In our laboratories, for example, it is trivial that gravity can never be screened, so our real system is necessarily not isolated. However it is not difficult to find situations in which gravity does not have any relevant physical effect from the computational point of view. For example on the computer that we use to write this paper.

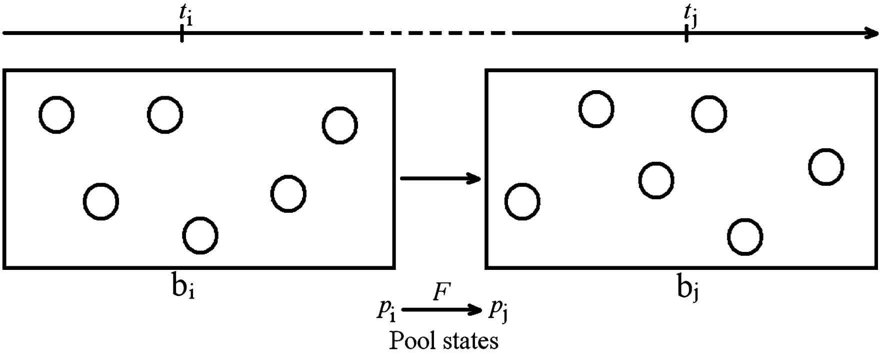


Fig. 3.1 A pool table with balls at time i and at the time j. A simple function F can predict from p_i, the pool state at time i, the state p_j at time j

Classical mechanics uses *state space* R_{mc} to represent some measurable[9] characteristics (positions and velocities) of the pool system in time: $P(b_i) = p_i$. We call p_i the pool state at time i. We assume that time is discrete, distinguishable from space and equipped with a linear order,[10] since on the one hand we will deal with digital and not with analog computability,[11] on the other in our case relativistic effects are irrelevant. At every moment i, P brings b_i in only one state p_i. Moreover classical mechanics has a simple[12] function F that has as arguments a couple constituted by a point of R_{mc} and a lapse of time, and as value a point of R_{mc} which can predict the next state, i.e. such that if p_i corresponds to b_i, then $P\left(b_j\right) = p_j = F(p_i, j - i)$, with $i > j$ (Fig. 3.1). More in general we say that the pool table is a *model* of a theory T with laws L (classical mechanics) and that the discrete time is an *Intermediation* function between the continuous physical system and the discrete computation.

Moving, now, from this simple example we will try to analyse the relationship between the notion of computability and that of physical reality. More precisely, we will focus on the connection between the two elements most far apart in the relation between computational representation and the physical world, i.e. machine states on one side, physical states on the other. We can start by refining the notion of model and intermediation function.

[9] Characteristics could be non observable as well.

[10] From now on we move in a classical physics context, where space is represented in Euclidean terms.

[11] Gandy (1993) considers the enlargement to analog machines mathematically irrelevant. But see Kieu (2002).

[12] Such functions always exist, but in general they are not simple. Physics can reduce their complexity.

Given a physical system w, first we have to find the best available theory, which governs it. In particular we will talk about w as a model of a theory T with laws L (Beggs and Tucker 2007).

When we build a correspondence between physical systems and computations, we need to interpose an *Intermediation* function between them. In fact, physical systems could be in a continuous set of states, whereas computations are discrete, so a sort of Intermediation (I) function is needed in order to make the state space of the physical system discrete.[13]

Let us consider, at this point, the state[14] of a Turing machine represented as Ps_kq_lQ, where P and Q are variables on strings of symbols s printed on the corresponding boxes of the tape, while s_kq_l is the box in which you will find the machine head, that is the internal state q_l of the head is over a particular box of the tape and the instruction is performed over the scanned box s_k. We know that Ps_kq_lQ is potentially infinite, i.e. as long as you want, even though always finite, while the indices k and l vary over a bounded domain. Ps_kq_lQ can be thought of as a point (call it m_i) in a state space, which we can call R_{TM}. It is easy to find the physical system that realizes Turing machines. The personal computer on which you are reading or printing this paper is surely one of such examples.

We note that physics moves bottom-up, so to speak, i.e. it seeks mathematical representations of parts of the world, whereas computer science moves top-down,[15] i.e. it seeks achievements of computations in parts of the world. In this respect the *realization concept* is a central notion in computer science. In fact, we look at the world through the glasses of our best physical theories; therefore, when we attempt the realization of a computation we do not establish a correspondence between points of R_{TM} and elements of the class B of b_i, but between R_{TM} and R_{TL}, where R_{TL} is the state space of a physical theory TL.[16] However, a Turing machine is not defined only by its states Ps_kq_lQ, but also by a sufficiently large[17] set of quadruples of the type $s_kq_lOq_m$ where s_kq_l is the internal state of the head and the symbol of the

[13] It is true that our current computers are all based on quantum-mechanical effects due to the doping of semiconductors, but in general a doped semiconductor could stay in a continuous set of possible physical states (velocity, temperature, positions etc) and our intermediation function chooses exactly the one we are interested in, that is the two levels of potential.

[14] We use the term 'state' (distinguishing it from the term 'internal state') because we wish to highlight the connection between the concepts of machine states and physical states, but our reasoning could easily be reformulated using the more familiar notion of 'configuration'.

[15] Something very similar is maintained in Horsman et al. (2014). The latter's approach is comparable to ours in the sense that it emphasizes the importance of the relation between computational and physical space. They express something very similar to condition 2. of our definition of realization (see *infra*), speaking of commuting diagrams. In spite of this our feeling is that their approach, though interesting, is naive from an epistemological point of view.

[16] Remember that in the pools case, TL is classical mechanics.

[17] There need not necessarily be a different quadruple for each couple s_kq_l.

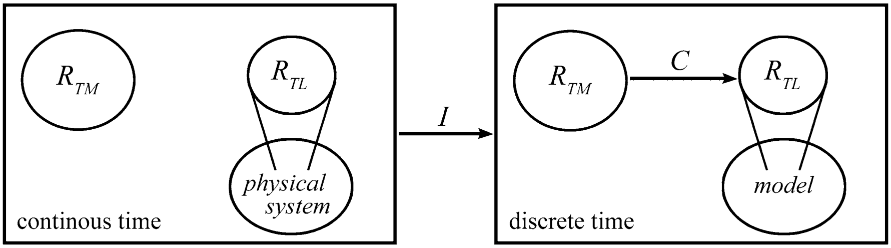


Fig. 3.2 The injective function C establishes a correspondence between the point of R_{TM}, the state space of a Turing machine (*TM*), and the point of the state space of a physical theory with laws R_{TL}, discretized through an intermediation I. When a physical system w implements a computation, w becomes a model

box in which it is located, while Oq_m indicates that the successive internal state of the head is q_m and O is the operation of head movement either forward or backward or the change of the symbol printed in the head position.

Therefore, in order for an *isolated* system w to be the *realization* of a Turing machine *TM* with respect to Intermediation I it must hold that[18]:

1. w is a model of a physical theory *TL* with its state-space R_{TL} and deterministic laws L which apply in w. R_{TL} must be a discrete space through an intermediation I in order to make a correspondence with computations. We know that measured time is discrete as well, since from a technological point of view there is always a minimal threshold of observability. At each instant of time w must be represented by a single point of R_{TL} (Fig. 3.2).
2. There is an injective function[19] C that associates each possible state of *TM*, i.e. arrays of type Ps_kq_lQ (that is points m_i of R_{TM}), to a point $\tilde{p}_i$ of R_{TL} such that $C(m_i) = \tilde{p}_i$ and $C\left(m_j\right) = \tilde{p}_j$ and the quadruples of *TM* stipulate that after n steps m_i goes into m_j then it must be *in condition of normal operation*[20] $L(\tilde{p}_i, n) = \tilde{p}_j$ (Fig. 3.2).

A few observations about this definition are in order.

Informally, condition 1 states that we look at the world through the glasses of our best physical theories and not at the world directly; and it highlights the structure of the specific theory which is adequate to analyze the specific parts of the world we would like to look at.

Informally, condition 2 states that if *TM* leads from a certain input to a certain output, then C must be so built that the physical state corresponding to the input must

[18] On the notion of 'realization' it is necessary to move from Giunti's important study, 1997, especially par. 16. Our approach is partially similar from a formal point of view, but it is conceptually different, since it involves the laws of physics.

[19] In general, C will not be surjective, since not all the R_{TL} space is used. Being C injective, it will be invertible as well.

[20] We will explain this condition in a subsequent comment.

cause deterministically the physical state corresponding to the output in accordance with the L laws. Note that our definition is not based, as usual, on an interpretation function of physical states, but on an implementation function from machine states to physical states.[21]

It is clear, from what we have said above, that before establishing C it is necessary to make the state space R_{TL} discrete. This is the *intermediation* between computations and physical systems.[22] Obviously time is discrete since, from a technological point of view, there is always a minimal threshold of observability. In addition, C must be so constituted that it does not associate a different physical state to each state of the TM, since in the case we had to find an infinite series of discrete points in R_{TL}. Therefore, C must be based on a code so as to transform this actual infinite sequence into a finite number of correspondences, which can generate a potentially infinite sequence (like ciphers for numbers, Stabler 1987).

We think that this concept of realization has some advantages with respect to others presented in the scientific literature (Stabler 1987; Scheutz 2001; Piccinini 2012). For example it is important to note that:

- we, as suggested by Cotogno (2003, pp. 187–188), have specified the notion of ‘realization’ by making explicit the fact that it is not a relationship between the mathematical concept of a Turing machine and a part of the world, but between the former and the physical theory true for that part of the world. This is an important upgrade[23] with respect to preceding literature on the subject. Thus, in this case TL will be a physical theory endowed with various principles;
- our definition, as suggested by Piccinini (2012), is not based, as usual, on an interpretation function of physical states, but on an implementation function from machine states to physical states[24];
- our definition should be sufficiently restrictive to block Putnam’s theorem (1988, pp. 121–125), according to which each open physical system would be able to realize every automatic procedure.[25] Indeed our system is *isolated*, so that in this

[21] As suggested by Piccinini (2012, p. 9).

[22] See Scheutz (1999).

[23] See also Horsman et al. (2014).

[24] This definition is different from what Piccinini dubs a ‘simple mapping account’. The latter turns out to be mathematically more vague. As a result, the perspective added is that we restrict ourselves to mappings that are acceptable. Nevertheless the decision to change the mapping from functional states to states of the machine is not the significant part of our argument. Rather, this choice depends on the fact that we want to tackle the relationship between the term ‘computational’, on the one hand, and ‘physical world’, on the other, at the greatest possible distance between them. For this reason we speak of states of the machine. Our work differs from the standard philosophical literature, not only in the use of the state of the machine, but also in the definition of the realization, which takes into account all the epistemological problems, to our knowledge, raised till now upon this concept.

[25] Using the words of Chalmers: “The ambitions of artificial intelligence rest on a related claim of computational sufficiency” (Chalmers 1996, pp. 309), that is, “the right kind of computational structure suffices for the possession of a mind” (Chalmers 2011, p. 325). So, “computation will

case what Putnam calls the *Principle of Noncyclical Behavior* (that is a "system is in different maximal states at different times" (Putnam 1988, p. 121)) does not hold. The principle, fundamental for Putnam's proof, "will hold true for all systems that can 'see' (are not shielded from electromagnetic and gravitational signals from) a clock. Since there are natural clocks from which no ordinary open system is shielded, all such systems satisfy this principle" (Putnam 1988, p. 121). But our system is isolated and so cannot see any external clock. Our definition avoids Putnam's objection also because it is based on L's (laws) capacity to justify counterfactuals. Putnam's theorem in fact is founded on a mere *a posteriori* projection of a computation on a physical system; in our perspective, on the contrary, w can realize a computation with different inputs and outputs;

- our definition is not what Piccinini (2012) dubs "simple mapping account", since it is based on physical laws L: we do not use either epistemological or ontological concepts, such as counterfactuals, dispositions or causality; instead our definition is based only on the notion of physical law and its capacity to justify counterfactuals, w can realize a computation with different inputs and outputs;
- our definition avoids Kripke's objection, discussed by Stabler (1987) and Scheutz (2001), according to which the respect of laws L is not sufficient to distinguish a functioning machine from a broken one, because we have added the clause "in condition of normal operation". The introduction of this surreptitiously normative assumption is due to the fact that, in our opinion, there is no principle able to respond to Kripke's objection. It is necessary to adopt a *design stance* with respect to the part of the world that we are considering, to overcome the Kripke's objection. Obviously the endorsed design could be changed. Therefore the proposed solution can only be a form of reflective equilibrium between the normative and the explanatory point of view.

It is possible to understand our definition also through a simple image (Fig. 3.3), taken from Horsman et al. (2014) concept of commuting diagram.

That is if either we apply C^{-1} to the state $\tilde{p}_i$ and then the computation TM, or before we apply the laws of TL and then C^{-1}, we achieve the same result m_j.

Therefore, the present concept of realization appears to be an interesting candidate to study computation in physical systems and in this perspective we will analyse Gandy machines as a test bench for it.

provide a powerful formalism for the replication and explanation of mentality" (Chalmers 1996, pp. 309–310). Hilary Putnam's theorem says that "every ordinary open system is a realization of every abstract finite automaton" (Putnam 1988, p. 121), and its proof requires two physical principles; a principle of continuity, and the principle of Noncyclical behaviour (for more details see Putnam 1988, pp. 120–125). "Together with the thesis of computational sufficiency, this [theorem] would imply that a rock has a mind." [...] "We must either embrace an extreme form of panpsychism or reject the principle on which the hopes of artificial intelligence rest. Putnam himself takes the result to show that computational functionalism cannot provide a foundation for a theory of mind" (Chalmers 1996, pp. 309–310). See also Piccinini (2012).

Fig. 3.3 Commuting diagram

$$\begin{tikzcd} m_i \arrow[r, "TM"] & m_j \\ \tilde{p}_i \arrow[u, "c^{-1}"] \arrow[r, "L"'] & \tilde{p}_j \arrow[u, "c^{-1}"'] \end{tikzcd}$$

Taking its cue from our idea that the Church-Turing thesis can be regarded as a definition through Turing machines of what is computable, we can read Robin Gandy's 1980 paper as an attempt to provide an explanation of what is computable not by a human computor, but rather by a real machine that obeys the laws of physics and in particular the constraint imposed by the relativistic speed limit. Then, using our definition of realization, we can introduce Gandy machines and analyze the relation between these and parts of the physical world. This, for us, is a case study to show the relevance of our realization concept.

3.3 Gandy Machines: A Case Study

Robin Gandy's 1980 paper provides a very useful formalism to investigate the relation between computation as a mathematical concept and computation as a physical dynamics.

Gandy thought that the concept of Turing Machine was too anthropomorphic and for this reason all the artificial computing devices cannot be reduced to it. Therefore, Gandy generalized Turing's analysis. While, in fact, Turing had identified some constraints that had to be fulfilled by a man who applies an algorithm, Gandy identified more general constraints that must be fulfilled by any algorithmic computation process. Gandy chose constraints that depend only on physical considerations.[26]

Keeping in mind the *Church-Turing Definition*, we can introduce Gandy's constraints using the following *Gandy Thesis*:

Gandy Thesis: a deterministic and discrete physical machine that satisfies a principle of impenetrability of bodies together with a principle of locality and classical laws for shock and motions can realize (in the sense defined above) only Turing-computable functions.[27]

[26] For a interesting discussion of these issues see Tamburrini (2002).

[27] We attributed this view to Robin Gandy, because it is very similar to what he calls "Thesis M" (Gandy 1980, p. 124). So do Copeland and Shagrir (2007), although they are convinced that there

In his 1980 paper Gandy provides an apparently decisive argument favouring the significance of this thesis. In fact, he takes into account a certain class of concrete machines, currently called *Gandy machines*,[28] which respect Gandy's constraints. Gandy showed that if a function is computable by a Gandy machine, then it is always also computable by a Turing machine.[29]

Let us analyze the constraints in question starting with the *principle of impenetrability of bodies*. To formulate this principle satisfied by Gandy Machines we have to assume that space and matter are discrete, that is for space regions and parts of matter a Mereology with atoms holds.[30] It is clear that this is a methodological assumption, useful for our analysis about the link between computation and physical reality, not a reasonable hypothesis about the *nature* of space and matter.[31] We can formulate the principle in the following way:

Impenetrability: given an atomic body c_1 that occupies an atomic region of space r_1 no proper or improper part of another body can occupy a proper or improper part of r_1 at the same instant t_1.

In addition to this, (a) Gandy machines respect the principle of *conservation of momentum and kinetic energy*, which regulates the motions of the parts of the machine (we instead consider both gravity and friction irrelevant); (b) they are assembled at each step from a limited number of pieces or atomic parts, so they have *limited size*; (c) are *deterministic*, that is, the next state of the machine is determined by the previous state of the machine; (d) and they are *discrete*, i.e. they can assume a bounded number of distinct states in a finite time. Finally, Gandy machines respect a *principle of locality*, the core of our test bench; we will focus the next sections on this constraint.

Given aphysical theory we can enunciate this principle of locality in the following generic way:

are discrete and deterministic physical machines that can compute hyper-computable functions. This is not our topic. However Cotogno (2003) is rather sceptical about hyper-computation. Pitowsky and Shagrir (2003) show that a physical digital hypercomputer, whose existence is compatible with General Relativity, is a Gandy machine and it can compute a function that is non Turing-computable. See also Kieu (2002), Syropoulus (2008). Moreover consider that originally Gandy has in mind not only classical mechanics, but classical electromagnetism as well.

[28] Later we will give a sketchy presentation of this notion.

[29] Remember that we have specified the notion of 'realization' making explicit the fact that it is not a relationship between the mathematical concept of a Turing machine and a part of the world, but between the former and the physical theory true for that part of the world. Therefore, in this case *TL* will be a physical theory endowed with various principles.

[30] See for instance Simons (1987, pp. 41–45). See also Calosi and Graziani (2014).

[31] Nonetheless many physical theories based on an atomistic conception of matter and space have been proposed; one of the most recent and interesting is Rovelli (2004).

Locality: given two regions of spaces r_1 and r_2 separated by a distance l, a physical change in r_1 at time t_1 cannot cause a physical change in r_2 at time t_2, with $l > c\,(t_2 - t_1)$.

In this context c is the speed of light in vacuum; note that it is not necessary to specify further the notions of causality and physical change used in the definition.

In the 1980 paper, Gandy (1980, p. 135) provided the following definition of locality for his machines:

Gandy's Locality Property: The next state, $\mathbf{F}x$, of a machine can be reassembled from its restrictions to overlapping "regions" s and these restrictions are locally caused. That is, for each region s of $\mathbf{F}x$ there is a "causal neighbourhood" $t \subseteq TC(x)$ of bounded size such that $\mathbf{F}x \upharpoonleft s$ depends only on $x \upharpoonleft t$.

Our aim is to analyse this last principle in more detail.

In the definition of Gandy's Locality, x is a possible state of the machine and $\mathbf{F}$ is a function that turns a deterministic machine from one state to another. Note that $\mathbf{F}$ must allow a limited increase in machine parts.[32] Furthermore, s and t are either atomic or not atomic parts of the machine and $TC(x)$ is the transitive closure of x. Finally $x \upharpoonleft t$ means what is left of x after we removed all *Urelemente* and sets that do not appear in t, i.e. the restriction of x to t.

In his analyses Gandy uses the formalism of hereditarily finite sets over a set of atoms (*Urelemente*)[33] to represent his machines. Note that Gandy's intermediation I is precisely the hereditary set theory with a finite number of *Urelemente*. Since Gandy machines are bounded automata, their hierarchical representation necessarily has a bounded number of levels. The elements of this hierarchy, in contrast to what happens in today's axiomatization of sets, are not sets, but *labels*, which refer to parts of the machine, to their physical properties and regions in space.

In this language, for example, a machine state of *type* Ps_mq_nQ can be easily represented. Let's consider the following simple situation:

We have a tape with 4 boxes on which 0s and 1s are printed (Fig. 3.4). The head is marking the third box and the internal state is q_5. We assign labels $e_1 - e_4$ to the

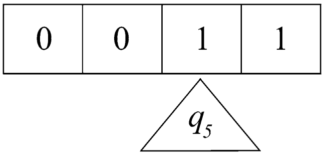


Fig. 3.4 An example of a machine state of type Ps_mq_nQ

[32] Gandy also shows that several forms of weakening on the conditions of function $\mathbf{F}$ would allow computing functions not computable in the sense of Turing.

[33] See Sieg and Byrne (1999) for a clear introduction to Gandy's formalism. Sieg (2002) radically shifts the methodological perspective of Gandy's argument from simulation by one computational mechanism, to a perspective by an axiomatic approach.

4 boxes, e_5 and e_6 respectively to the two symbols 0 and 1; the n possible internal states have labels $e_7 - e_{6+n}$. So the state of the above machine becomes:

$$\{\{e_1, e_2\}, \{e_2, e_3\}, \{e_3, e_4\}, \{e_1, e_5\}, \{e_2, e_5\}, \{e_3, e_6\}, \{e_4, e_6\}, \{e_3, e_{11}\}\}$$

where the first 3 pairs of *Urelemente* describe the tape, the 4 successive, the symbols printed in the boxes and the last symbol refers to the box indicated by the head with the internal state of the latter.

Note again that the language introduced by Turing is extremely intuitive and schematically depicts the human computor as evidenced by the famous section 9 of his 1936 paper. On the contrary, the language developed by Gandy allows us to represent very different machines from that of Turing; in particular it can represent the evolution step by step, of a bounded number of intertwined parallel computations, through the increasing of the levels of hierarchical sets.

As we have said, in *Gandy's Locality Property* s and t are either atomic or non-atomic parts of the machine and $TC(x)$ is the transitive closure of x. For instance, the transitive closure of $\{\{e_1, e_2\}, e_3\}$ will be $\{\{e_1, e_2\}, e_3, e_1, e_2\}$, since e_1 and e_2 belong to $\{e_1, e_2\}$, which in turn belongs to $\{\{e_1, e_2\}, e_3\}$, therefore e_1 and e_2 must be added to $\{\{e_1, e_2\}, e_3\}$ in order to attain its transitive closure. The last part of *Gandy's Locality Property* says that: "for each region s of $\mathbf{F}x$ there is a 'causal neighbourhood' $t \subseteq TC(x)$ of bounded size such that $\mathbf{F}x \uparrow s$ depends only on $x \uparrow t$". Therefore, this last part means that for each part of the new state of the machine (s is a part of $\mathbf{F}x$) there must be a part (t) of the previous state (x) of limited size such that the restriction to s of the new state ($\mathbf{F}x \uparrow s$) depends only on the restriction to t of the old state ($x \uparrow t$). This, finally, means that for any part of the new assembly of the machine, there must be a limited part of the previous assembly that influences only that part of the new step.

Gandy (1980, p. 126) justifies the physical locality constraint by writing that "there is an upper bound (the velocity of light) on the speed of the propagation of changes". In addition, in the final part of the essay (*Ibid*., p. 145) he regrets not being able to find a formulation of his machine fitting with classical mechanics, where perfectly rigid bodies may transmit signals with infinite speed; or better, a machine that, in addition to allowing infinite speed transmissions, does not compute all functions, including those not computable according to Turing. These quotations suggest that Gandy's bound of locality imposed on the $\mathbf{F}$ function really has a physical nature[34] and requires a deeper analysis.

[34] Probably for this reason David Deutsch (1985), moving explicitly from Gandy (1980), introduced his "Church-Turing principle", which would have physical as well as mathematical meaning. The principle proposed by Deutsch is rightly criticized by Copeland (2015) for its vagueness.

3.4 The Physical Sense of Gandy Machines and Locality: An Inaccuracy

Using our definition of realization and the previous analyses of Gandy machines, *interpreted in a physical sense*, we can return to discuss Gandy's Thesis[35]:

Gandy's Thesis: a deterministic and discrete physical machine that satisfies a principle of impenetrability of bodies together with a principle of locality and classical laws for shock and motions, can realize only Turing-computable functions.

The first question we must ask is: what is *TL* in our case? That is, what is the physical theory and its laws that we assume to apply to the part of world we are investigating? Here the answer is simple: *TL* consists of four principles that we have assumed: impenetrability, locality, conservation of momentum and kinetic energy. We must now ask what the *C* function that represents the realization is. In fact we have not precisely defined *C* yet, but the example is clear enough. And e_i are either bodies or regions,[36] for which the hereditary theory of sets with atoms establishes a correspondence between states of the machine (represented by non-empty hereditarily finite sets over an infinite set of atoms (Sieg 2008, p. 147)) and different physical situations.[37] Let us introduce a Gandy **F** function: using the words of Sieg (2008, p. 147)[38] we consider pairs (**D**,**F**), where **D** is a structural class of states, i.e. class of states closed under $\in$-isomorphism[39] determined by a permutation of machine states, and **F** an operation from **D** to **D** that transforms a

[35] See Shagrir (2002) for various interpretations of Gandy Machines. Shagrir (2002) emphasizes that in Gandy's paper there is an *ambiguity*, since it is not clear whether he intends his thesis as a physical one – as in our paper – or as a mathematical one. That is, is Gandy's machine a mathematical concept inspired by physical models, or is it an actual physical system? In the first case, Gandy shows an important theorem according to which Gandy's machines (parallel) computes only Turing computable functions. But we are interested in the physical interpretation of Gandy's point, which is present in many passages of this splendid paper. We also think that the *ambiguity* noted by Shagrir has contributed to hiding the *inaccuracy* we are considering here. See also Shagrir (1997).

[36] In Gandy's formalism there is no distinction between labels of regions and labels of bodies.

[37] Gandy dubs this hierarchical structure "stereotype", because isomorphic physical situations have the same computational relevance.

[38] We do not discuss the Gandy's formalism but, for our purpose, we refer the reader to the simpler and clearer formalism of Sieg (2008). In our opinion, the use of Sieg's formalism does not affect our contention that there is an inaccuracy in Gandy.

[39] Roughly speaking two structures x and y are $\in$ -isomorphic if they are *isomorphic over the empty set*, that is there is a permutation π which is the identity on an empty set and which carries x into y. For more details see Gandy (1980, pp. 127–129). See also Sieg (2008, pp. 147–148).

given state into the next one. Letting $\pi(x)$ stand for the result of applying the $\in$-isomorphism determined by a permutation π to the machine state x. **F** is a Gandy structural function if and only if $\mathbf{F}(\pi(x))$ is $\in$-isomorphic to π $(\mathbf{F}(x))$, and this isomorphism must be the identity on the atoms occurring in $\pi(x)$.[40] In other words, **F** is a function that turns a deterministic machine from one state to another, and allows a limited increase in machine parts. Let us assume that e_1 and e_2 are two bodies and that e_3 and e_4 are two regions. In step 1 e_1 is 1 m far from e_2. A rather "big" machine! Imagine that the discreteness of the evolution of the machine has a temporal rhythm of 2 GHz per second, like today's best personal computer. This means that if state 2 comes after 0.5×10^{-9} seconds, and in step 2 bodies e_1 and e_2 have traded places – which is possible given the Gandy **F** function – locality is violated, because the bodies should move at a greater speed than that of light, namely 2×10^6 km per second. It follows that the locality respected by Gandy machines (*G-Locality*) has this form:

G-Locality: given two regions of spaces r_1 and r_2 that are l far, a physical change in r_1 at time t_1 cannot cause a physical change in r_2 at time t_2, with $l > V(t_2 - t_1)$, with $V \in \Re$ bounded.

In other words, the speed of casual influence is certainly limited, but as large as you want, i.e. there may be very large machines with very high frequency, in which the speed of light is easily exceeded.

Pay attention that all Gandy's proof is crucially based on the boundedness of his machine. There is a limit in the number of parts, which could not be exceeded by the application of **F**. But this limit is not determined, so that if one chooses a sufficiently large bound, the machine can violate relativity. Gandy argued that classical mechanics admits interactions with infinite speed, so it would violate this type of *G-Locality*. In fact, few pre-Einsteinian physicists were really convinced of this possibility and certainly not Newton, who does not explicitly admit a hypothesis of this type in the *General Scholium* of Newton's *Principia*.

Our feeling is therefore that Gandy machines is a realization that respects if not the letter at least the spirit of the laws of classical mechanics, but could violate the limit of light velocity.[41] By contrast in relativistic theories the speed is not only bounded, but never larger than the speed of light in vacuum ($c = 299792,458$ km/s). To implement this in the principles that govern the machine, Gandy should add a condition on the transition function **F** to prevent too fast movements between one assembly and the next. This would involve a determination of the bounded number of parts, of their dimensions and of the time frequency.

[40] For mathematical details see Gandy (1980, pp. 127–129) and Sieg (2008, pp. 147–148).

[41] Note that this is an inaccuracy in Gandy's analyses. A very interesting second question is: can we find the same inaccuracy also in other formalizations of Gandy Machines? We will analyze this problem in a new paper.

3.5 The Case of Quantum Mechanics

It is well known that in quantum mechanics, locality is seriously questioned, for which it is worthwhile to briefly discuss whether quantum machines are Gandy machines. Recall that Gandy machines have bounded velocity. Therefore we have to wonder whether quantum mechanics allows superluminal causation. Note that this is a secondary problem concerning quantum computation, since the most important effect, emphasized by Deutsch (1985), would be superposition.

The best way to express quantum violation of locality is to refer to factorizability. Consider two measurements of the spin M and N made on two spacelike separated particles. M is measured in direction a and N in direction b. We call the corresponding results r_m and r_n. Then we can thus express factorizability between results:

$$P(r_m, r_n/a, b) = P(r_m/a)P(r_n/b) \tag{3.1}$$

where P is a probability measure. This equality expresses a *sufficient* condition for the causal independence of the two measurements. This means that "not (3.1)" does not entail a violation of locality. We know that a consequence of this equation, namely Bell inequality, in the case of two particles linked, for example, by a singlet state, is experimentally violated.

Let us now ask whether this means that locality as we have defined it is violated.

The (3.1) is equivalent to:

$$P(r_m/a) = P(r_m/a, b, r_n) \tag{3.2}$$

$$P(r_n/b) = P(r_n/b, a, r_m)$$

As demonstrated by Jarrett (1984), (3.2) is equivalent to:

$$P(r_m, a) = P(r_m, a/b) \tag{3.3}$$

$P(r_n, b) = P(r_n, b/a)$ (parameter independence).

Together with:

$$P(r_m, a/b) = P(r_m, a/b, r_n) \tag{3.4}$$

$P(r_n, b/a) = P(r_n, b/a, r_m)$ (outcome independence).

If (3.1) is violated then either parameter independence or outcome independence must be violated.

We note, however, that while the violation of the parameter independence would allow superluminal signals to be sent, the same is not true for outcome independence. So only the violation of parameter independence would conflict with locality. It is often argued (in different contributions of the volume edited by Cushing and McMullin 1989) that quantum mechanics requires only violation of the outcome independence. However Maudlin (1994) showed that we cannot know whether quantum non-locality entails the violation of either outcome independence or parameter independence.

Our short observations lead to the conclusion that till now we do not know whether or not quantum effects violate the condition we have called "locality". If it turns out that locality was violated, we should seriously consider the possibility that there are machines that can calculate functions that are not computable. For now, however, we must suspend judgment.

3.6 Conclusion

To sum up, Gandy's contribution provides actual elements to reflect on the empirical problem of the relationship between parts of the world and computation. Gandy shows that parallel computation does not expand the Turing computability concept. He also demonstrates that a physical machine that evolves with bounded velocity only calculates Turing-computable functions. Our analysis does not question these points, since if a Gandy machine is T-computable, this would be even more true with respect to actual locality machines. However, it teaches us that we must be very careful in distinguishing between the physical and the mathematical level when dealing with these problems and that this distinction, by means of a precise definition of realization, therefore allows us to highlight an interesting inaccuracy.

Acknowledgments We thank Claudio Calosi, Maurizio Colucci, Paola Gentili, Marco Giunti, Rossella Lupacchini, Wilfried Sieg and Guglielmo Tamburrini for their comments and suggestions. We thank the reviewers who have commented our paper and the scholars attending the IACAP 2014 Conference for their questions.

References

Beggs, E. J., & Tucker, J. V. (2007). Can Newtonian systems, bounded in space, time, mass and energy, compute all functions? *Theoretical Computer Science, 371*(1), 4–19.

Calosi, C., & Graziani, P. (2014). *Mereology and the sciences. Parts and wholes in the contemporary scientific context* (Synthese library, Vol. 371). Heidelberg/New York/Dordrecht/London: Springer.

Carnap, R. (1945). The two concepts of probability. *Philosophy and Phenomenological Research, 5*(4), 513–532.

Carnap, R. (1950). *Logical foundations of probability*. Chicago: Chicago University of Chicago Press.

Chalmers, D. J. (1996). Does a rock implement every finite-state automaton? *Synthese, 108*, 309–333.

Chalmers, D. J. (2011). A computational foundation for the study of cognition. *Journal of Cognitive Science, 12*, 325–359.

Copeland, B. J. (2015). The Church-Turing thesis. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy*. Stanford: Metaphysics Research Lab. http://plato.stanford.edu/archives/sum2015/entries/church-turing. Accessed 24 Sep 2015.

Copeland, B. J., & Shagrir, O. (2007). Physical computation: How general are Gandy's principles for mechanism. *Mind and Machines, 17*(2), 217–231.

Cotogno, P. (2003). Hypercomputation and the physical Church-Turing Thesis. *British Journal for the Philosophy of Science, 54*(2), 181–223.

Cushing, J. T., & McMullin, E. (1989). *Philosophical consequences of quantum theory*. Notre Dame: University of Notre Dame Press.

Deutsch, D. (1985). Quantum theory, the Church-Turing principle and the universal quantum computer. *Proceedings of the Royal Society A, 400*, 97–117.

Gandy, R. (1980). Church's thesis and principles for mechanisms. In J. Barwise, H. J. Keisler, & K. Kunen (Eds.), *The Kleene symposium* (pp. 123–148). Dordrecht: North Holland.

Gandy, R. (1993). *On the impossibility of using analogue machines to calculate non-computable functions*. Unpublished.

Giunti, M. (1997). *Computation, dynamics and cognition*. Oxford: Oxford University Press.

Herken, R. (1995). *The Universal Turing machine: A half-century survey*. Wien/New York: Springer.

Horsman, C., Stepney, S., Wagner, R., & Kendon, V. (2014). When does a physical system compute? *Proceeding of the Royal Society A, 470*(2169), 20140182.

Jarrett, J. P. (1984). On the physical significance of the locality conditions in Bell argument. *Noûs, 18*, 569–589.

Kieu, T. D. (2002). Quantum hypercomputation. *Minds and Machines, 12*(4), 461–502.

Kleene, S. C. (1952). *Introduction to metamathematics*. Princeton: Van Nostrand.

Kripke, S. A. (2013). The Church-Turing 'Thesis' as a special corollary of Gödel's completeness theorem. In B. J. Copeland, C. J. Posy, & O. Shagrir (Eds.), *Computability: Turing, Gödel, Church, and Beyond* (pp. 77–104). Cambridge: MIT Press.

Maudlin, T. (1994). *Quantum non-locality and relativity*. Oxford: Blackwell.

Mendelson, E. (1990). Second thoughts about Church's thesis and mathematical proof's. *The Journal of Philosophy, 87*(5), 225–233.

Olszewski, A., Wolenski, J., & Janusz, R. (2007). *Church's thesis after 70 years*. Frankfurt: Ontos Verlag.

Piccinini, G. (2012). *Computation in physical systems. The Stanford encyclopedia of philosophy*. http://plato.stanford.edu/archives/fall2012/entries/computation-physicalsystems. Accessed 5 Dec 2013.

Putnam, H. (1988). *Representation and reality*. Cambridge: MIT Press.

Rovelli, C. (2004). *Quantum gravity*. Cambridge: Cambridge University Press.

Scheutz, M. (1999). When physical systems realize functions.... *Minds and Machines, 9*(2), 161–196.

Scheutz, M. (2001). *What is not to implement a computation: A critical analysis of Chalmers' notion of implementation*. http://hrilab.tufts.edu/publications/scheutzcogsci12chalmers.pdf. Accessed 5 Dec 2013.

Shagrir, O. (1997). Two dogmas of computationalism. *Minds and Machines, 7*(3), 321–344.

Shagrir, O. (2002). Effective computation by humans and machines. *Minds and Machines, 12*(2), 221–240.

Shagrir, O., & Pitowsky, I. (2003). Physical hypercomputation and the Church-Turing Thesis. *Minds and Machines, 13*(1), 87–101.

Sieg, W. (2002). Calculations by man & machine: Mathematical presentation. In *Synthese series, proceedings of the Cracow international congress of logic, methodology and philosophy of science* (pp. 245–260). Dordrecht: Kluwer Academic Publishers.

Sieg, W. (2008). Church without dogma: Axioms for computability. In S. B. Cooper, B. Löwe, & A. Sorbi (Eds.), *New computational paradigms: Changing conceptions of what is computable* (pp. 139–152). Berlin: Springer.

Sieg, W., & Byrnes, J. (1999). An abstract model for parallel computations. Gandy's thesis. *The Monist, 82*(1), 150–164.

Simons, P. (1987). *Parts*. Oxford: Clarendon.

Soare, R. I. (1996). Computability and recursion. *The Bulletin of Symbolic Logic, 2*(3), 284–321.

Stabler, E. P., Jr. (1987). Kripke on functionalism and automata. *Synthese, 70*(1), 1–22.

Syropoulus, A. (2008). *Hypercomputation: Computing beyond Church-Turing barrier*. New York: Springer.

Tamburrini, G. (2002). *I matematici e le macchine intelligenti*. Milano: Bruno Mondadori.

Turing, A. M. (1936). On computable numbers with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, 42*(2), 230–265. A correction in 43(2) (1937), (pp. 544–546). Oxford: Oxford University Press.

Chapter 4
A Refutation of the Church-Turing Thesis According to Some Interpretation of What the Thesis Says

Doukas Kapantaïs

Abstract I present a method by which an idealized human calculator can compute the values of the original Ackermann function that is both effective and mechanical, and which no Turing Machine, or calculator isomorphic to a Turing Machine, can successfully bring to an end. Since the algorithm used for this computation is not Turing Machine reducible, neither is it Turing Machine describable, or describable by an equivalent formalism. That it is effective and mechanical, however, is intuitively obvious. I draw two conclusions. First, that there are calculators that are not isomorphic to Turing Machines and, so, the Church-Turing thesis, according to at least one interpretation, fails. The interpretation that is challenged is that all maximal models of computation are equivalent up to isomorphism to Turing Machines. Second, I draw the conclusion that the human brain/mind is not equivalent up to isomorphism to Turing Machines. Finally, I put forward a conjecture about why this might be so.

Keywords Church-Turing thesis • Turing Machine isomorphism • Ackermann function • Abstract state machines • Recursive way of thinking

4.1 What a Function Is

What is a "function"? In essence, what a function is comes down to this: Something, an item, the so-called "argument" of the function, is substituted by another (not necessarily different) thing, item. What performs this substitution is the function itself. The "not necessarily different" clause above is suggestive and not contradictory with respect to the verb "is substituted" it depends upon. For the important thing, as far as functions are concerned, is only this: they take some arguments as their input and they return some values as their output. So, the function is an operation

D. Kapantaïs (✉)
Academy of Athens, Research Centre for Greek Philosophy, Anagnostopoulou 14,
10673 Athens, Greece
e-mail: dkapa@academyofathens.gr

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_4

that operates on items (trivially, on what else could an operation operate upon?) and produces items. The set of items, from which the function takes the items it operates upon, is called the "domain of arguments", and the set of items, from which the function takes the items that result from the operation, is called "the domain of values". It is not compulsory that the operation is fruitful for all arguments. Some functions produce no result, when given some argument values. These functions are called "partial". The rest are called "total".

You can easily –and safely, from the formal point of view– make an anthropomorphic model of what a function is. You can imagine it as a person to whom you give some items and who, then, returns some items back.

So far we've only said that functions operate on items and return items. A natural question is what kind of items these items might be. The straightforward answer is: "any kind". Provided they satisfy suitable identity criteria, people, atoms, streets, sets, numbers, functions, intensions, feelings –you name it!– might serve as the arguments or values of functions.

The way we have presented things thus far suggests that we consider functions to be items in their own right, and so, since no item can be an argument/value of a function, unless it satisfies appropriate identity criteria, we now ask: What kind of criteria are these with respect to functions? The criteria we will employ are these: two functions will be identical, if and only if they return the same values for same arguments. This is called an "extensional" criterion, since it ignores the way the function arrives at its values. The only thing that matters is that, on being given these specific arguments, it provides these specific values.

Seen thus, functions exist "out there", along with trees, human beings, sets, numbers and any other item that belongs to our world. Upon the same extensional criterion, functions can ultimately be reduced to sets of ordered pairs.

Functions from natural numbers to natural numbers are called "numeric functions".

4.2 What an "Effective Calculation" of a (Numeric) Function Is

What can one know about numeric functions? Obviously, one cannot grasp a function having an infinite domain of arguments by making a list of all the pairs it consists in. Fortunately, there is an alternative way to grasp functions – namely, by finding a method that produces the value of the function for any arbitrary element of its domain of arguments. That is to say that, instead of a list, one can opt for a method such that, for any argument x of the domain of arguments of the function f, the method finds the value that f provides for x, i.e. f(x). In order to remain faithful to our previous extensional criterion of identity, let us add that the exact nature of the method is of no importance. If there are several methods of coming forward with the value that function f provides for an arbitrary argument, grasping any one of these will equally count as grasping the function. However, for reasons made clear by the following example, it is necessary to impose some restrictions upon what counts as "a method" for calculating the values of a function.

Assume some effective codification of the formulae of Peano Arithmetic by natural numbers. Now, imagine a person who claims to have found a method for computing the following function: as another person loudly counts from zero onwards, this person utters "zero", when the coded formula, if any, is true, and "one", otherwise. Obviously, what this person is doing consists in the unraveling of some numeric function[1]; her doings correspond to the gradual formation of some infinite set of ordered pairs. But are we allowed to say that she is actually "calculating" the values of this function? (I.e. are we allowed to say this even though she is perhaps simply throwing a dice and being miraculously lucky?) Well, if the Church-Turing thesis[2] is true, the same person must partly be an Oracle, and so –according to our terminological preferences, at least– we are not allowed to say that. This, of course, does not undermine the logical possibility of the state of affairs the above thought experiment represents. There is nothing contradictory in the assumption that such a test is successfully passed by some candidate.[3] However, the success of this candidate, *if* the Church-Turing thesis is true, cannot be credited to her calculating capabilities.

I stress this point because there is an ongoing discussion in the literature[4] that, in my view, puts too much emphasis on the adjective "effective" within the expression "effectively calculable". That is to say that, some authors tend to think that, when Church proposed the identification of recursive functions with the effectively calculable ones, he was allowing for the existence of some *calculable* functions such that, by (i) not being "effectively calculable", fall outside the scope of "Turing Machine computable", but (ii) are calculable nonetheless. The thought experiment I presented in the previous paragraph suggests that, although there might be Machines that calculate these functions, their existence is not compatible with the Church-Turing thesis. For, as I see things, either the lady of my thought experiment has found some non-recursive method of calculating the function she unravels, and, so, the Church-Turing thesis is false, or she is not calculating at all. I do not know whether this is just a terminological point or there is more to it. The considerations of the following part suggest that there is more to it,[5] but, if it is just a terminological point, it amounts to saying that, in this paper, "calculable", as well as "computable"

[1] Namely, f(x) = 0, if x is the code of a true arithmetical formula, and f(x) = 1 otherwise.

[2] In its formulation by Church (1936): "We now define the notion, already discussed, of an *effectively calculable* function of positive integers by identifying it with the notion of a recursive function of positive integers (or of a λ-definable function of positive integers)".

[3] Although, if the Church-Thesis is true, there is no effective way for anyone to recognize that the same person is unravelling this function, since it corresponds to the unravelling of True Arithmetic.

[4] In Copeland (1997), one can find a thorough presentation of the aspects of this debate with the corresponding bibliography.

[5] See esp. the tension between INT1 and INT2 in the next part.

will be coextensive with “effectively calculable” and “effectively computable”; the idea being that a function that cannot be effectively calculated/computed is not calculable/computable at all.[6]

4.3 What the Church-Turing Thesis Says

I now present three different interpretations of the Church-Turing thesis and single out the one I believe that my paper refutes.

In Church’s (1936) initial formulation, the Thesis consists in the claim that the class of effectively calculable numeric functions is identical to the class of recursive numeric functions, which is also identical to the class of λ-definable numeric functions. Turing (1936) proved that the latter two are identical to the class of Turing Machine computable numeric functions, and, like Church, he assumed that they all coincide with the effectively calculable numeric functions (although, unlike Church, he did not confine the domains of functions to numeric ones). Among the notions appearing in all these identity statements the only one that is not formal is the notion of “effectively calculable numeric function”.

Several alternative ways of interpreting the Thesis have been suggested. I will now classify them under three major categories/interpretations.

INT1. The Church-Turing thesis is in reality a definition. It has been proved that there is an idealized calculator with huge computational powers (i.e. the Turing Machine) and a family of equipotent machines[7] and formalisms. We should all agree to *name* the functions these machines and formalisms can compute “effectively calculable functions”.

INT2. The Church-Turing thesis is in fact a conjecture. After the discovery of this idealized calculator with these huge computational powers, and the discovery of a family of other equipotent machines and formalisms, the conviction grew that these machines/formalisms really exhaust the computational powers in general. Now, if they really do so, anything that can be effectively calculated can be computed by a Turing Machine.

INT3. The Church-Turing thesis is the conjecture according to which not only anything that can be calculated can be computed by a Turing Machine, but also any formalism or machine that is equipotent to a Turing Machine is equivalent to it up to isomorphism. It comes down to the conjecture that all these maximal computational systems and machines share the same abstract structure, and so they do not only have the same computational power but, which is more, they compute in the same abstract way.

[6] E.g. think of a Turing Machine with an Oracle. In the literature, it is often said that such a Machine calculates, albeit not effectively. We will here say that it does not calculate. Readers who prefer the former idiolect should systematically read, “effectively calculate” instead of “calculate” in what follows.

[7] I occasionally write “Machine” for “Turing Machine”; “machine” will mean: either Turing Machine, or Post Machine, or Gandy Machine, etc.

Were one to put INT1 to INT3 into slogans, INT1 would be: "Turing Machine computable functions are called 'effectively calculable'", INT2: "No numeric function that can be calculated cannot be computed by a Turing Machine", and INT3: "All maximal models of computation share the same abstract structure with Turing Machines."

In what follows, I will be referring to the Church-Turing thesis as the thesis behind INT3, an interpretation well attested in the literature (see below). Notice that, even in case my refutation of INT3 turns out to be effective, it will not count as a refutation of the weaker INT2. The way one calculates the values of the Ackermann function is of no importance for INT2. A legitimate counterexample to INT2 would have to be one that presents a mechanical/effective way to calculate some non-recursive function. My example of a non Turing-Machine-translatable way to calculate a recursive function will not do. Notice also that my example does not count as a refutation of INT1 either. According to INT1, Church and Turing have not *proposed* a tentative candidate for the formal counterpart of "effectively calculable". After having arrived at some formal definition of a family of computations, they have *named* the functions these computations compute "effectively calculable". So, even if some computation is found such that it computes some function other than these, this computation would *by definition* not be a computation of an "effectively calculable" function. And this no matter how effective, reliable, repeatable, mechanical, etc., would this computation possibly be.

A few more comments on these interpretations, before moving onto the argument proper.

1. INT1 is hardly credible. To my mind, it represents a desperate effort to make the Thesis *a priori* irrefutable. If "effectively calculable" is *the name* the founding fathers of the theory of computation have decided to give to some functions, then, there can be no genuine debate over the Thesis. A more interesting way to see through INT1 comes from a comment of Post, according to which Church has masked an (informative) identification under a definition (Post (1936), p. 105). Seen that way, INT1 is supposed to be upheld by people who are not as unsophisticated as to launch a war over the defense of a non informative (Russell-Whitehead) definition, but are cunningly presenting a thesis/claim under the guise of a definition. This amounts in saying that in essence these people do not uphold that the functions that are Turing Machine computable are just *called* "effectively calculable", but that "effectively calculable" is indeed coextensive with computable *simpliciter*. The latter claim, an empirical conjecture in essence, is for dialectical/strategic reasons presented as a non informative, and thereby non falsifiable, definition. Interesting as this might be, I think that none among Church's formulations suggests such a cunning underlying scheme.[8]

[8] Although Church uses the verb "to define", this does not mean that he had Russell-Whitehead definitions in mind. A scientific trial-like definition seems much more likely, since, as he says, he "proposes" it, and one can hardly "propose" a name or an abbreviation; at least not as a factual conjecture.

2. INT2 deserves the name of "a thesis" much more than INT1. It asserts that, if anything can be calculated at all, it can be computed by a Turing Machine. This, obviously, is no definition but a working hypothesis, just as Post would like it to be. Moreover, there seems to be some considerable empirical evidence for the thesis INT2 stands for. This is because all interesting formalisms/machines that have been found after the initial discovery of general recursive functions, λ-calculus and Turing Machines[9] are also equipotent with the former. If there are no miraculous coincidences, then one must find a way to account for this phenomenon. Why not try: "Because they all exhaust the limits of computation". Of course, all this remains at the level of additional evidence and is not a proof. As we will see, some version of INT3 embarks at such an effort.
3. INT3 goes beyond INT2 in the following two respects. First, it tries to provide an explanation of the otherwise mysterious co-extensiveness of this family of formalisms and machines. For to say that they are all equipotent, because they exhaust the limits of computation, is not an explanation of *why* they exhaust the limits of computation; it's just some further evidence that they do so. On the other hand, bringing forward what they themselves share in common (i.e. a common structure) is much more ambitious. For, after having brought forward what these formalisms/machines share in common, one could further suppose that any formalism/machine that exhausts the limits of computation shares this element in common too. Notice here that, if this last supposition turns out to be correct, one would also have a *formal* proof of the Thesis. For consider it this way. *Prima facie*, the Church-Turing thesis cannot be formally proved, since it claims that a non-formal item (i.e. "effectively calculable") is identical with a formal one (i.e. "Turing Machine computable"). Now, there can be no formal proof of any identity statement relating a formal and a non-formal item. The only identity statements that can be formally proved are statements relating items within a formal language of a theory. So, suppose that some explanation has been provided as for why these formalisms/machines are equipotent. If this explanation consists in the finding of yet another formal item, i.e. their common structure, and you further assume that any formalism/machine that exhausts the limits of computation must be also characterized by this item then, what you are actually doing is proposing a formal interpretation for the "effectively calculable". I.e. you do not only prove that all these formalisms/machines share a formal element in common, you further propose that this element is shared by all formalisms and machines that exhaust the limits of computation. So, you can now formally define "effectively calculable" through this.

An enterprise of this sort has been undertaken in Dershowitz and Gurevich (2008).[10] What these authors did is the following. On the one hand, they have proposed a specific axiomatization as the formal counterpart of "effectively calcula-

[9] E.g. Post's, Kolmogorov's, Gandy's, etc.

[10] See also Boker and Dershowitz (2008).

ble".On the other, they have proved that all the formalisms/machines of the second part of the equivalence are interpretations of this axiomatization.[11] So, in case this axiomatization really captures the informal notion of "effectively calculable", then, their proof must also be a formal proof of the Thesis. Additionally, they have informally argued that this axiomatization must indeed be capturing the informal notion of "effectively calculable", for, in order not to be capturing it, one would need to be able to imagine a computational method that is both effective and, at the same time, falsifies at least one among the four Postulates of their system.[12] So, now, the burden of (dis)proof is on the opponent, who must either argue against these Postulates directly, or come forward with an effective computation that cannot be translated *salva* isomorphism into a computation of a machine with this abstract structure.

In what follows, I will do the latter by indicating a certain way of mechanically calculating the original Ackermann function that we, humans, can perform and that cannot be mimicked by any Turing Machine. If this is exact, the isomorphism between us, as calculators, and these formalisms/machines fails.

More precisely, I will claim that there are some updates in this particular way of computing the original Ackermann function that have no isomorphic counterparts in any computation of the same function as performed by these machines. This implies not only that the set of Postulates of Dershowitz and Gurevich need to be loosened in order to be able to capture "effectively calculable", but also that INT3 is false, since the same Postulates provably capture the abstract structure behind Turing Machines and equipotent machines and formalisms.[13]

[11] This is a tightening of the Abstract State Machines in Gurevich (2000). I will not enter into the details of the system itself, since I will focus on the philosophical consequences of the main proof in their paper. I take it for granted that the same proof is formally valid. More precisely, I will claim that this is a valid proof relating two formal items, but that the item the authors think it is the formal counterpart of "effectively calculable" is not.

[12] The fact that the defense they have made of the claim that their axiomatization really captures "effectively calculable" is informal represents no setback to the general project. For, even if one can propose a formalization of a non formal notion, there can be no formal proof that this formalization captures what the non formal notion speaks about.

[13] Through a more historical perspective, and if our focus of attention is not the specific formulations of the Thesis but what Church and Turing actually had in mind, this means that, if they had in mind something along the lines of Dershowitz and Gurevich (2008), then the proof of the latter paper, combined with our purported counterexample, instead of a proof of the Thesis, is in fact a refutation of it.

4.4 A Non Turing Machine Equivalent Way to Calculate the Ackermann Function

Task: calculate A(n,m,s) of the original Ackermann function[14] without going through any open ended search.

$$A(n,m,s) = \begin{cases} A(0,m,s) = m + s & \\ A(1,m,0) = 0 & \\ A(2,m,0) = 1 & \\ A(n,m,0) = m & \text{for } n > 2 \\ A(n,m,s) = A(n-1, A(n, m, s-1), m) & \text{for } n > 0 \text{ and } s > 0 \end{cases}$$

In "open ended searches", I include all searches made by "unbounded do until" loop programs. "Non open ended" or "bounded" searches are the ones made either by "for" loop programs involving no nested open ended search, or "bounded do until" programs.

A 'for' loop program is:

Definition 4.1
A 'for' loop program is one that initiates an iterative procedure that is repeated some given number of times (a number which is fixed before the loop is started).[15]

And a 'do until' loop program is:

Definition 4.2
A 'do until' loop program allows some process to be iterated until a given condition is satisfied.[16]

There are no operations[17] other than those containing 'for' loops, those containing 'do until' loops, and those containing both.[18] A natural question is to ask why

[14] Source: https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Ackermann_function.html. I have changed the order of the arguments for the sake of the presentation.

[15] Cf. Smith (2007), p. 89.

[16] Cf. Smith (2007), p. 91.

[17] I follow Kleene in calling an operation: "The change from the initial situation to the terminal situation (when there is one) performed by the Machine", Kleene (1952), p. 358.

[18] "Do until" loop programs can be further distinguished in "while" and "do while" but this distinction changes nothing in the argument of this paper.

these methods do not occasionally overlap. For one might think that a *bounded* 'do until' program can also be seen as a 'for' loop program, since it imposes an upper bound on the number of iterations. Although this last remark is correct, the programs are essentially different. The bound might be there in both, but it remains the case that a bounded 'do until' program is not a 'for' loop program, for in a bounded 'do until' program the Machine is told to execute a certain routine a certain number of times and stop *as soon as it finds the desired value*. So, in 'do until' programs, we do not know beforehand the exact number of iterations that are needed (although in *bounded* 'do until' programs we know that they cannot exceed a certain number). The Machine tests its successive updates, and accepts or rejects them. As long as it rejects them, the calculation goes on to the next update. In contrast, 'for' loop programs determine beforehand the *exact* number of iterations needed and print the final update, without any test. They –so to speak– "know beforehand" that this update represents the desired value.

If there are no general programming routines other than the ones involving 'for' loops and 'do until' loops, asking the calculator to calculate A(n,m,s) of the original Ackermann function without going through any open ended search is to ask the calculator to employ only 'for' loops that involve no nested 'do until' loops (henceforth, simply "'for' loops") and/or 'bounded do until' loops.

Here is a way to mechanically[19] do this, in case the calculator is an idealized[20] human being:

> Unless s = 0, where the routine is obvious, I construct the nth function (counting from the 0th) upon the sequence: addition, multiplication, exponentiation,..., i.e. upon the Knuth up-arrow notation hierarchy. That is to say that I 'feed myself' with addition and I 'output' multiplication. Then, having multiplication as the update, I 'output' exponentiation, and so on until the nth iteration. Call the function corresponding to the nth output 'An'. Since An is primitive recursive,[21] there is a 'for' loop algorithm for the computation of its values. I implement this algorithm for computing An(m,s) = A(n,m,s).

This way of calculating A(n,m,s) breaks the calculation down to two successive no open ended searches.

In step (1), I use a recursive function from numbers (or numerals, if you prefer) to functions (or programs, if you prefer). This way of constructing functions/programs

[19] A "mechanical way to perform a calculation" will be a way such that: 1. It is set out in terms of a finite number of exact instructions (each instruction being expressed by means of a finite number of symbols), 2. If carried out without error, produces the desired result in a finite number of steps, 3. It can (in practice or in principle) be carried out by a human being unaided by any machinery save paper and pencil, 4. It demands no insight or ingenuity on the part of the human being carrying it out. (From Copeland (1997)). For further support of the claim that the following is a "mechanical procedure" see the Appendix.

[20] By "idealized" I mean: 1. Having an endless amount of time at her disposal, 2. Never falling victim of her own carelessness and never been prevented from fulfilling her tasks by physical hindrances, 3. Being capable of storing an endless amount of data, while calculating, and 4. Being able to use any of these data, though perhaps not all at once.

[21] Notice that all functions upon the Knuth up-arrow notation hierarchy can be defined by other primitive recursive functions and composition.

is both recursive and mechanical. It is recursive in the "generalized" sense of Hilbert's "On the Infinite", for, according to that sense, recursion "rests upon the general principle that the value of the function for a value of the variable is determined by the preceding values of the variable".[22] Clearly, the function of step (1) takes a natural number as the argument and returns a function in a way that stays obedient to Hilbert's formulation. The value for argument n depends on the value for argument $n - 1$. Moreover, this is definitely a mechanical method. First, I take for granted that one knows what the specific (initial) value for argument 0 consists in. Addition is a numerical function one can handle and recognize in every context. Moreover, there is a mechanical way of constructing any value for all arguments $n > 0$. Just use the following recipe:

> Provided that the value for $n = 0$ is already given, the human calculator can unmistakably learn how to handle the function corresponding to any n upon the hierarchy in n successive steps that involve not any hidden 'do until' command. The method for outputting (mastering) the function for any n upon the sequence comes down to the following guidance: The function for argument n is a two arguments function, such that, in order to arrive at An(m,s), iteratively perform $s - 1$ many times the operation corresponding to the function for argument $n - 1$ to arguments m and m. ("Iteratively" meaning here: operate the function for argument $n - 1$ to m and m; then, operate it to the result of the first operation and m; then, to the result of this operation and m, and so on until $s - 1$.) So, our human calculator (provided that she knows addition) can arrive at handling the function corresponding to any n. If $n = 1$, she follows the guidance just once. Then, she knows multiplication. If $n = 2$, she follows the same guidance once more (i.e. now that she knows multiplication, she follows the guidance by applying it to multiplication), and so on for any n. Now, if we call n, "the input", and the nth function upon the Knuth up-arrow notation hierarchy, "the appropriate output", this method can be seen as a way to mechanically and effectively arrive at the appropriate output in exactly n steps that involve not any 'do until' command.[23]

Obviously, this procedure for arriving at function An involves no intuition whatsoever and, provided that the provisos of note 20 are satisfied, we have no reason to disbelieve that, if one can handle addition, and one understands the above recipe, one can eventually handle any function upon the sequence. (See also the Appendix.) Any idealized human being can pass the rest of eternity by routinely grasping what each function upon the hierarchy does. Notice that there is not any "magic" involved in constructing/grasping any of these functions, if its predecessor is already available. On the contrary, because of the essentially recursive character of the recipe, it would have been some "magic" involved, only if there were some threshold, upon the hierarchy, beyond which our idealized human calculator could not go on. The procedure for stage n is, in principle, exactly the same as the procedure for stage $n - 1$. So, if one can mechanically construct multiplication upon addition, one can, in principle, mechanically construct any function upon the hierarchy; as with addition and multiplication, so, in principle, with any An

[22] Hilbert (1925) in Van Heijenoort (1967), p. 386.

[23] It is essential here not to confuse the function we are now describing and which takes numbers as inputs and outputs functions with the Ackermann function itself, which takes numbers as inputs and outputs numbers.

and $An-1$. Finally, notice that, while one is constructing the functions upon this sequence, one is under a ‘for’ loop program. I.e. one knows exactly how many iterations one needs to go through in order to reach the desired value, and just allows oneself to do the mechanical building of each intermediate value by repeating the same routine over and over again from the first function of the sequence (i.e. addition) to the one she is occasionally after. For anyone that knows addition can learn how to multiply, if one explains to her that multiplication with arguments x and y consists in adding $y-1$ many times the first argument: first to itself, then to the sum that results from this first addition, then to the sum of this second addition, and so on so forth. Similarly, anyone can master exponentiation, if one explains to her that exponentiation with arguments x and y consists in multiplying $y-1$ many times the first argument: first with itself, then with this first product, then with the following product, and so on so forth. But also, anyone can realize that there is a pattern there, and that one, starting with addition can build/grasp recursively any function upon the hierarchy. The entire procedure consists in an initial argument (addition) that is iteratively updated in the exact same manner a precise number of times. No ‘do until’ command in there at all. So, no open ended search thus far.

Neither does step (2) require any open ended search. All functions upon the sequence are primitive recursive and, so, there is a ‘for’ loop program for their computation. There is, moreover, no doubt about the mechanical nature of step (2) either.

As opposed to this way of calculating the function, the corresponding “natural” ways for the Machine to proceed are two. First, it can go through the double (arithmetical) recursion involved in the function, until it reaches the desired value. This way definitely involves ‘do until’ commands. Second, it can run a ‘do until’ loop program by checking all numbers from 0 on, to see whether they are A(n,m,s) or not. This is also effective, since the relation $A(n,m,s)=y$ is primitive recursive, and, so, one will eventually receive an answer for any input. But, here again, the search will not be bounded, since the function itself is not primitive recursive, and, so, there is no possible way for one to situate an upper bound, below which the value for arbitrary arguments n,m,s is to be found. Hence, both these methods involve unbounded searches.

Of course, an idealized human being can execute any program a Turing Machine can, and, so, can follow both these methods. But can a Machine mimic the human method? The answer is negative.

To begin with, we have to notice that the Machine cannot handle functions from numbers to functions directly. In general, the Machine cannot handle functions having domains other than numeric ones. As a matter of fact, this is not exactly right either, for *stricto sensu* the only thing a, e.g., Turing Machine can do is manipulation of strings,[24] which, for convenience, we will assume here that they are strings of

[24] *Mutatis mutandis* for the rest of machines and formalisms (e.g. they manipulate graphs, sets, formulas, etc.).

(arabic) numerals. Now, in an initial interpretation of these strings, these strings have been interpreted as natural numbers, and it is in that derivative sense that we now say that the Machine can handle functions from numbers to numbers. But, even in this derivative sense, it initially looks like that the function we have presented, and which has natural numbers as arguments and functions as values cannot fit in. However, this is not an insurmountable obstacle, for all the functions upon the Knuth up-arrow notation hierarchy are primitive recursive and, as such, can be mapped onto the natural numbers. Consequently, and while the strings the Machine manipulates will keep being interpreted as natural numbers, these natural numbers will now be codifying both the natural numbers and these functions. Now, there needs to be a numeric function f, such that, on input the natural number n, returns the natural number [An].[25] If there is any chance for the Machine to successfully mimic what the human calculator was doing, this function needs to be computable by the Machine. Let us, then, assume that this function is computable by the Machine, and try to imagine how the Machine tries to mimic the human calculator. In so doing, it cuts the calculation of A(n,m,s) down to two stages. In the first, it computes [A(n)], and then it computes [An(m,s)]. The problem with this is that the Machine cannot perform it by two consecutive bounded searches.[26]

Proof Assume that f(n) = [An] is primitive recursive. Consider the function g(f(n),m,s) with the following characteristics. g(f(n),m,s) finds within f(n)[27] the codes of the two free variables of An and substitutes them by the codes of its own two arguments, m and s. (E.g. g(f(n),2,3) results from [An], when [m] is substituted by [2], and [s] by [3].) Obviously, g(f(n),m,s) is meant to represent A(n(m,s)). Now, if f(n) is primitive recursive, g(f(n),m,s) is primitive recursive too, for notice that, apart from f(n), which we assume to be primitive recursive, the rest is primitive recursive as well.[28] Since there is a way to calculate an upper bound for the values of all primitive recursive functions, there is a way to calculate an upper bound for the values to the variables n,m,s of g(f(n),m,s) too. However, g(f(n),m,s) = [A(n,m,s)], and, so, there is a way for the Turing Machine to calculate an upper bound for the values of the Ackermann function. Therefore, f(n) is not primitive recursive. If f(n) is calculable by the Machine but not primitive recursive, f(n) must be recursive and involving some μ-minimization. Therefore, in order to find f(n), the Machine must

[25] I will use "[$]" for the code for $ (numeric code here).

[26] Remember what the human calculator was doing. She had, first, put some effort in order to find/learn An, through n consecutive steps involving no 'do until' commands, and, then, she proceeded with the calculation of An(m,s).

[27] Some ambiguities between function symbols and value-of-function symbols are unavoidable. (They would not have been, had we used λ-calculus notation.) I hope that they are all decidable by the context.

[28] For example. Assume a Gödelian encoding. Prime factorization is p.r., searching for specific exponents within the codes of p.r. functions is p.r., substituting exponents by other exponents is p.r.

go through an open ended search. Since a stage in the calculation of g(f(n),m,s) involves an open ended search, the entire calculation involves an open ended search. □

4.5 Conclusions

What the above proof establishes is that the Machine cannot mimic the way the human calculator finds An in the following essential respect. The human calculator launches a program, implementing an algorithm such that it builds recursively the function An by n consecutive updates of A0 that involve no ‘do until’ commands. Had the operation of the Machine and of the human calculator been isomorphic, there should be a translation function such that it takes each element of the stages the human calculator goes through, while computing An, to the stages the Machine goes through, while computing f(n), and this translation would have left the abstract algorithm they both implement, by their distinct programs, intact. Suppose that there is such a translation function τ. Obviously, $\tau(A0) = [A0]$, $\tau(A1) = [A1], \ldots$ Say that “u” denotes the update function upon states during a calculation. What is missing for the isomorphism to be preserved is $\tau(u(Ax))$. For example, u(Ax) is $Ax + 1$, but there can be no $u([Ax])=[Ax+1]$, with a structurally identical update function. There is simply is no way for the Machine to update [Ax] in a way that preserves isomorphism.[29] That is to say that, there can be no translation *of the way* the human calculator updates states along her way to An to a similar way that the Machine updates states along its way to [An]. The particular details as for how exactly the Machine arrives at [An] are of no importance. The “dumbest” but still effective way would be to check all numbers from 0 onwards and see whether they are [An] or not. If $f(x) = y$ is decidable, the Machine will eventually stop at [An]. One can imagine several interesting shortcuts, but, still, there would be no program available to the Machine in order to calculate an upper bound for (or to calculate the exact number of) the updates it needs for reaching [An], and, so, the way it updates its states must be structurally different.[30]

Notice that this is stronger than saying that the Turing Machine as a calculator and the human being as a calculator differ. After all, a one dimensional Turing Machine also differs from a two dimensional Turing Machine, and both differ from a Post Machine. However, these machines are interpretations of the same abstract

[29] Of course, one can effectively program a Machine to recognize and print the codes of A0, A1, ..., An, in a second printer along its way to [An]. One might even program it not to print anything else (i.e. to do the rest of its calculations in the dark). But this is not what the isomorphism requires, because, if one does so, ones loads a supplementary program to the Machine, other than the one it executes in order to arrive at [An].

[30] See the Appendix for a graphic juxtaposition of the “human” program with the inoperable “machine” program that tries to mimic it.

structure, and, so, their calculations can be translated into one another. Strings are strings, graphs are graphs, sets are sets, but this variety reflects no structural difference.

On the other hand, the difference between the idealized human calculator and the Turing Machine is structural, and, so:

(i) Not all maximal models of mechanical computation are equivalent up to isomorphism to Turing Machines (This is a refutation of the Church-Turing thesis in the form of INT3).

And:

(ii) The human brain/mind is not equivalent up to isomorphism to Turing Machines. (Notice that the counterexample to the Turing Machine was an algorithm implemented by the idealized human calculator.)

4.6 An Explanatory Conjecture

If I may allow myself a final conjecture here, the reason why the Machine cannot recursively construct f(n) upon f(0), in the way our human calculator can, could possibly be that it cannot help itself further in its computation by providing a many sorted domain for the items it manipulates. For consider it this way. What a Turing Machine does consists in string manipulation. In an initial stage, we interpret strings as numbers, but, from then on, any computation the Machine performs keeps being a computation within this homogenous domain. So, as far as the Machine is concerned, the semantically simple function from numbers to numbers and the (semantically) more complex one from numbers to functions collapse into a single domain. For example, what the Machine does when asked to add, say, 3–5 and what it does when asked to reach the (code of the) nth function upon the Knuth up-arrow notation hierarchy is exactly the same. It is to perform some operation and transform, thereby, a string into another. (Following the initial interpretation of the strings, one should say “a number into (another) number”.) In that respect, the Machine is “semantically bind” with respect to the items it operates upon, or, at best, it cannot generically discern them; they are all numbers.[31]

I suggest that the reason why the human calculator can construct An with a ‘for’ loop program, while the Machine cannot compute f(n) with a similar program, is that, unlike the Machine, the human calculator can appreciate the function she calculates as a function from numbers to functions, and not as a function from numbers to numbers. In other words, it is possible that the confidence the same calculator has that she will eventually “reach” the function she is after, after exactly n recursive steps, is due to the fact that she treats these steps as operations on

[31] Even in case it can represent “functions”, this is only through their numeric codes.

functions or, if you prefer, programs; i.e. on programs capable of calculating the values of these functions. The Machine, in contrast, operates only on numbers, and the result of these operations are numbers as well. Therefore, the bare algorithmic complexity of some particular kind of arithmetic operations might be such that it prevents the Machine from being able to calculate any upper bound for them.[32] Now, if the Machine were able to single out the set of numbers of the sequence [A0], [A1],..., in an isomorphic way with the one that we, humans, can single out the set of functions upon the Knuth up-arrow notation hierarchy, then, I see no reason why the Machine would not be able to reach [An] by a 'for' loop program. But, provably, the Machine cannot do that, and, so, there must be something that gets "lost in translation" during the whole mapping procedure (i.e. the one of functions to numbers). Possibly, the human calculator has an advantage over the machine, because the human calculator can make sense of a fundamentally non homogenous domain, of a domain, that is, where some items are numbers, some items are functions and the latter are not ultimately reducible to the former. This would imply that "reaching An" is not ultimately reducible to "reaching [An]", and that the two operations are fundamentally incomparable.

Acknowledgments Special thanks to Nicola Angius, Douglas Campbell, Nachum Dershowitz, Marie Dimarongkona, Florent Franchette, Petros Stephaneas for some useful comments. My gratitude goes also to Evagelia Ksinopoulou, without the aid of which my presentation in the Conference would not have been possible in the first place. Also thanks to an anonymous referee of *Synthese Library* for putting some pressure on the description of the "human" program.

Appendix

The 'for' loop "human" program for the calculation of A(n,m,s) and its non isomorphic approximation by the Machine

```
Stage 1 (human calculator)
for i = 1 to n
old = A0
new = φ(old)
old = new
```

```
Stage 1 (Machine, inoperable)
for i = 1 to n
old = [A0]
new = [φ(old)]
old = new
```

[32] Notice that the algorithmic complexity of the numeric function *per se* has nothing to do with the confidence our human calculator has that, because she understands addition, and because she understands how to built multiplication upon addition, she is able to go on *ad infinitum* along the Knuth up-arrow notation hierarchy.

Stage 2 (alike both for the human calculator and the Machine)

```
for i=1 to (s-1)
old=m
new=(An-1)(old,m)
    for i=1 to (m-1)
    old=m
    new=(An-2)(old,m)
                .
                    .
                        .
                            for i=1 to (m-1)
                            old=m
                            new=(An-(n-1))(old,m)
                                for i=1 to (m-1)
                                old=m
                                new=(An-n)(old,m)
                                old=new
                            old=old
                        .
                    .
                .
    old=old
old=old
```

Explanations

1. The program of Stage 1 (human calculator) is mechanical. This means that it can be executed by a human computer without any assistance from intuitions, interpretations, meanings, etc., i.e. as a mechanical manipulation of an initial meaningless input. In order to see this clearly, consider how this program can be "loaded" (taught) to an idealized human computer. (In what follows "Ax" of Stage 1 will mean: the program for function x upon the Knuth up-arrow notation

hierarchy.) At the beginning, the human computer is mechanically taught how to add. Consider this as equivalent with: being able to mechanically respond to the command "add x to y". Then, she is asked, given the numerals x, y, to iteratively perform addition on "input" <x,x>, $y-1$ many times over. This command is called "multiply x by y". Since there is nothing non mechanical in adding, and since counting is embedded in adding, there is nothing non mechanical in multiplying. So, the human computer mechanically learns how to mechanically multiply. Then, she is asked to put some Counter on 1. (Again, counting is mechanical, and, so, nothing non mechanical up to this point.) Then, on "input" <x,y>, she is commanded as follows: "Proceed with the same manipulation you have proceeded, when learning how to multiply, but, this time, instead of performing addition $y-1$ times over, perform multiplication $y-1$ times over". Since the apprentice already knows multiplication, she also knows how to use addition in order to multiply x by y. So, there is no problem with her to understand the command and to now use multiplication, instead of addition. So, she mechanically learns exponentiation. Then, she is asked to put the Counter on 2. Finally, and by a single command, she is asked not to stop until the Counter goes at some arbitrary natural number of the choice of the instructor. If she does not grasp this immediately, she is told that: "In general, and when the Counter is at z, in order for you to be allowed to move it to $z+1$, you have to prove yourself capable of manipulating <x,y> in the exact same way you have learnt how to manipulate it in order to move it to z, but, this time, upon the manipulation you have learnt, in order to be allowed to move it to z." At the end of this procedure the pupil is considered as able to construct upon addition any program for any function upon the Knuth up-arrow hierarchy. This ability is itself a program: it takes a natural number (i.e. n) as input and returns a program (i.e. the program for An) after exactly n updates of "addition".

2. The program of Stage 2 is the same both for the human calculator and the Machine. Only Stage 1 varies. For humans, it involves operations on functions, while, for the Machine, all operations are on numbers. For the Machine, Stage 1 cannot be 'do until' free, as established in the Proof of part 4, and, so, Stage 1 for the Machine is inoperable, if one wishes it to be the isomorphic counterpart of Stage 1 for the human calculator.
3. The final answer for input <n,m,s> is in the leftmost 'old' of Stage 2, after the $s-1$th loop. Each column of Stage 2 represents a different 'for' loop. As one moves from left to right, each 'for' loop embeds a decreasing amount of other such loops. 'Olds' are hereditary from right to left. I.e. when a 'for' loop is finished, the final 'old' becomes the old for the new loop of its neighbor to the left.
4. Notice that, had the Machine have an infinite (and not just inexhaustible) memory, Stage 2 alone would have sufficed for the calculation of the value for any argument <n,m,s>. This would have been, because the Machine would have been capable of storing all programs corresponding to all functions upon the Knuth up-arrow notation hierarchy and, so, it would have simply called the

"already stored" program for function Ax, for all inputs <x,y,z>.[33] Since the Machine cannot have an infinite amount of already stored data, it needs to find a general method to construct the program corresponding to any n, and, hence, the need for some Stage 1 both for us and the Machine. However, these two stages –and as the claim of this paper goes– cannot be isomorphic.

References

Boker, U., & Dershowitz, N. (2008). The Church-Turing thesis over arbitrary domains. In A. Avron, N. Derschowitz & A. Rabinovitz (Eds.), *Pillars of computer science; essays dedicated to Boris (Boaz) Trakhtenbrot…* (Lecture notes in computer science, LNCS 4800, pp. 199–229). Berlin: Springer.

Church, A. (1936). An unsolvable problem of elementary number theory. *American Journal of Mathematics, 58*, 345–363.

Copeland, B. J. (1997). The Church-Turing thesis. In E. Zalta (Main Ed.), *The Stanford encyclopedia of philosophy*. http://plato.stanford.edu

Dershowitz, N., & Gurevich, Y. (2008). A natural axiomatization of computability and proof of Church's thesis. *The Bulletin of Symbolic Logic, 14*, 299–350.

Gurevich, Y. (2000). Sequential abstract state machines capture sequential algorithms. *ACM Transactions on Computational Logic, 1*, 77–111.

Hilbert, D. (1925). On the infinite. In Van Heijenoort (1967), 369–392.

Kleene, S. C. (1952). *Introduction to metamathematics*. Amsterdam – New York: North-Holland.

Post, E. (1936). Finite combinatory processes – Formulation. *Journal of Symbolic Logic, 1*, 103–105.

Smith, P. (2007). *An introduction to Gödel's theorems*. Cambridge: Cambridge University Press.

Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, 2*(42), 230–265.

Van Heijenoort, J. (1967). *From Frege to Gödel; a source book in mathematical logic*. Cambridge: Harvard University Press.

[33] Besides, the reason that all programs with 'n' locked (i.e. each and every program for the functions upon the hierarchy) are 'do until' free is exactly this. The Machine has memory enough to store all programs upon the hierarchy up to any particular An.

Chapter 5
In What Sense Does the Brain Compute?

Paul Schweizer

Abstract I analyse the notion of computation in the physical world and argue against the widely held view that merely implementing the 'right' sort of computational procedure is sufficient to transform a given configuration of matter and energy into a genuinely mental system. Instead, I advocate a more scientifically plausible version of the Computational Theory of Mind, wherein the interpretation of the brain as a computational device should (i) provide the theoretical bridge between high level intentional states and causally efficacious physical structure, and (ii) supply the integrated key for *predicting* both future brain states viewed as implementations of abstract computational states, and output behaviour viewed in cognitive terms.

5.1 Engineered Implementation

In this chapter I will not be concerned with fine-grained details of human brain mechanics, but instead with more abstract issues pertaining to the general question – in what sense *could* viewing the brain as a computational device contribute to a scientific understanding of the mind? In order to address this question, it is first necessary to engage in an examination of the nature of computation itself, and the sense in which actual physical systems can be said to compute. This issue has proved surprisingly subtle and controversial, and yet is central to any proposed science of the mind-brain which adopts computation as an explanatory tool.

From a disembodied mathematical perspective, computation comprises an extremely well defined phenomenon. Central to the mathematical theory of computation is the intuitive notion of an effective or 'mechanical' procedure for syntax manipulation, and there are any number of different possible frameworks for filling in the details and making the idea rigorous and precise. Turing's (1936) 'automatic computing machines' (TMs), supply an intuitive and elegant rendition of the notion of an effective procedure, but there is a well known variety of

P. Schweizer (✉)
Institute for Language, Cognition and Computation, School of Informatics,
University of Edinburgh, Edinburgh, UK
e-mail: paul@inf.ed.ac.uk

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_5

alternative and provably equivalent frameworks. Turing machines (and other types of computational formalisms) are of course not concrete mechanisms but rather are *mathematical abstractions* that do not exist in real time or space. In order to perform *actual* computations, an abstract Turing machine must be realized by a suitable arrangement of matter and energy. And as Turing (1950) observed long ago, there is no privileged or unique way to do this. Like other abstract structures, Turing machines are *multiply realizable* – what unites different types of physical implementation of the same formal TM is nothing that they have in common as physical systems, but rather a structural isomorphism expressed in terms of a higher level of description.

Adopting terminology and notation introduced in Schweizer (2012), let us call this 'downward' multiple realizability, wherein, for any given abstract structure or formal procedure, this *same* structure can be implemented via an arbitrarily large number of *distinct* physical systems. The key feature here is that downward multiple realizability entails substrate neutrality. Let us denote this type of realizability as '↓MR'. After the essential foundations of the mathematical theory of computation were laid, the vital issue then became one of engineering – how to utilize state of the art technology to construct rapid and powerful physical implementations of our abstract mathematical blueprints. This is a deliberate ↓MR endeavour, involving the intentional construction of artefacts, painstakingly designed to follow the algorithms that we have specified. From this top-down perspective, there is a clear and pragmatically indispensible sense in which the hardware that we have designed and built can be said to perform computations in physical space-time.

5.2 Natural Computation?

In addition to these comparatively recent engineering achievements, but presumably still members of a single underlying category of phenomena, various authors and disciplines propound the notion of 'Natural Computation' (NC), and invoke a host of indigenous processes and occurrences as cases in point, including neural computation, DNA computing, biological evolution, molecular and membrane computing, slime mould growth, cellular automata, ant swarm optimization, etc. According to such views, computation in the physical world is not merely artificial – it is not restricted to the devices specifically designed and constructed by humans. Instead, computation is a seemingly ubiquitous feature of the natural order, and the artefacts that we have produced constitute only a very small subset of the overall class of computational systems that inhabit the physical universe.

The disciplinary and terminological practices surrounding NC plainly invite a more thorough and rigorous examination of the underlying assumptions involved. Salient questions in need of scrutiny include: To what extent, if any, is computation a genuine *natural* kind – is there any intrinsic unity or core of traits systematically held in common by the myriad of purported examples of computation in the physical world? In what sense, if any, can computation be said to take place spontaneously, as a truly native, 'bottom-up' phenomenon?

The issue has pronounced conceptual importance with respect to positions on the conjectured computational nature of *mentality and cognition*, since according to the widely embraced Computational Theory of Mind (CTM), computation (of one sort or another) is held to provide the scientific key to explaining the mind and, in principle, reproducing it artificially. The paradigm maintains that cognitive processes are essentially computational processes, and that intelligence in the physical world arises when a material system implements the appropriate kind of computational formalism. So it's an immediate corollary of CTM that the human brain must count as an exemplary instance of natural computation, and hence it is crucial to CTM's theoretical stance that there be a rigorous and precise analysis of physically grounded computation in the case of organically engendered human brains.

5.3 Three Different Senses

For the sake of conceptual clarity, it is useful to distinguish three possible (and non-exhaustive) senses in which real physical systems might be thought of as 'performing a computation'.

First (**1**), a physical system or object may be said to *obey or satisfy* a particular equation or mathematical function. For example, a falling body in the earth's gravitational field will satisfy or obey Newton's equation for gravitational acceleration. Similarly, the planets orbiting the sun satisfy or obey Kepler's laws of planetary motion. This has lead various NC enthusiasts to claim that the planets orbiting the sun, falling bodies in the earth's gravitational field, etc., are in fact *computing the values* of the equations in question. Taken to its most extreme form, this becomes the assertion that physical processes and natural laws are themselves fundamentally computational, and hence that computation constitutes the foundational key to the natural order.

In terms of the brain, sense (**1**) seems to be commonly adopted in computational neuroscience, e.g. when particular brain mechanisms or processes are said to *compute* the values of particular functions. For a salient case in point exposited by Shagrir (2014), consider the neural integrator in the oculomotor system. The scientific account given is that the system produces eye-position codes by *computing* mathematical integration over eye-velocity encoded inputs, thereby enabling the brain to move the eyes to the right position. In this case it is not the motion of some macroscopic body that satisfies a mathematical equation, but rather electrochemical states of the brain viewed as codes which satisfy the input/output values specified by the mathematical function of integration.

Second (**2**), the activities of a physical system or process may be *modelled or simulated* by a given computational rendition or depiction. For example, it is possible to create highly accurate and predictively valuable computer models which simulate the behaviour of various complex physical processes and phenomena such as earthquakes, climate change, hurricanes, particle collisions, molecular properties of materials, etc. Again, the usefulness and accuracy of these computational models has lead proponents of NC to claim that the physical phenomena *themselves* are performing such computations or are somehow instances of such computations occurring in nature. In terms of the brain, sense (**2**) is commonly adopted in neuroscience where various aspects of brain mechanics and processes are modelled computationally. Additionally, 'brain-like' connectionist architecture and artificial neural networks are used to model a host of higher level *cognitive* phenomena, such as the seminal connectionist model for past-tense acquisition of Rumelhardt and McClelland (1986). In such cases, there is usually no attempt to map these high level models directly to brain activity.

And third (**3**), a physical device or process may be said to literally *implement, realize or execute* a particular algorithm or effective procedure. Thus when I write a piece of code in some artificial programming language, say Prolog, and then run this code on my desktop computer, there is a very clear sense in which the electro-mechanical hardware is performing or executing the algorithm explicitly encoded in Prolog. In this case the computation is not a natural occurrence – rather it's a direct result of human design and engineering. However, there could also be genuine *natural* computation in the stringent sense of (**3**), since, as above, it's possible that some version of classical CTM is true. For example, if Fodor (1975) is right, then the human brain is running the Language of Thought (LOT) as an indigenous formal system of rule governed symbol manipulation, in a manner directly comparable with a computational artefact. Thus on Fodor's account, the human brain as a wetware device would count as an exemplary instance of NC in sense (**3**). And this is compatible with the foregoing discussion of downward multiple realization, since the relation between LOT and the brain could then be viewed in typical ↓MR terms. For example, an alternative mechanical device, physically quite unlike the brain, could presumably be constructed to implement the LOT in an artificial medium.

5.4 Critique of Senses (1) and (2)

As noted in sense (**1**) above, a physical system or object, such as a falling body in the earth's gravitational field or a piece of electromechanical hardware, may be described as *obeying or satisfying* various equations or mathematical functions. However, I would contend that it is under-motivated to further describe such phenomena as 'performing a computation', and the temptation to do so appears to be founded on a conflation between two distinct levels of analysis. At the most basic level, as in formal logic, functions are defined *in extension* simply as sets of ordered pairs. In extensional terms the function is a 'black box' – there is no story about the rule-like transformations from input to output. The infinite set of (nested) ordered pairs

$$\Big\{ << 0,0 >,0 >, << 0,1 >,1 >, << 0,2 >,2 >, << 0,3 >,3 >,\ldots,$$
$$<< 1,0 >,1 >, << 1,1 >,2 >, << 1,2 >,3 >, << 1,3 >,4 >,\ldots,$$
$$<< 2,0 >,2 >, << 2,1 >,3 >, << 2,2 >,4 >, << 2,3 >,5 >,\ldots \Big\}$$

may suggest addition to the average human observer, with the corresponding 'rule' being the equation $x + y = z$. This 'intensional' arithmetical component specifies a systematic transformation from binary inputs to output value, but of course there are more convoluted transformations that are extensionally equivalent while intensionally distinct. For example, the equation $(3 \cdot (x + y)) / \sqrt{9} = z$ also yields the same set but by a different method.

Similarly, for any function defined in intension by a given equation, there are any number of *different algorithms* for computing this same equation, and a mathematical function on its own does not specify a corresponding formal method or effective procedure. The equation $x + y = z$ which yields the foregoing set of pairs *could* be computed in primitive recursive function theory using the two standard Peano axioms:

$$x + 0 = x$$

$$x + \boldsymbol{s}(y) = \boldsymbol{s}\,(x + y)$$

Alternatively, addition *could* be computed using a Turing Machine program specified as a finite set of instructions in standard quadruple notation. For example, the following four state Turing Machine given by six quadruples can be interpreted as computing addition on the positive integers expressed in monadic notation:

$$\mathrm{q_1 1Bq_{1;}\ q_1BRq_{2;}\ q_2 1Rq_{2;}\ q_2B1q_3;\ q_3 1Lq_{3;}\ q_3BRq_4}.$$

Or the equation $x + y = z$ *could* be computed using Church's lambda calculus, etc.

The salient difference between what is going on in sense (**1**) as opposed to sense (**3**) is precisely the difference between the level of bare mathematical equation and the level of computational algorithm. Hence merely *satisfying an equation*, as in the case of a hardware device obeying a lawlike physical regularity, is too weak to underwrite an assertion of distinctively *computational* processing, because it leaves the vital procedural details completely unspecified. Exactly *which* algorithm for computing the values of Kepler's laws of planetary motion is the earth currently implementing? And articulated in precisely *which* abstract computational framework?

Data pertaining to physical systems, gathered via experimentation and measurement, are presented to human observers in purely extensional form, and it is the task of science to adduce mathematical regularities and characterize them with overarching equations or 'laws'. Thus a vast body of extensional measurements are given an elegant intensional characterization with the classical equation $f = m \cdot a$.

Although an accelerating mass will in fact satisfy or ‘obey’ this regularity, it is difficult to see what might be gained by further asserting that the object in question is literally *computing* the salient force. Without providing the procedural details and explicitly mapping abstract state transitions to intervening physical states mediating the inputs and outputs, attributions of computational activity to the physical system would appear to be radically underspecified.

And as in the general case, so too with the brain. It seems clear that the extensional pairs of eye-velocity inputs converted to eye-position-outputs in the case of the ocular mechanism above can be captured *intensionally* with the mathematical function for integration. But which algorithm does the brain use to *compute* the integration function, and how are the abstract state transitions entailed by the formal procedure mapped to intervening brain states? Unless these details are provided, the sense in which computation is involved remains unclear. In the case of the visual system, there is compelling reason to view the internal brain process as mirroring or calibrating itself with distal factors in order to successfully control eye position. But from this alone it does not follow that explicit computation is involved.

Perhaps it will be said that the brain is not implementing a classic digital formalism to compute the integration function, but rather is performing an *analogue computation* instead. This is not an implausible claim, and analogue constitutes yet a fourth sense in which a physical system might be thought of as ‘performing a computation’. However, we would still need to be supplied with the specific details, the actual analogue *method* by which the function is being computed in the brain. For example, the differential analyser is an analogue computer, and it works in accordance with well defined principles (Shannon 1941). Similarly in the case of purported analogue computations performed by the brain, we would still need to know the details.

In the case of sense (**2**), the procedural details missing from sense (**1**) are provided, but they are located in the wrong place. When complex physical events and processes are accurately simulated via computational models, it is the *artificial* computational structures which compute the values of the laws, equations and regularities governing the physical phenomena being simulated. And indeed, this is why the *models* are accurate and useful. But what motivates the further claim that the complex physical phenomena are *themselves* somehow implementations of the computations performed by the artificial simulations? Again, the same equations and regularities could be computed by another computational model using different underlying algorithms, programming languages, etc. to calculate the relevant values. Which of the many distinct computational possibilities is privileged or singled out by nature?

In the ensuing discussion I will treat (**3**) as the literal and canonical sense of computation in the space-time realm. I would diagnose (**1**) as derived from the mathematical characterization of regularities in nature, but where the additional attribution of computational activity is undermotivated, unless the procedural details are specified and mapped to intervening physical state transitions. Finally, (**2**) is a case of artificial *simulation* of natural events and processes, where the values of the regularities salient to sense (**1**) are explicitly computed, but where this computation is merely a tool of human heuristics and is not supported by nature. In this respect sense (**2**) is a hybrid of the more basic content involved in (**1**) and (**3**), and will not receive any further investigation.

5.5 Computation Is Non-intrinsic

What is it for a physical system to implement a computational procedure, and hence for a physical device to count as a computer in sense (**3**)? A very straightforward and elegant account, articulated by Putnam, Searle and others, is based on a simple mapping between formalism and physical structure. Accordingly, a physical system *P* performs a computation *C* just in case there is a mapping from the physical states of *P* to the abstract computational states of *C*, such that the transitions between physical states reflect the abstract state transitions as specified by the mapping.

The minimalism, neutrality and generality of the Simple Mapping Account (henceforth SMA, adopting the terminology of Piccinini 2010) make it the natural choice as the in-principle standard for physical implementation. It takes the mathematical theory of computation as its starting point and adds no substantive assumptions. In line with SMA, I argue that computation is not an 'intrinsic' property of physical systems, in the sense that (a) it is founded on an observer-dependent act of ascription, upon a purely conventional correlation or mapping between abstract formalism and physical structure. Furthermore, (b) this conventional mapping is essentially *prescriptive* in nature, and hence projects an outside normative standard onto the activities of a physical device. This latter point will be developed in more detail in the next section.

There has been a long standing conflict between proponents of SMA and advocates of CTM, since critics have used SMA to argue that a computational approach to the mind is empirically vacuous. These 'trivialization' arguments hold that a mapping will obtain between any sufficiently complex physical system and virtually any formalism (and thus are akin to Newman's objection to Russell's structuralism). This is construed as fatally undermining CTM, since whatever computational procedure is held to account for our cognitive attributes will also be realized by a myriad of other sufficiently complex arrangements of matter and energy, from buckets of water to possibly even stones. As a case in point, Putnam (1988) offers a proof of the thesis that *every* open physical system can be interpreted as the realization of *every* finite state automaton. And in a closely related vein Searle (1980, 1990) argues that virtually any physical system can be interpreted as following virtually any program.

Let us label multiple realizability in this direction, wherein any given *physical system* can be interpreted as implementing an arbitrarily large number of different *computational formalisms* 'upward MR' and denote it as '↑MR'. In the context of the trivialization arguments, a physical system is viewed as a bounded, continuous region of space-time, and the basic idea is that the region is held constant but is sliced up in an as many different ways as one likes in order to define a chronological sequence of ‘physical states’ that can be mapped to any given finite run of a formal procedure. In the extreme versions suggested by Putnam, Searle, and more recently Bishop (2009), there are apparently no significant constraints whatever – it is possible in principle to interpret every open physical system as realizing every computational procedure.

In response to the trivialization arguments, various authors, including Fodor (1981), Chrisley (1994), Chalmers (1996), Copeland (1996), and Block (2002) have criticized SMA as being itself unduly liberal, and have proposed a number of constraints on computational interpretations in an attempt to distinguish ‘true’ cases of implementation from the myriad of purportedly ‘false’ cases utilized by Putnam and others. Three primary categories of constraint put forward by defenders of CTM include the semantic account, the counterfactual account and the causal account. I give extended arguments against the first two constraints in Schweizer (2014), so in the next section will only consider the causal constraint, which is perhaps the most compelling of the three (see also Bishop (2009) for arguments against the counterfactual account, and Piccinini (2006) against the semantic).

5.6 Computational Ascriptions Are Normative, Not Causal

The causal constraint maintains that a necessary condition for counting as a legitimate implementation is that the pattern of abstract state transitions constituting a particular run of the computational procedure on a particular input, must map to an appropriate transition of physical states of the machine, where the relation between succeeding states in this latter sequence is governed by proper *causal regularities*, and where these regularities *mirror* the structure of the abstract formalism. This suggestion might seem to constitute a plausible restrictive measure, since the physical states in the chronological progressions exploited by Putnam’s method are sliced up at will and have no proper nomological connection.

Nevertheless, I argue that the constraint is too restrictive and emphasizes the wrong level of analysis. Rather than causal structure, I contend that the fundamental criterion is normative; as long as what can be described or interpreted as the *correct* sequence of states actually occurs, then the underlying mechanics of how this takes place are not strictly relevant – the physical *how* is a different question. For example, standard computers rely on a hierarchy of levels of description pertaining to ‘virtual machines’, and it is entirely natural to construe high level virtual machines as genuinely implementing computations, even though the states at this level of description are not themselves causally connected. Of course, in the case of standard computers, there *will be* underlying causal regularities in the hardware which generate the relevant sequence of virtual machine states. But my point is that we don’t need to know anything about this complex causal architecture in order to ascertain that the respective computation is successfully being carried out at the virtual machine level. Just as in the case of Putnam’s SMA states, all we need to take into account is what actually happens at the given level of description, and at this level there are no causal relations between states.

As another example where causal considerations play no role in our judgment that the physical system in question is an implementation, consider Turing's original 1936 heuristics, where the paradigm of actualized computation is a *human computor*, meticulously following a program of instructions and executing computations by hand with pencil and paper. In this exemplary case of concrete realization, the transitions from one state to the next are not governed by causal regularities in any straightforward or mechanical sense. When I take a table of instructions specifying a particular TM and perform a computation on some input by sketching the configuration of the tape and read/write head at each step in the sequence, the transitions sketched on the paper are not themselves causally connected: as in the virtual machine states above, one sketch in the sequence in no way causes the next to occur.

In terms of the underlying causal factors responsible for their occurrence, it is through my understanding of the instructions and intentional choice to execute the procedure that the next stage in the computation appears. But my complex behaviour as a human agent is not something that we currently have any hope of being able to recast in terms of causal relations at a purely mechanical level. In cases of *intentionally mediated* causation, we accept the sequence of configurations on the paper as an execution of the procedure, not because we have any clue concerning the actual causal story, but rather because the sequence itself is *correct* and can be seen to follow the procedural rules.

Of course there will ultimately be some extremely convoluted physical pathway by which the correct sequence of configurations is generated. But again, my point is that the details of this pathway are conceptually irrelevant to answering the question "is the given sequence of configurations an implementation of the formal procedure?" Furthermore, the causal constraint requires that the relation between succeeding states is governed by proper causal *regularities*, and where these regularities *mirror the structure of the abstract formalism*. In the case of computational artefacts, there will be such mechanical regularities at some level. But in the case of intentionally mediated causation, it seems highly implausible that the ultimate causal pathway will bear any resemblance to the algorithm in question, nor be based on any sort of natural 'regularities'. Hence the decisive constraint is not the (completely unknown) structure of the underlying causal pathway. Instead, the key consideration is that the intended mapping, specified by SMA, has been preserved. And this is because the fundamental criterion is *normative* and not causal.

Similarly in the (in)famous Chinese room scenario, it is merely through Searle's subjective *understanding* of English, his voluntary *choice* to behave in a certain manner, and a number of highly disjointed physical processes (finding bits of paper in a certain location, turning the pages in the instruction manual) that the implementation takes place. Undoubtedly Searle's behavior must have a cause, but from this it does not follow that it is governed by any physically characterizable regularities that even remotely reflect the structure of the algorithm. And to further undermine the idea that causal regularities are required, we can let chance and randomness into the scenario. Suppose at each step in the computation Searle flips a coin, and will only follow the rule if the coin comes up heads. And suppose further that, for a particular run on an input question, the coin comes up heads every time and Searle outputs the correct Chinese answer. He has still implemented the formalism, even though this outcome was not predictable on the basis of causal regularities or natural law.

And how could we know that the right causal connections are preserved via Searle's agency, even in the cases where he *sincerely intends* to follow the rule book? – how could we know that at some crucial stage he did not *misunderstand* the rules, and the step he actually intended to perform would have been a mistake, but that by a slip of attention he did *not* perform the step he intended but rather accidentally performed the correct one? As long as the step was correct we would count this as a physical realization of the abstract procedure. And indeed, how do we know that such self cancelling pairs of mistakes don't occur in our computational artefacts?

Before pursuing the vital issue of error and malfunction, I will first consider yet another example where normativity rather than causation is the conceptually essential criterion. In the four state Turing Machine for computing addition on the positive integers given earlier, the program consisted of six quadruples:

$$q_1 1 B q_{1;}\ q_1 B R q_{2;}\ q_2 1 R q_{2;}\ q_2 B 1 q_{3;}\ q_3 1 L q_{3;}\ q_3 B R q_4.$$

The first element in each quadruple (e.g. q_1 in the first case) is the current state, the second element is the currently scanned symbol (either 1 or B for blank, i.e. 0) the third element is the overt action (*move* R or L one square, or *print* a 1 or a B), and the last element is the covert 'act' of entering the next state. In this manner each quadruple can be seen as a conditional instruction: *if* in state q_1 reading a 1, *then* print a B and enter state q_1. Hence it is the *logical* form of the *if-then* statement that captures the import of the effective procedure, and this is what must be satisfied by an implementation. Again, this is an essentially *normative* constraint, and it's a basic truth of logic that the material conditional does not entail any *causal connection* between antecedent and consequent.

This fact is made even more graphic by noting that all possible Turing machine computations can themselves be formalized in first-order logic as in Boolos et al. (2007). For any machine M and input n we can construct a finite set of sentences $\triangle$ that completely encodes the 'actions' of machine M on input n. Each step in the sequence of configurations constituting the computation on n is formalized by a sentence in first order logic which is *logically entailed* by $\triangle$. In this manner, every Turing machine computation is equivalent to a proof in first-order logic, and any such proof carried out with pencil and paper, following the rules of one's favorite first-order deductive system, counts as a *physical implementation* of the computation.

And it seems very odd and implausible to maintain that the property of being a proof in first-order logic is constrained by underlying causal regularities. Indeed, when I mark student exams in my Introduction to Logic course, considerations of underlying causal regularities play no role whatever in determining whether some sequence of formulas is or is not a proof. The only thing that matters is whether or not the rules have been correctly followed, and this is a purely normative consideration.

5.7 Error and Malfunction

Underlying causal considerations are the wrong level of analysis, partly because from this more basic perspective there is no room for error or malfunction to occur – physical systems simply evolve over time in accord with natural law. Malfunction and error can be specified only relative to a higher and non-intrinsic 'design stance' (as in Dennett 1981). Hence when viewed from a purely physical/causal level, a piece of computer hardware (like any other artefact) simply behaves in accord with fundamental causal regularities. If these regularities result in some component becoming overheated, giving rise to an output signal that does not comply with our intended interpretation of the device's behavior as, say, computing the addition function, then we say that an 'error' has occurred.

But as in sense (**1**) above, physical processes 'obey' natural law-like regularities in a purely descriptive manner, and over the time evolution of a physical system the trajectory through state-space may or may not correspond to our projected computational interpretation. If not, then there has been a 'malfunction' in the hardware. But of course, systems governed by causal regularities cannot malfunction as such, and it is only at a higher and *non-intrinsic* level of description that 'breakdowns' can occur. We characterize these phenomena as malfunctions, not because underlying scientific laws have been broken, but rather because our prescriptive and extrinsic interpretation has been violated (as in Kripke (1982)). This is a fundamental respect in which senses (**1**) and (**3**) diverge, and why I take the latter to be the canonical reading of 'computation'.

Accordingly, I would argue that the status of computation is very different than the status of abstract mathematical theories in physics. In physics we are attempting to give a fundamental characterization of 'reality', and in principle at least all existent phenomena supervene upon this fundamental level. There is no substrate neutrality in this case, and instead we are attempting to arrive at a theoretical description of the fixed and given natural order. So the mapping from abstract formalism to physical values is not purely conventional as in the SMA – e.g. the variables are mapped to basic physical magnitudes and not just anything we please. And in the mathematical descriptions of basic physical theory there is *no normativity involved*. If the predictions of a particular theory, say Newtonian mechanics, turn out to be incorrect in certain cases, we do not say that physical reality has therefore 'malfunctioned'. Instead we are forced to say that Newtonian mechanics is at fault and our mathematical description *itself* is incorrect.

In contrast, suppose we take a device intended to compute some given arithmetical function. There is always a non-zero probability of error for any algorithm implemented in the physical world, and since error is always possible it follows that there is no independent fact of the matter regarding which function or algorithm is ‘really’ being computed. Suppose we say that the physical device is computing addition. We confirm this by testing its behaviour on 50,000 inputs and it gives the correct outputs. But unknown to us there is a design fault in the mechanism and when we keep going it gives the ‘wrong’ answers for larger inputs. So which function is it really computing – addition with errors, or the actual function in extension that corresponds to its physically caused behaviour? There is no objective fact to the matter, because computation is a purely extrinsic level of description.

5.8 Computation Versus Explanation

I endorse a non-intrinsic account of computation wherein SMA allows a maximally liberal space of possible interpretations of physical systems. Within this framework we can then apply various interest-relative, purely pragmatic constraints to narrow down the space of possibilities to a proper subset of interpretations, depending upon our varying purposes. There are no overarching necessary and sufficient conditions distinguishing ‘real’ cases of physical implementation. Instead there is an assortment of pragmatic desiderata, where different considerations will take priority in different contexts of application. Consequently SMA and the mathematical theory of computation alone are not sufficient to provide a full explanatory *theory of* particular subject disciplines, such as a *computational theory of the mind*. This is a particular scientific application that adopts computation as a formal tool in its explanatory project, and requires additional resources appropriate to the phenomena and subject area in question.

In this respect what SMA directly threatens, and what has served as the implicit fulcrum in the trivialization debate, is not simply a generic version of CTM, but rather a very specific articulation in the form of the Computational Sufficiency Thesis (CST) (Chalmers (2012)). CST maintains that merely implementing a computational formalism of the appropriate sort constitutes a *sufficient condition* for mentality in the physical world. In order to diagnose the sense in which SMA is supposed to threaten CTM, it is useful to make the tacit structure of the trivialization strategy more explicit with the following reductio argument:

(1) Common sense pre-theoretical truth: this bucket of water *is not* a mental system.
(2) CTM: the human brain is a mental system *‘in virtue of’* implementing (fill in the blank with your favourite theory) $\underline{\text{LOT}}$, which is the appropriate formal architecture for mentality.
(3) CST: any physical system implementing the appropriate formal architecture for mentality is thereby a mental system.
(4) SMA: there is a mapping between this bucket of water and $\underline{\text{LOT}}$, so this bucket of water is an implementation of the appropriate formal architecture for mentality.
(5) Therefore, by CST, this bucket of water *is* a mental system.

Defenders of CTM typically try to block the reductio by rejecting SMA and premise (4). Instead, I would advocate retaining (4) while rejecting (3). Part of the motivation for (3) is that it focuses the basic CTM stance articulated in (2), by construing the locution *'in virtue of'* explicitly in terms of a sufficient condition. In response, I would contend that CST is overly simplified and far too strong, and will argue that the locution in (2) should be interpreted in a more expansive and empirically plausible manner. It's worth noting that from a normal scientific perspective, CST is curious indeed. There are many different levels of description and explanation in the natural world, from quarks all the way to quasars. But there is no comparable sufficiency thesis in chemistry, biology, geology, astronomy. In other 'special sciences', inclusion in the relevant level of description is a matter of degree and scientific utility, not a matter of some uniformly applicable sufficient condition or 'intrinsic' property. I would diagnose much of the controversy over CTM, the trivialization arguments, and concomitant defensive critiques of SMA to be engendered by ill advised allegiance to CST. In doing so, the CTM camp places far too great a theoretical (ideological?) burden on computation. How could the mere fact of implementing the 'right' type of abstract procedure be enough to magically transform an insentient arrangement of matter and energy into a genuine cognitive system?

In contrast, I would argue that much more is required − the system must be anchored in and interact with the real world in a host of rich and multifaceted ways not satisfied by a mere stone or a bucket of water. In terms of a computationally based science of mind, a number of pragmatic and application-specific considerations should come to the fore, to augment the bare and global framework provided by the mathematical theory of computation. In order to give an explanatory account of the mind-brain utilizing high level computational description in classic sense (**3**), we need to treat the brain along the lines of a biologically engineered device comparable to one of our computational artefacts, since many of the same kinds of pragmatic constraints should be invoked.

For example, a crucial difference between our computational artefacts and the attributions of formal structure as employed by ↑MR exercises is that the mapping in the latter case is entirely *ex post facto* – the abstract sequence of states on a particular input is already known and then used as the basis for interpreting various state transitions in the system and hence characterizing it as an implementation. And this is clearly unsuitable as an *explanatory* depiction of the physical device, where we want to make *testable predictions* about its future states. As in the case of artefacts, we want the computational interpretation of brain activity to be systematically preserved over a wide range of future processing. So the ascribed formal structure needs to mesh with the underlying *causal* structure that enables the system to behave in the ways that it does, i.e. in ways salient to its status as a *cognitive* system.

As a provisional standard for purposes of theoretical discussion, it might be thought sufficient for 'genuine mentality' that a system is able to pass the linguistic and robotic Total Turing Test (TTT) (Harnad (1991)). Clearly a stone will never meet this condition and will be ruled out from the start. The typical human being can pass the test, and if we want to explain this ability in high level computational terms, then the mapping must be specified in a way that theoretically integrates the causal and cognitive levels. This theme will be exposited in more detail in the following section.

5.9 The Brain as Cognitive Computer

When viewing a system *qua* mind, as opposed to a mere complex system of some non-mental variety, a standard move is to apply the Belief–Desire (BD) framework of explanation, wherein cognitive agents are seen as possessing a store of propositional attitudes, which rationally combine via psychological processing to *cause* actions. The basic scheme is to ascribe to the agent assorted beliefs and desires and explain/predict that (other things being equal) the system will act to achieve its desires in light of its beliefs. I propose that we restrict CTM to this schematic belief-desire framework and leave conscious experience out of its purview. Within this restricted context, I argue that it is possible to give an account of how a classic computational approach could, at least in principle, offer an effective theoretical handle on the mind/brain, even if we reject CST and accept the view that computation does not comprise a natural kind and is not intrinsic to physical systems.

I will use LOT as an exemplar of the classical approach, although my overall goal is not to argue that CTM (of any variety) is *in fact* the right approach – this remains an open question which must be settled by future scientific research. I merely want to establish that, contra the 'received reading' of the trivialization critiques, SMA is not incompatible with the possibility of a non-trivial development of CTM. According to LOT, propositional attitudes are treated as computational relations to sentences in an internal processing language, and the LOT sentence serves to represent or encode the propositional content of the intentional state. Symbolic representations are thus posited as the internal structures that carry the information utilized by intelligent systems, and they also comprise the formal elements over which cognitive computations are performed. The actual computations are carried out by the causal machinery of the brain, and hence tokens of LOT syntax must ultimately map to neuronal configurations or brain states/processes. As in the case of our artefacts, there may be a hierarchy of levels involved, but in the end there must be a systematic scheme of implementation reaching the physical level.

The formal syntax of LOT thus plays a crucial triad of roles: it can represent meaning, it's the medium of cognitive computation, and it can be physically realized. So a primary theoretical virtue of the classic approach is that the syntax of LOT can supply a link between the high level intentional description of a cognitive agent, and the actual neuronal process that enjoy causal power. This triad of roles allows content bearing states, such as beliefs and desires, to explain salient pieces of behaviour, such as bodily motions, if the intermediary syntax is seen as realized in neurophysiological configurations of the brain. Because the tokens of LOT are semantically interpretable and physically realizable, they form a key theoretical bridge between content and causation.

And this theoretical virtue is not in itself undermined by ‘trivialization’. In contrast to the many possibilities allowed by SMA, a *scientifically significant* mapping is not free to slice up the arrangement of matter and energy comprising the human brain in any way we please. Instead, it must restrict itself to salient causal and functional factors pertaining to the physical system’s time-evolution when viewed *as a brain*, so that the prediction and experimental testing of future states can be used to confirm or disconfirm the scientific utility of the particular mapping. Thus there will be a myriad of pre-existing and empirically intransigent ‘wet-ware’ constraints that the mapping must satisfy, in order to respect the causal structure of brain activity as discovered by neuroscience. This largely independent body of functional and anatomical data would supply a host of highly non-trivial restrictions on what the *material state transitions* should look like that are interpreted as implementations of the abstract computational procedures.

This requirement is in some ways akin to the ‘mechanistic’ accounts of, e.g. Craver (2007), Piccinini (2007), and Milkowski (2013), but is not imposed as a necessary or global constraint on physical computation *per se*. Correspondence with causal/functional mechanisms is invoked only with respect to an *explanatory theory of* a particular domain which adopts computation as a formal tool and utilizes testable predictions to establish its scientific credence. On my account, this approach is justified only by its instrumental success, and it makes no dubious metaphysical claims about ‘real’ implementation or the ‘intrinsic’ purpose or function of causal mechanisms. Nor does it employ any version of CST. Instead, its ‘sufficiency’ in terms of empirical adequacy is based on its predictive success and not by appeal to any mysterious transformational powers of computation.

From a purely computational perspective, LOT must specify the formal pathway responsible for the intelligent input/output profiles of the human mind. In principle then, LOT could be run on a computational artefact and used to *predict* human outputs on given inputs. On this sense (**2**) approach, we could test the accuracy of hypothesized internal processing stories as *computational models* of human cognition. The criterion of successful modelling would be given in terms of ‘macroscopic’ predictions of the extensionally specified input/output behaviour of the system viewed as a black box. In order to interpret the *brain* as implementing LOT and hence performing computations in sense (**3**), the macroscopic LOT input/outputs must be linked by a ‘microscopic’ internal processing story, such that the transitions between the various neurological states implementing respective tokens of mentalese symbols, obeys a *causal progression* in accord with the transformation of these symbols as prescribed by the formalism.

In this fashion, the abstract computational interpretation of brain activity would provide the internal processing account for the input and output capabilities that we want to explain *via* the attribution of internal cognitive structure, e.g. intelligent linguistic performance during a Turing test. And it would involve dual, integrated levels of empirical constraint satisfaction. The computational level of description would have to yield successful predictions of both (i) new macroscopic outputs given new inputs, e.g. sentences in an English conversation and (ii) predictions correctly describing new *brain configurations* entailed by the theory as realizations of the appropriate formal transformations required to produce the predicted macroscopic output.

If this could be done, it would supply a unified perspective wherein the computational level of description supplies the integrating link between actual brain function and the standard belief-desire framework, to account for the successful performance of an intentional agent in the real environment. So the computational interpretation of the causal mechanism would underpin testable predictions of both external cognitive behaviour and internal *physical* state. This is a much richer and more empirically robust version of CTM than a simple Computational Sufficiency Thesis, and is not undermined by the possibility of *ad hoc* mappings between LOT and a bucket of water. Indeed, SMA and the spectre of ↑MR no more threaten this scientifically rigorous version of CTM than they currently undermine the interpretation and utilization of our computational artefacts.

Acknowledgments I would like to thank the reviewers Marcin Milkowski and Marcin Schroeder for a number of helpful comments, as well as Piotr Jablonski for useful general discussion.

References

Bishop, J. M. (2009). Why computers can't feel pain. *Minds and Machines, 19*, 507–516.

Block, N. (2002). Searle's arguments against cognitive science. In J. Preston & J. M. Bishop (Eds.), *Views into the Chinese room*. Oxford: Oxford University Press.

Boolos, G., Burgess, J. P., & Jeffrey, R. C. (2007). *Computability and logic* (5th ed.). Cambridge: Cambridge University Press.

Chalmers, D. J. (1996). Does a rock implement every finite-state automaton?'. *Synthese, 108*, 309–333.

Chalmers, D. J. (2012). A computational foundation for the study of cognition. *Journal of Cognitive Science, 12*(4), 323–357.

Chrisley, R. L. (1994). Why everything doesn't realize every computation. *Minds and Machines, 4*, 403–420.

Copeland, J. (1996). What is computation? *Synthese, 108*, 335–359.

Craver, C. (2007). *Explaining the brain*. Oxford: Oxford University Press.

Dennett, D. (1981). True believers: The intentional strategy and why it works'. In A. F. Heath (Ed.), *Scientific explanation: Papers based on Herbert Spencer lectures given in the University of Oxford*. Oxford: University Press.

Fodor, J. (1975). *The language of thought*. Cambridge: Harvard University Press.

Fodor, J. (1981). The mind-body problem. *Scientific American, 244*, 114–125.

Harnad, S. (1991). Other bodies, other minds: A machine incarnation of an old philosophical problem. *Minds and Machines, 1*, 43–54.

Kripke, S. (1982). *Wittgenstein on rules and private language*. Cambridge: Harvard University Press.

Milkowski, M. (2013). *Explaining the computational mind.* Cambridge: MIT Press.

Piccinini, G. (2006). Computation without representation. *Philosophical Studies, 137*, 205–241.

Piccinini, G. (2007). Computing mechanisms. *Philosophy of Science, 74*, 501–526.

Piccinini, G. (2010). Computation in physical systems. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy*. http://plato.stanford.edu/archives/fall2012/entries/computation-physicalsystems/

Putnam, H. (1988). *Representation and reality*. Cambridge: MIT Press.

Rumelhardt, D. E., & McClelland, J. L. (1986). On learning the past tense of English verbs. In D. E. Rumelhardt & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructures of cognition* (Vol. 2, pp. 216–271). Cambridge: MIT Press.

Schweizer, P. (2012). Physical instantiation and the propositional attitudes. *Cognitive Computation, 4*, 226–235.

Schweizer, P. (2014). Algorithms implemented in space and time. In *Selected papers from the 50th anniversary convention of the AISB* (pp. 128–135). London: Society for the study of artificial intelligence and the simulation of behaviour.

Searle, J. (1980). Minds, brains and programs. *Behavioral and Brain Sciences, 3*, 417–424.

Searle, J. (1990). Is the brain a digital computer? *Proceedings of the American Philosophical Association, 64*, 21–37.

Shagrir, O. (2014). The brain as a model of the world. In *Proceedings of the 50th anniversary convention of the AISB, symposium on computing and philosophy*. http://doc.gold.ac.uk/aisb50. Accessed 15 July 2014.

Shannon, C. E. (1941). Mathematical theory of the differential analyzer. *Journal of Mathematics and Physics, 20*, 337–354.

Turing, A. (1936). On computable numbers, with an application to the entscheidungsproblem. *Proceeding of the London Mathematical Society*, (series 2)*, 42*, 230–265.

Turing, A. (1950). Computing machinery and intelligence. *Mind, 59*, 433–460.

Part II
Philosophy of Computer Science & Discovery

Chapter 6
Computational Scientific Discovery and Cognitive Science Theories

Mark Addis, Peter D. Sozou, Peter C. Lane, and Fernand Gobet

Abstract This study is concerned with processes for discovering new theories in science. It considers a computational approach to scientific discovery, as applied to the discovery of theories in cognitive science. The approach combines two ideas. First, a process-based scientific theory can be represented as a computer program. Second, an evolutionary computational method, genetic programming, allows computer programs to be improved through a process of computational trial-and-error. Putting these two ideas together leads to a system that can automatically generate and improve scientific theories. The application of this method to the discovery of theories in cognitive science is examined. Theories are built up from primitive operators. These are contained in a theory language that defines the space of possible theories. An example of a theory generated by this method is described. These results support the idea that scientific discovery can be achieved through a heuristic search process, even for theories involving a sequence of steps. However, this computational approach to scientific discovery does not eliminate the need for human input. Human judgment is needed to make reasonable prior assumptions about the characteristics of operators used in the theory generation process, and to interpret and provide context for the computationally generated theories.

M. Addis (✉)
Faculty of the Arts, Design and Media, Birmingham City University, Birmingham, UK

Centre for Philosophy of Natural and Social Science, London School of Economics
and Political Science, London, UK
e-mail: Mark.Addis@bcu.ac.uk

P.D. Sozou • F. Gobet
Department of Psychological Sciences, University of Liverpool, Liverpool, UK

Centre for Philosophy of Natural and Social Science, London School of Economics
and Political Science, London, UK
e-mail: p.sozou@lse.ac.uk; fernand.gobet@liverpool.ac.uk

P.C. Lane
School of Computer Science, University of Hertfordshire, Hatfield, UK
e-mail: p.c.lane@herts.ac.uk

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_6

6.1 Introduction

Philosophers of science have, to date, taken a strong interest in how existing or novel scientific ideas are assessed, tested and interpreted. Specific problems that have been considered in this way include: what criteria should be used for rejecting (or accepting the falsification of) scientific theories (such as Popper 1959 and Lakatos 1970), and the implications of scientific uncertainty for public policy (for example Frigg et al. 2013). The question of how new theories are generated has received less attention, although Simon (1973) has suggested that normative, logical processes can be applied to at least some aspects of scientific discovery, such as the discovery of laws.

This study builds on Simon's (1973) approach by describing a computational approach to scientific discovery. It combines two important ideas. The first is that a certain type of scientific theory can be represented as a computer program (Langley et al. 1987). We will refer to this type of theory as a process-based theory: it is a theory that, in some specific form, can be represented directly as an algorithm specifying a sequence of operations that, for a given set of parameters, predicts a set of observations (data). The second is that an evolutionary computational method, known as genetic programming, allows computer programs to be progressively improved through a computational trial-and-error process. Putting these two ideas together leads to a system that can automatically generate and improve scientific theories.

In Sect. 6.2 scientific discovery, including previous work on computational systems to aid the discovery process is reviewed. Section 6.3 considers theories and models in cognitive science. Section 6.4 introduces the genetic programming approach to scientific discovery. Section 6.5 describes an application of this approach to scientific discovery in cognitive science. Philosophical implications of this work are discussed in Sect. 6.6.

6.2 Scientific Discovery

Science is concerned with explaining observations and phenomena (or "data") by means of underlying principles and processes, in a way which is coherent and parsimonious; it has been cast as a method for finding (or at least seeking) a best explanation for data (Thagard 1978). This enables a human understanding of the observations and phenomena, in a form which may encompass mental models, allows commonalities between different phenomena to be established, and (ideally) facilitates predictions. It is often useful to invoke a distinction which can be made between "observational laws", that is, empirical descriptions of the world which directly describe data, and "theoretical laws" or theories which explain data by reference to entities which are generally not directly observable (Holland et al. 1986), although the distinction is not absolute (Langley et al. 1987); historically, the possible role of non-observable entities in descriptions of the world has been of significant interest to philosophers (Achinstein 1965). The development of science involves accumulation (additions to the general body of knowledge), competition (where there are two or more fundamentally incompatible theories or laws), and correction (where a hitherto accepted notion is, in the light of new information, deemed incorrect or at least highly suspect). It is possible for incompatible ideas to coexist within a society or even within the brain of an individual scientist, although such an individual cannot coherently apply such incompatible ideas simultaneously to a given observation (Holland et al. 1986). Where incompatibility occurs at a deep level, and across an entire scientific discipline, it may in some cases be resolved by means of a revolution, that is, an established idea being fully displaced by a new idea which may be better in terms of its ability to explain data or its simplicity (Kuhn 1962).

There are two main conceptions in psychology of the way humans produce discoveries in science: random search and selective search. In the first view (for example Campbell 1960 and Simonton 1999), two processes are postulated: random generation of solutions and then selection of the best solutions. These mechanisms are akin to Darwinian mechanisms of variation and selection. Theories based on this conception predict that more famous scientists produce more output than less famous ones: they generate both more hits and more misses. Because they generate many more ideas, by chance some of them will be highly successful. This prediction is supported by a statistical analysis of historical data (Simonton 1999). For example, Herbert Simon, who was recognised as a leader in several fields including computer science, psychology and economics (for which he won the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel in 1978) was extremely prolific, having produced about 1,000 publications. If the mechanism is Darwinian in form, then it follows that the output of famous scientists should include a substantial number of misses. Simonton (1985) analyzed the output of ten distinguished psychologists, including B.F. Skinner and Donald Campbell. In line with the prediction, he found that, over a 5-year period, 44 % of their publications were never cited. The computational method described in Sect. 6.4 is based on this idea that scientific discovery can be seen as Darwinian variation and selection.

The second view considers scientific discovery as heuristic search – that is, a search involving the application of sensible rules – in a very large space of potential theories (Langley et al. 1987). It is based on Newell and Simon's (1972) theory of problem solving. Creativity in general and scientific discovery in particular are considered as a kind of problem solving, not fundamentally different from tasks such as solving the Tower of Hanoi puzzle or finding the solution of a system of linear equations. Within scientific discovery the heuristics themselves are widely diverse (Thagard 1988), as noted by several philosophers and others, including abduction (Burks 1946), search for patterns (Hanson 1958), search for simplicity and symmetry (Langley et al. 1987), use of mental models (Holland et al. 1986) and even hunches and intuition (Polanyi 1964). Langley et al.'s view is supported by two kinds of evidence. First, experiments have indicated that naïve participants, given the relevant data, can replicate famous scientific discoveries. When they do so, they use simple heuristics (Langley et al. 1987). Second, computer programs based on heuristic search can also replicate famous scientific discoveries, such as Kepler's third law in astronomy, Boyle's law in physics and Snell's law in optics. The heuristics used search for patterns in the data in order to find regularities: the process is akin to dimension reduction in data analysis.

While the search process has been cast as either random or selective, in reality it should be recognized that there is a continuum with respect to the degree of selectivity: any search method involving a degree of randomness requires some specification of the way that search is conducted, i.e. which parts of the search space are more likely to be searched. From this point of view, it can be said that Langley et al. (1987) conceptualise scientific discovery as a more selective process than Campbell (1960) and Simonton (1999).

Statistical and computational advances have contributed to scientific discovery. The development of Bayesian networks – graph-based probabilistic models of dependencies between variables in complex systems – has enabled influences between variables to be uncovered, with applications to problems such as characterising interactions between genes and proteins (Friedman et al. 2000). There have also been important developments toward more general automation of the process of formulating, sifting and testing hypotheses. DENDRAL (Lindsay et al. 1993) was developed to find chemical structures from mass spectrometry data. The BACON research programme showed that computational techniques can be applied to the discovery of laws (Langley 1981). In principle this programme has a certain amount in common with established statistical methods such as regression and dimension reduction, but it can go beyond these direct data description methods by generating variables representing intrinsic properties of entities, such as the refractive index (Langley et al. 1981). More recently King et al. (2009) have described the operation of a "robot scientist" which iteratively collects and analyses data and generates hypotheses; these hypotheses are fairly directly determined from the data, such as establishing which genes are involved in encoding enzymes. However, neither Bayesian networks nor BACON nor the robot scientist of King et al. (2009) have the capacity to develop more complex theories involving a sequence of steps in a processed-based theoretical model of the sort described in Sect. 6.3 and illustrated in Sect. 6.5.

6.3 Theories in Behavioural and Cognitive Science

There may be more than one level of explanation for a given phenomenon. Humans and other animals are purposeful: many of their actions can be understood strategically by reference to goals they are trying to achieve, as a result of their behaviour having been shaped by natural selection. This can lead at its simplest to two clear, complementary levels of explanation for a given behaviour. It can, on the one hand, be explained in terms of the strategic objectives the person or animal is trying to achieve, leading to a research programme exemplified by behavioural ecology (for example Krebs and Davies 1993). Testing such strategic theories requires a high degree of theoretical inference to derive predictions regarding patterns in data (e.g. Charnov 1976). Or, on the other hand, it can be explained in terms of underlying processes, cognitive or neural, concerned with how specific functions and mechanisms lead to behaviour. An example of this approach is the application of information processing models to cognition (Simon 1979). An important characteristic of this process-based approach is that specific theories yield predictions, without the need for additional high-level theoretical inference. This allows such theories to be expressed algorithmically and manipulated computationally. Section 6.4 describes a modelling method which is a development of this approach.

Recent attention has been paid to the question of whether or not explanations of cognitive processes, based on combinations of different functional capacities and modules, can be regarded as mechanistic (Piccinini and Craver 2011 and Barrett 2014). The functional view (Barrett 2014) models cognition as though it arises from a combination of relatively high-level processes known as functions, and generally takes these functions as given without a need for a detailed consideration of the lower-level processes that give rise to them. Conversely, a mechanistic view tends to put emphasis ultimately on low-level processes, known as mechanisms, from which functions are presumed to be built up (Piccinini and Craver 2011). The approach we describe in detail in Sect. 6.4 has some common elements with both of these views, in that the cognitive process underlying a given behaviour is modelled as a mechanistic sequence of functional operations, involving such specific operations as putting items into short-term memory, or comparing items in different short-term memory slots.

This approach raises the more general question: what exactly is a model, and how is it related to a theory? A model can be regarded as a representation of a real system; the purpose of the model is to facilitate the answering of “what if” questions (for example Kowald 1997). A quantitative model can be regarded as an instantiation (that is a specific version) of a theory (Lane and Gobet 2012). Models usually involve a deliberate simplification of reality. The purpose of this simplification is to make the model computationally tractable, or to restrict consideration to factors which are causally relevant to the process being investigated (Weisberg 2007). Sometimes these two reasons for simplification act in concert. For example, Sozou and Kirkwood (2001) were concerned with understanding the possible factors that cause human fibroblast cells to stop dividing after a finite number of replications. They developed a model which involved a combination of telomere loss, the build up of mitochondrial mutations causing oxidative stress, and somatic mutations, and implemented it as a computer program. The model was restricted to a small number of factors which had a large influence on the replicative potential of cells, for reasons of both computational simplicity and explanatory power.

Models in which some aspect of physical reality is discarded for the sake of explanatory convenience are sometimes characterised as fictional models (Frigg 2010). An example is the ideal pendulum, which is not subject to frictional forces (Contessa 2010). This notion of models as fiction raises the issue of how fictional entities and truth ascriptions to them should be regarded, with work on the philosophy of fiction (such as Walton 1990) providing a variety of possibilities.

In what follows it will be assumed that a satisfactory account of the nature of fictional entities and their associated truth claims is possible as going into the details of this would lead too far afield. It is probably the case that most theoretical models in science can be regarded as fictional. The aforementioned model of Sozou and Kirkwood (2001) can be regarded as a fictional model of an ideal cell. Similarly, the cognitive modelling approach described below in Sect. 6.4 can be regarded as a fictional modelling approach.

If a model is successful at explaining a given set of observations, does this make it a "good" model? As Fodor (1968) has pointed out, successfully describing a data set does not imply that a model is correct in any meaningful sense other than describing the data. Yet almost all moderately complex process-based models in science are "wrong" in some respects, and can be made to fail (in the sense of making a prediction that differs in a substantive way from what is observed in the real world) in some circumstances. Most scientists and philosophers would accept that some models are more generally useful than others. Considerations such as how general a model's predictions are, and how simple it is, play a part in this, but the applicability of a given model in a given situation must largely be a matter of human judgment, a point we return to in Sect. 6.6.

6.4 A Computational System for Theory Discovery in Cognitive Science

As has been indicated in Sect. 6.2 above, scientific research can be conceptualised as a heuristic search process and candidate theories can be represented as computer programs (Langley et al. 1987). A significant challenge is to locate a good theory from within the combinatorially large space of candidate theories. The space of all possible computer programs provides the search space for candidate theories. In our approach, the quality of a theory is determined by testing the theory against a set of experimental results. This is achieved by treating the theory as a participant in an experiment, collecting its responses to the experimental stimuli, and evaluating the theory's performance to compute its 'fit'. Better theories will provide closer fits to the experimental data. The search process is used to refine a current set of theories into a new set, and relies on heuristics to guide the search towards better theories: heuristics will tend to generate better theories, but they do not provide any guarantee that the process will yield an optimum or "best" theory.

There are many ways in which this broad picture can be converted into a working theory-discovery system. Pilot studies (Frias-Martinez and Gobet 2007; Lane et al. 2014 and Gobet and Parker 2005) have described a theory representation language to create the space of candidate theories. The search through this space is achieved using a form of evolutionary computation known as genetic programming (Koza 1992 and Poli et al. 2008). This enables computer programs to evolve to fit a particular task, by simulating the process of natural selection.

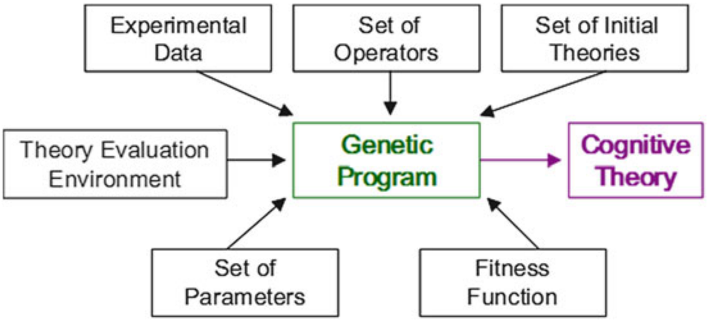


Fig. 6.1 Schematic overview of the theory discovery system

Figure 6.1 illustrates the general structure of the system. At the heart of the system is the genetic program which outputs a candidate cognitive theory. The genetic program requires a number of inputs. Some of these are relatively general, such as the set of operators (defined by the theory representation language noted above), the set of parameters (required for the genetic program), and the set of initial theories (created from the set of operators, usually randomly). Some of the inputs are specific to the problem or class of problem being modelled, including the experimental data and theory evaluation environment, which are used to compute the fitness of a candidate theory.

The aim of the system is to discover new cognitive theories, which are represented as computer programs. A computer program is a sequence of instructions corresponding, in this case, to an algorithm defining a model of the processes believed to underlie some human cognitive behaviour. There are many kinds of models of cognitive processes: here a representation of a typical symbolic computation model (Simon 1979) is used. This kind of model can interact with an external environment through gaining information or responding with actions. The model has two kinds of internal memory: a short-term memory (STM), for holding temporary information, and a long-term memory (LTM), for holding more permanent information. Each program, specifying a model is built up from a set of primitive operators, each representing a basic psychological process. These operators come in three forms: operators to interact with the external environment, operators to interact with the elements of the model, and general purpose operators. It is assumed that there are specific error rates associated with each operator.

Each cognitive model is represented as a tree, which is the internal representation used within the search process. Each node in the tree holds an operator. The descendants of each node are its children, and represent the sub-trees that the node's operator works on. The tree should be read from the top down. The interpretation of the operators, their semantics, and how they use their children are all specified within the theory representation language. A specific example of such a program representing a scientific theory, and generated by genetic programming, is given in Sect. 6.5.

The complete theory language includes general operators, including conditional operations which test a condition and decide which of two sub-trees to execute depending on the value of the condition. Another important type of operator is the iteration operator, which can be used to execute a sub-tree a given number of times, or while some condition holds true. The language includes values for Boolean data types (representing true and false), integer and real numbers. Although restricted in syntax, the theory language is a Turing-complete language which is capable of representing any computational process.

The range of possible programs that can be created in the theory language can be thought of abstractly as a space of possible programs. The search process is used to locate, within this space, one or more candidate programs that have a reasonable fit to the target data. The search process is based on making repeated changes to a population of candidate programs, making alterations which should create "better" programs. This search process is managed using genetic programming. The process begins with an initial population of computer programs (usually randomly generated); the population size is one of the parameters of the genetic program implementation. Each cycle creates a new population by taking the best programs from the previous population and constructing new programs from them. The new programs are created using the processes of mutation and crossover (Fig. 6.2). Mutation involves making a random change to a program, by replacing a node or sub-tree of the program. Crossover involves two programs exchanging sections of instructions with each other. Those programs which perform badly will not be selected to create new programs, and so they will be selected out of the population. Those programs which perform well will be used to create new programs, and so their properties will tend to feature in the later populations. In this way, a search space of possible programs is explored through an evolutionary process, with each population tending to perform better than the one before. A fuller discussion of how the genetic programming technique can facilitate evolvability of programs is given by Altenberg (1994).

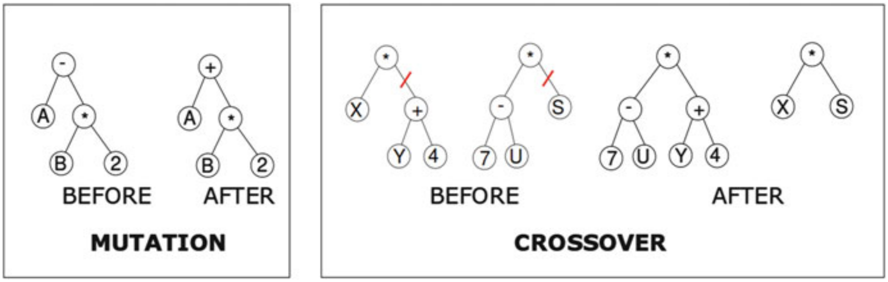


Fig. 6.2 Mutation and crossover in genetic programming. Mutation involves a random change to a single program: in the case shown, a subtract operator (−) has been changed to an add operator (+). Crossover involves transposing section of two programs: in the case shown, these are the right-hand branches of the program immediately below the top node

The genetic programming operation ends when a specified end-point has been reached: either the best program from the population has achieved a desired level of performance, the overall performance has stopped improving significantly over subsequent generations, or a maximum number of generations has been executed. Commonly, the search process is terminated after a fixed number of generations. This is the case in the examples presented below.

As with any modelling system, the theory discovery system incorporates a number of assumptions and biases which make the search space tractable; a bias can be a fixed limitation to what can be achieved, or it can simply be a disposition of the system to return certain kinds of results. The theory representation language defines the basic operators from which any candidate theory will be constructed. This language is ‘unrestricted’, in the sense that it represents a complete model of computation, so any candidate theory could, in principle, be represented using the language. The search process uses heuristics which impose a bias on the way search is conducted through the space of candidate theories. The local nature of the mutation and crossover operators means that all generated theories will be based on the original set of candidate theories. This problem is a typical problem in evolutionary systems, and is widely acknowledged in biological evolution, where species often possess vestigial limbs or other features due to their evolutionary history.

These biases mean that the system tends to explore a limited space of candidate theories which represent a particular class of cognitive theory. This kind of bias is typical for any kind of discovery or learning system, and indeed, an unbiased system would never be able to generate theories which generalise to future situations (Mitchell 1980). Different decisions for the theory language or the search process would lead to different kinds of bias, and so the system would discover different candidate theories.

This framework provides a plausible space for theory discovery, for three main reasons. First, the genetic program process of search has proved successful in over two decades of applications (Poli et al. 2008). It should be recognised that there remain a number of open problems in genetic programming (O’Neill et al. 2010), but this suggests that there is scope for further improvement of this technique. Second, there is a prevalence of raw data in the domains which we are interested in modelling, such as categorisation (Lane and Gobet 2013 and Smith and Minda 2000). Third, the class of cognitive theory being captured is typical of widely used symbolic cognitive architectures, such as CHREST (Gobet et al. 2001) and Soar (Newell 1990), which have already provided a number of successful models for these kinds of applications (see Samsonovich 2010 for a recent survey).

6.5 Example of Theory Discovery

In this section, a more detailed example of theory discovery in action is considered. The focus is upon the representation of the theory, how ‘quality’ is measured, and typical results (for more details see Frias-Martinez and Gobet 2007, and Lane et al. 2014). The task used is the delayed match to sample (DMTS) task, which explores

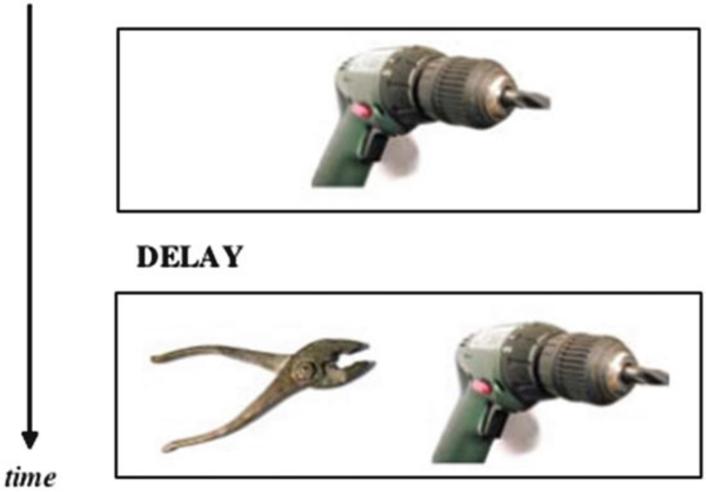
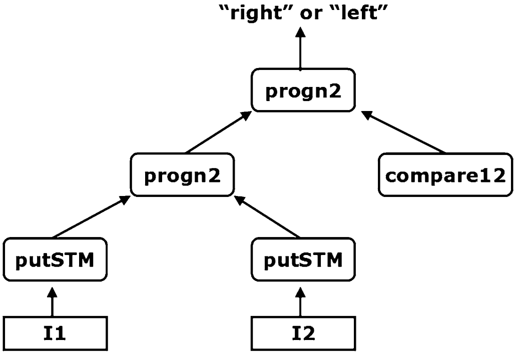


Fig. 6.3 Delayed match to sample task. Photographs courtesy of www.freeimages.co.uk.

Fig. 6.4 Example theory



processes of short-term memory, categorisation and object recognition. Figure 6.3 illustrates the task. First, an image is shown. Then, after a delay, the same image is shown alongside a second one. The task is simply to indicate which of the two new images corresponds to the one first shown. Chao et al. (1999) explored how people perform on this task, with images of tools and animals. Subjects achieved a mean accuracy at this task of 95 %, with a mean time of response of 767 ms (only the experiment where tools were used as the source of images is modelled here).

The theory is constructed from a number of possible operators. Figure 6.4 shows an example (in tree form) of a relatively simple theory generated by genetic programming, taken from Frias-Martinez and Gobet (2007). This study involved fitting only the predicted accuracy, and not the reaction time, to the data. A Lisp implementation of genetic programming was used (Koza 1992). The population

Table 6.1 Operators used in the theory shown in Fig. 6.4

Operator	Description
Progn2	Function: executes two inputs sequentially
	Input: Input1, Input2
	Output: the output produced by Input2
PutSTM	Function: writes the input into STM
	Input: Input1
	Output: the element written in STM (Input 1)
Compare12	Function: compares positions 1 and 2 of STM and returns NIL if they are not equal or the element if they are equal
	Input: none
	Output: NIL or the element being compared

size was 20, and the maximum number of generations was set at 50. Table 6.1 lists the operators used in this theory. The top-most operator in the tree is a general-purpose operator, Progn2. This operator takes two other sub-trees. It first executes the sub-tree on the left, and then the sub-tree on the right. The result of the right-most sub-tree is then output, as a result. The other operators illustrated in Fig. 6.4 include PutSTM, which accepts one input, and places that input into short-term memory (STM); Input1, which reads input 1 from the external environment; Input2, which reads input 2 from the external environment; and Compare12, which checks whether STM positions 1 and 2 contain the same or differing data.

Reading the program as a whole, we have the following process. The program first reads input 1, and places the result into STM. It then reads input 2, and places the result into STM. Next it compares the values in STM positions 1 and 2, and uses the result of that comparison to decide on a response. Also note that, as stated in Sect. 6.4, it is assumed that there are specific error rates associated with each operator: it is through these errors that the theory predicts, in accordance with experimental data, that subjects sometimes give the wrong answer.

A more recent study of applying genetic programming to the DMTS problem (Lane et al. 2014) considered the problem of getting a good fit between simulations and the real experimental data for both accuracy and predicted reaction time. This used an open-source Java implementation of genetic programming known as ECJ (Luke 2010). The genetic program was run with a population size of 500, over 500 generations. When the theory is executed, the program is run in simulated time; each operator has an associated time, and these are simply added during execution, leading to a total simulated reaction time. The best of the theories found in this study (that is, those closest to the experimental data) differ from the experimental data by less than 23 % in predicted accuracy and less than 2 % in predicted reaction time. In this particular domain, it can be said that the theory discovery process was successful in locating theories which fit the target data reasonably well. Current work is looking to apply this same technique to problems of visual attention and categorisation.

6.6 Discussion

It might be useful at this juncture to consider some of the characteristics of the theories that are generated by the genetic programming method. First, theories are formal and can be easily, if necessary, translated into a syntactic language. Second, they are mechanistic (that is processes such as retrieving information from short-term memory are clearly specified) and thus explanatory. Third, they can be tested, and indeed have been tested with the data used during the genetic program evolution. Fourth, they make clear cut predictions, both about performance and response time – the latter being made possible by the time parameters used in our approach. For example, in the delayed match to sample experiment, the evolved theories can be tested in variations where the presentation time of the images is varied, and new human data collected to test their predictions. Fifth, as the theories generally involve relative simple sequences of processes, they can be easily understood by humans. This is unlike, for example, neural networks, which might account for empirical data but whose "explanations" are not transparent for humans. Sixth, they are flexible, in the sense that they can easily be modified by a human theorist, either to simplify them or to add new mechanisms. Finally, if parsimony is deemed a valuable feature, theories can have this characteristic by incorporating it in the fitness function: penalizing larger programs (where size can be defined as the number of nodes in the output of the genetic program) will tend to result in smaller programs (e.g. Zhang and Mühlenbein 1995), corresponding to more parsimonious theories. A trade-off between a theory's size and its accuracy in explaining the data can be captured by varying the relative weight applied to the size penalty term in the fitness function. Such trade-offs between different desirable attributes of a theory are central to one of the main justifications for the maintenance of multiple, incompatible models to describe a given system, a situation termed multiple model idealization (Weisberg 2007).

The generated theories are at the same level of complexity as many theories published in psychology and neuroscience journals, bearing in mind the qualification that many of those theories are informally stated and thus contain much vagueness and ambiguity. The delayed match to sample example in Sect. 6.5 has shown that a heuristic search process using evolutionary computation as a search tool can generate process-based theories involving a sequence of steps.

It must nevertheless be recognized that this process of automating scientific discovery does not eliminate the need for human input. One reason for this is that reasonable prior assumptions about the characteristics of operators, as represented in the theory language, must be made. These assumptions must be consistent with available data about processes in the brain (Frias Martinez and Gobet 2007), including any stochasticity; occasional errors in the actions of operators, defined by error rates, can be thought of as one form of stochasticity, but the method can accommodate other forms of stochasticity such as randomness in execution times. Another reason that human input is required is that, once basic theories that fit data have been discovered, the human scientist is important in interpreting and providing context for these theories. In the first place, a judgment must be made about the plausibility of a theory. The basis for the theories generated by the genetic programming method is that they provide a good statistical fit to the data, but a good fit to data does not guarantee that a model is plausible (Fodor 1968). A human scientist, with a deep understanding of the relevant field, must make a judgment about plausibility. There may come a day, through further developments in artificial intelligence, when a computational system will capture this broader form of scientific understanding, but this day does not appear to be imminent. Even where a theory is plausible, there are likely to be limits to its external validity, i.e. to the circumstances to which it can be expected to apply. For psychological tasks, these limits would be determined by such factors as the characteristics of the people undertaking the tasks, or of the nature of the tasks themselves. To take the theory shown in Fig. 6.4 as an example: how similar do two images need to be before a comparison operator can be expected to find them to be the same? This will be a function of the visual system, but as the simple theory of Fig. 6.4 does not incorporate the visual system explicitly, human judgment is needed.

In conclusion, while there is potential for computational methods to discover basic scientific theories, the human scientist is not about to be made redundant.

Acknowledgments This research was supported by Economic and Social Research Council grant ES/L003090/1. We thank three referees for comments on this paper.

References

Achinstein, P. (1965). The problem of theoretical terms. *American Philosophical Quarterly, 2*(3), 193–203.

Altenberg, L. (1994). The evolution of evolvability in genetic programming. In K. Kinnear (Ed.), *Advances in genetic programming* (pp. 47–74). Cambridge: MIT Press.

Barrett, D. (2014). Functional analysis and mechanistic explanation. *Synthese, 191*, 2695–2714.

Burks, A. W. (1946). Peirce's theory of abduction. *Philosophy of Science, 13*, 301–306.

Campbell, D. (1960). Blind variation and selective retention in creative thought as in other knowledge processes. *Psychological Review, 67*, 380–400.

Chao, L., Haxby, J., & Martin, A. (1999). Attribute-based neural substrates in temporal cortex for perceiving and knowing about objects. *Nature Neuroscience, 2*, 913–919.

Charnov, E. L. (1976). Optimal foraging, the marginal value theorem. *Theoretical Population Biology, 9*(2), 129–136.

Contessa, G. (2010). Scientific models and fictional objects. *Synthese, 172*, 215–229.

Fodor, J. (1968). *Psychological explanation: An introduction to the philosophy of psychology*. New York: Random House.

Frias-Martinez, E., & Gobet, F. (2007). Automatic generation of cognitive theories using genetic programming. *Minds and Machines, 17*, 287–309.

Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology, 7*(3–4), 601–620.

Frigg, R. (2010). Models and fiction. *Synthese, 172*, 251–268.

Frigg, R., Smith, L., & Stainforth, D. (2013). The myopia of imperfect climate models: The case of UKCP09. *Philosophy of Science, 80*, 886–897.

Gobet, F., & Parker, A. (2005). Evolving structure-function mappings in cognitive neuroscience using genetic programming. *Swiss Journal of Psychology, 64*, 231–239.

Gobet, F., Lane, P. C., Croker, S., Cheng, P. C., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences, 5*, 236–243.

Hanson, N. (1958). *Patterns of discovery*. Cambridge: Cambridge University Press.

Holland, J., Holyoak, K., Nisbett, R., & Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge: MIT Press.

King, R., Rowland, J., Oliver, S., et al. (2009). The automation of science. *Science, 324*, 85–89.

Kowald, A. (1997). Possible mechanisms for the regulation of telomere length. *Journal of Molecular Biology, 273*, 814–825.

Koza, J. (1992). *Genetic programming: On the programming of computers by means of natural selection* (Vol. 1). Cambridge: MIT Press.

Krebs, J., & Davies, N. (1993). *An introduction to behavioural ecology*. Oxford: Blackwell.

Kuhn, T. (1962). *The structure of scientific revolutions*. Chicago: University of Chicago Press.

Lakatos, I. (1970). Falsification and the methodology of scientific research programmes. In I. Lakatos & A. Musgrave (Eds.), *Criticism and the growth of knowledge* (pp. 91–196). Cambridge: Cambridge University Press.

Lane, P., & Gobet, F. (2012). A theory-driven testing methodology for developing scientific software. *Journal for Experimental and Theoretical Artificial Intelligence, 24*, 421–456.

Lane, P., & Gobet, F. (2013). Evolving non-dominated parameter sets for computational models from multiple experiments. *Journal of Artificial General Intelligence, 4*, 1–30.

Lane, P., Sozou, P., Addis, M., & Gobet, F. (2014). Evolving process-based models from psychological data using genetic programming. In R. Kibble (Ed.), *Proceedings of the 50th anniversary convention of the AISB*. London: AISB.

Langley, P. (1981). Data-driven discovery of physical laws. *Cognitive Science, 5*, 31–34.

Langley, P., Bradshaw, G., & Simon, H. (1981). BACON 5: The discovery of conservation laws. In *Proceedings of the 7th IJCAI* (pp. 121–126). San Francisco, CA: Morgan Kaufman.

Langley, P., Simon, H., Bradshaw, G., et al. (1987). *Scientific discovery: Computational explorations of the creative processes*. Cambridge: MIT Press.

Lindsay, R., Buchanan, B., Feigenbaum, E., et al. (1993). DENDRAL: A case study of the first expert system for scientific hypothesis formation. *Artificial Intelligence, 61*, 209–261.

Luke, S. (2010). The ECJ owner's manual. In *A user manual for the ECJ evolutionary computation library,* San Francisco, California. Available at http://cs.gmu.edu/~eclab/projects/ecj/docs/manual/manual.pdf

Mitchell, T. (1980). *The need for biases in learning generalizations* (Technical report). New Brunswick, NJ: Rutgers University. Laboratory for Computer Science Research: Rutgers University.

Newell, A. (1990). *Unified theories of cognition*. Cambridge: Harvard University Press.

Newell, A., & Simon, H. (1972). *Human problem solving*. Englewood Cliffs: Prentice-Hall.

O'Neill, M., Vanneschi, L., Gustafson, S., & Banzhaf, W. (2010). Open issues in genetic programming. *Genetic Programming and Evolvable Machines, 11*, 339–363.

Piccinini, G., & Craver, C. (2011). Integrating psychology and neuroscience: Functional analyses as mechanism sketches. *Synthese, 183*, 283–311.

Polanyi, M. (1964). *Personal knowledge: Towards a post-critical philosophy*. London: Routledge.

Poli, R., Langdon, W., & McPhee, N. (2008). *A field guide to genetic programming*. Available from http://www.gp-field-guide.org.uk

Popper, K. (1959). *The logic of scientific discovery*. London: Hutchinson.

Samsonovich, A. (2010). Toward a unified catalog of implemented cognitive architectures. *BICA, 221*, 195–244.

Simon, H. A. (1973). Does scientific discovery have a logic? *Philosophy of Science, 40*, 471–480.

Simon, H. (1979). Information processing models of cognition. *Annual Review of Psychology, 30*, 363–396.

Simonton, D. K. (1985). Quality, quantity, and age: The careers of ten distinguished psychologists. *International Journal of Aging and Human Development, 21*, 241–254.

Simonton, D. (1999). *Origins of genius*. Oxford: Oxford University Press.

Smith, D., & Minda, J. (2000). Thirty categorization results in search of a model. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 26*, 3–27.

Sozou, P., & Kirkwood, T. (2001). A stochastic model of cell replicative senescence based on telomere shortening, oxidative stress, and somatic mutations in nuclear and mitochondrial DNA. *Journal of Theoretical Biology, 213*, 573–586.

Thagard, P. R. (1978). The best explanation: Criteria for theory choice. *The Journal of Philosophy, 75*, 76–92.

Thagard, P. (1988). *Computational philosophy of science*. Cambridge: MIT Press.

Walton, K. (1990). *Mimesis as make-believe: On the foundations of the representational arts*. Cambridge: Harvard University Press.

Weisberg, M. (2007). Three kinds of idealization. *The Journal of Philosophy, 104*, 639–659.

Zhang, B. T., & Mühlenbein, H. (1995). Balancing accuracy and parsimony in genetic programming. *Evolutionary Computation, 3*, 17–38.

Chapter 7
Discovering Empirical Theories of Modular Software Systems. An Algebraic Approach

Nicola Angius and Petros Stefaneas

Abstract This paper is concerned with the construction of theories of software systems yielding adequate predictions of their target systems' computations. It is first argued that mathematical theories of programs are not able to provide predictions that are consistent with observed executions. Empirical theories of software systems are here introduced semantically, in terms of a hierarchy of computational models that are supplied by formal methods and testing techniques in computer science. Both deductive top-down and inductive bottom-up approaches in the discovery of semantic software theories are refused to argue in favour of the abductive process of hypothesising and refining models at each level in the hierarchy, until they become satisfactorily predictive. Empirical theories of computational systems are required to be modular, as modular are most software verification and testing activities. We argue that logic relations must be thereby defined among models representing different modules in a semantic theory of a modular software system. We exclude that scientific structuralism is able to define module relations needed in software modular theories. The algebraic Theory of Institutions is finally introduced to specify the logic structure of modular semantic theories of computational systems.

Keywords Philosophy of computer science • Semantic view of theories • Modelling • Scientific structuralism • Abstract model theory

N. Angius (✉)
Dipartimento di Storia, Scienze dell'Uomo e della Formazione, University of Sassari, Sassari, Italy
e-mail: nangius@uniss.it

P. Stefaneas
Department of Mathematics, School of Applied Mathematical and Physical Sciences, National Technical University of Athens, Athens, Greece
e-mail: petros@math.ntua.gr

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_7

7.1 Introduction: The Need of Empirical Theories of Software Systems

Aim of the software evaluation processes is *predicting* the future behaviours of the examined systems in order to know whether those behaviours are consistent with the required software specifications. Evaluations of programs' correctness might involve the mathematical approaches provided by formal methods (Fisher 2011), the more empirical practices of software testing (Ammann and Offutt 2008), or a combination of both. In any case, one would like to get to a set of sentences yielding predictions of the system's future computations. This paper is concerned with the construction of *theories* of software systems that be adequately predictive with respect to their target systems' behaviours. We focus on the process of *discovery* of those theories, that is, on the development of theories from the availability of computational models that have been verified or tested during some software evaluation phase. *Abstract model theory* is finally utilized to define the structure of predictive theories of modular systems.

Programs are abstract, human-made, entities, about which it is in principle possible to acquire an a-priori knowledge. Formal methods have been developed in theoretical computer science with the aim of performing a static code analysis, not involving the execution of the implemented software system. Benefits of formal methods concern the opportunity of performing, in principle, an exhaustive examination of the program's code and of coming to an effective answer as what concerns the behavioural properties of interest. Some of those properties, usually formalised in a specification language, are common to all programs, such as deadlock freedom; others are relative to classes of programs, as the liveness property of reaching a desired state or avoiding an undesired state under some specified conditions (Baier and Katoen 2008, 12). In formal methods, programs' code is algorithmically checked against those specifications. The effective methods thereby provided are of particular significance in all those contexts in which empirically testing the system is either unfeasible (such as with software involved in robotic-aided surgery) or has unacceptably expensive costs (such as testing rocket controller software).

Since the original development of formal methods in the 1970s, a debate arose between mathematicians and engineers as what concerns the actual reliability of formal methods in the evaluation of the correctness of programs with respect to the desired set of specifications (Shapiro 1997). In a famous paper, Hoare (1969) maintained that logic enables one to determine the set of allowed behaviours of a given program in terms of the closed set of consequences deduced within an axiomatic system representing the program. Since Hoare's original essay, Theorem Proving put faith in the opportunity of acquiring a-priori proofs of programs' correctness and thereby of defining *mathematical theories* of software systems. Such theories are syntactic theories characterised by a set of axioms, formalising enabling conditions for programs' executions, and by a set of rules of inference enabling one to deduce allowed computations from the specified set of axioms. Many objections to mathematical proofs of correctness appealed to the undecidability resulting from the incompleteness of those axiomatic systems, others on the complexity of carrying out proofs which therefore demand for computer aid, others on the difficulty of providing well-defined specifications (Shapiro 1997).

Fetzer (1988) provided a rather methodological objection to the feasibility, for formal methods, to provide predictions of programs' behaviours that are coherent with observed executions. Any formal method makes inferences on programs as abstract machines, that is, as mathematical entities about which it is unsurprisingly possible to perform deductive reasoning. However, one would like to evaluate the correctness of the physical machines instantiating those abstract machines. A mathematically correct program might produce incorrect executions when instantiated; this might be due to many reasons, including hardware exceptions or unexpected interactions with users and environments.

If one is involved in the predictions and explanations of actual executions of physical running machines, *empirical theories* are required. Angius and Tamburrini (2011) propose an account in this perspective introducing *semantic theories* (Suppe 1989) of software systems defined by a set of computational models representing *observed* executions of the verified program. Models are organised into an abstracting/instantiating hierarchy mapping executions of an abstract state transition system used in formal verification into paths of a *model of data* (Suppes 1962) representing observed executions. Mappings ensure that actual executions correspond to abstract paths in the state transition system, thus meeting Fetzer's objection.

Suppes (1962) suggested that, in a semantic theory, mappings from abstract theoretical models to concrete models of data cannot be given directly, but by means of a hierarchy of abstract mapped models, as depicted in Fig. 7.1. At the top of such hierarchy, an abstract transition system enabling one to perform an algorithmic check of the model, such as by means of the Model Checking techniques (Baier and Katoen 2008). Subsequent concretizations of the theoretical model lead to the *model of the experiment*, containing paths representing those executions that ought to be observed in order to empirically evaluate the correctness of the program involved. At the bottom of the semantic hierarchy, a model of data concretizes models of experiments by containing instances of the runs represented in the latter and which should correspond, in case of adequate representation, to observed runs.

It should be noted here how computational models identifiable with theoretical models, models of the experiments, and of models of data, are of different types in the actual practice of computer science. Kripke structures and other labelled automata are mostly used in Model Checking, whereas data flow graphs are often utilised to model observed executions in software testing. A first difficulty in the construction of the semantic theories of software systems of Fig. 7.1 is providing mapping relations among structures modelled using different formalisms. Most importantly, providing a state transition system or a data flow graph representing all executions of a non-trivial program is hardly feasible. Both formal verification and testing techniques are carried out modularly, that is, the program's modules are evaluated independently (Müller 2002). Models used to represent different modules are also represented using different logics, even when performing the same verification technique but in case divergent properties are to be checked in each module.

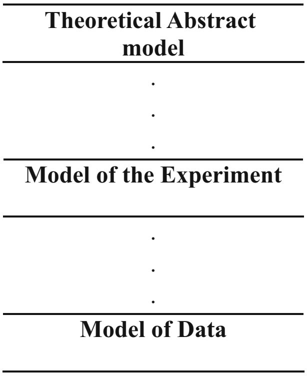


Fig. 7.1 A semantic theory's representational hierarchy according to Suppes (1962)[1]

This paper faces the problem of *discovery* of semantic theories of *modular* software systems, that is, of achieving the semantic hierarchy of Fig. 7.1 from a collection of models, expressed within heterogeneous formalisms and representing different modules of the same program. Section 7.2 suggests that logic relations among models representing different program's modules be provided in such modular theories. Before proposing a structure for those modular theories, Sect. 7.3 takes into consideration scientific structuralism (Balzer et al. 1987) to show how it is not able to represent heterogeneous modules interactions. Finally, Sect. 7.4 introduces to the *Theory of Institutions* (Goguen and Burstall 1992), an abstract model theory applied to software specification languages, to construct empirical theories of modular computational systems.

7.2 Discovering Empirical Theories of Software Systems

In the context of the ongoing debate dividing verificationists from testers, the mathematician Goguen (1992) highlighted how an "error – free" top-down approach from abstract machine correctness to physical machine correctness cannot be pursued: whether the running computational system is a fair instantiation of the abstract state transition system cannot be a-priori settled. Indeed, computational models used in formal verification are usually built upon the program's available code and hardware correctness is usually assumed in the form of the so-called fairness constraints (Clarke et al. 1999, p. 32). By contrast, models of experiments and models of data represent failures that might be due to both program faults (bugs) and hardware exceptions.

[1] Suppes (1962, 259) considers even more levels at the bottom of the hierarchy and corresponding to the experimental design conditions and to *ceteris paribus* conditions. As being involved in the process of the experiment set-up and of data collection, they will not be taken into consideration here.

We maintain here that a bottom-up approach is unattainable as well. State transition systems are used in formal verification to check whether desired behaviours belong to allowed behaviours; to do to this, the model is required to represent, in principle, all potential executions of the target program. This might even involve the representation of never-ending runs. On the contrary, models used to perform experiments on computational systems contain finite segments of those infinite paths and only a subset of runs needed to perform the required tests is represented. So it is not practicable to start with representations used in software testing to obtain state transition systems to be used to perform algorithmic verification.

Formal verification and testing are two distinct processes, in the evaluation of software systems, that are carried out independently. Code verification cannot be avoided in the construction of empirical theories of software systems: once a failure is observed by testing a system, what engendered that failure has to be identified, that is, testers are interested in *explanations* of the undesired observed behaviours. By only testing the system, it is not possible to determine whether observed failures are induced by hardware exceptions or faulty code lines.[2] The observation of incorrect runs of a program that is formally correct induces one to think that a problem arose with the implementation of the tested program; and in case of incorrect artefacts, formal methods enable one to isolate the error state among the instruction lines. Following Fetzer (1988), testing cannot be avoided as well: programs that are formally correct might result in incorrect executions both in case the developed models are not adequate representations of the verified program, and in case the implementing hardware is flawed by design errors or physical failures.

A semantic theory of a software system is a structure systematizing and justifying the attained knowledge about a studied class of software systems. Indeed, this is the aim of any scientific theory, being it expressed in a syntactic or a semantic way. To reword Goguen's (1992) remark, such theories are conceived in a *context of justification* that does not resemble the way theories are discovered. Computer science is characterised by many modelling activities of computational systems at different levels of representation, including the state transition system level, the programming language level, the hardware architecture level, the logic circuit design level, etc.

By excluding purely deductive top-down approaches in software verification, Goguen (1996) suggested that in-formal methods must be somehow involved in the process of evaluations of software systems. According to his view "any formalism is situated" and "without human intervention, a formalization may well be inadequate for its intended applications". So, even the most well defined formal approach requires a context for its interpretation. Nice examples for the limits of formalization come from requirements engineering, where documents are often vague or even deliberately misleading, but this informality has real advantages. Vagueness and

[2] Many debuggers simply face this problem by adding an exception handling in the tested program's code. Another strategy is to test the same program using a different implementation. (Ammann and Offutt 2008, 231)

ambiguity help to describe tradeoffs that need a lot of work to be resolved, usually at a latter stage of the software life cycle.

We address the informality demanded by Goguen in the *abductive process* of hypothesising models, at different levels in the semantic hierarchy, and of refining them until they provide successful predictions (Angius 2013). In formal verification, state transition systems are hypotheses concerning all potential executions of the represented software system; they are conceived by taking into consideration the program's instructions. Algorithmic verification is afterwards performed on those models and against the desired set of property specifications. Some specifications might be positively verified while others might be violated. Those results are model-based hypotheses whose predictive adequacy still have to be settled by testing the system. In case observed executions are not consistent with those predictions and the former are correct runs, the involved model is refined.

Computational paths of some given state transition system which are incorrect with respect to a property specification, and to which correspond any of the actual program's executions, are often called false negatives; whereas false positives are correct paths not representing concrete computations. Model refinements aim at deleting false negatives and false positives that are recognised as such while testing the software system. Both of them are usually produced by an inadequate granularity of the transitions of the state transition system involved (Clarke et al. 1999, p. 16). Granularity is higher when transitions occur between macro states to which correspond many actual states of the represented software system. And granularity is lower when transitions take place between states to which no actual state, nor set of states, correspond; rather, an actual system's state corresponds to a set of states in the model. Actual states are actual assignments to the program's variables and transitions occur when those variables are assigned new values, consistently with the program's instructions. Inadequate models may be refined by decreasing or increasing the granularity of the transitions. For instance, a false negative may be generated by a high granularity of transitions. Suppose a model checker detect an incorrect path from, say, state s_n to state s_m $(n,m \geq 0;\ n \neq m)$; suppose also that to state s_n in the model correspond states s_{n1} and s_{n2} in the system, and to state s_m in the model correspond states s_{m1} and s_{m2} in the system. Finally suppose that two executions are observed, one from state s_{n1} to state s_{n2}, and another one from state s_{m1} to s_{m2}. Clearly, there is no actual execution corresponding to the modelled transition from state s_n to state s_m: the latter is a false negative which can be removed by decreasing the granularity of the transition, that is, by modelling two distinct model transitions, one from state s_{n1} to state s_{n2}, and another one from state s_{m1} to s_{m2} (Clarke et al. 2000).

The discovery of models constituting semantic theories of computational systems goes through a *trial and error process* involving several refinements and human interventions. Statements holding true in a refined model, that is, logic formulas expressing the verified property specifications, are those discovered law-like statements the model makes true (Angius and Tamburrini 2011).

The same can be said about models involved in software testing. Data flow graphs hypothesise a set of error states and incorrect paths the tester believes might be executed by the program to be evaluated (Ammann and Offutt 2008). The system is subsequently tested according to that model, i.e., the tester uses the model to select executions to be observed. In case a represented failure is not detected among performed runs, the model is refined accordingly, that is, the counterexample path is removed from the hypothesised failures. And in case executed faults are observed which are not represented by some counterexample in the data flow graph, any corresponding false positive is removed to be replaced by such counterexample.

It should be noted how, both while refining state transition systems or data flow graphs, for a counterexample to be recognised as a false negative, a problem arises about when to stop testing the system. Indeed, for an hypothesised counterexample, not being observed during the testing phase is not a guarantee that it will not be observed in the future. This problem follows the 'Dijkstra's dictum': "Program testing can be used to show the presence of bugs, but never to show their absence" (Dijkstra 1970, p. 7). Model-based hypotheses concerning future computations of a represented software system are tested up to a certain time fixed by testers; they are assigned a probabilistic evaluation according to statistical estimations of future incorrect executions based on past observed failures (Littlewood and Strigini 2000). Accordingly, they assume the epistemological status of probabilistic statements that are corroborated by failed attempts of falsification (Angius 2014).

7.3 Modular Semantic Theories and Empirical Structuralism

Each module in a program can be represented by some state transition system including all the allowed transitions; and a semantic theory, in the form of the abstracting hierarchy described above, can be provided for each of those modules. Let us suppose to provide a semantic theory of a modular software system in terms of a set of representational hierarchies, each for any module in the program. Such theory would not be adequately predictive with respect to the target system's behaviours, the reason being that modules are not independent entities. Modules usually interact with each other, bringing about new computations that are not performed by any module in isolation. From a logical viewpoint, module interactions can be grouped into refinement, integration, and composition relations (Diaconescu et al. 1993). Informally, refinements are stepwise processes transforming an abstract – usually formal – specification into an executable lower level program.[3] Refinements follow logical rules of entailment from the one specification to the next more concrete one, which displays the same behaviour. 'More concrete', in this case, means be more effectively or more directly implemented. The composition (sum) of specifications is to "put together" two (or more) specifications usually written with

[3] Module refinements involved in the implementations of abstract specifications should be carefully distinguished from model refinements examined in the previous section and dealing with the discovery of semantic theories.

the help of the same formal language and create a new specification. Hence, the most straightforward application of composition is the potential creation of "large" specifications from "smaller" ones. By integration of specifications we mean a more abstract form of composition which can be realised over more than one formal language.

Let us consider a simple example of a library, liberally taken from (Guerra 2001; Sernadas et al. 1995). To build an initial specification for the library, some predicates are introduced, such as *available*, stating that a given item (a book) can be borrowed from the library; *taken*, expressing that the book has been borrowed; and *returned*, indicating that the book has been given back. These predicates allow to describe how the represented system (the library) ought to behave, in other words, they enable one to provide a specification, call it S, for the library. S can be expressed syntactically, by a set of statements $Sen(S)$, or semantically by a set of models $Mod(S)$ wherein those statements hold true. For instance, it should be required that only *available* book can be *taken* from the library; this can be expressed by Linear Time Temporal Logic (LTL) formula $\boldsymbol{G}(taken \rightarrow available)$. $\boldsymbol{G}$ is the temporal operator 'Globally' which, in this case, ensures that in any state (globally) of the transition system if *taken* holds then *available* must also hold. Another of such behavioural properties may be $\boldsymbol{G}(taken \rightarrow \boldsymbol{X}\neg available)$, stating that, for any state where *Taken* holds, in the next ($\boldsymbol{X}$) state *available* does not hold. This expresses that borrowed books are not available any more. But if a borrowed book is *returned*, it will be available again; formally: $\boldsymbol{G}(returned \rightarrow \boldsymbol{X}available)$.[4]

S can be refined by adding additional statements that render the description of the system more concrete; a refinement of S is a specialising specification S' which statements $sen(S') \supset sen(S)$ are a superset of the set of sentences of S. Books in a library can be reserved, making them un*available* but not borrowed yet. Supplementary predicates can be added, such as *suspended* and *resumed*, respectively expressing that a book cannot be borrowed even if it has not been *taken* and that the book is not reserved anymore. Refining statements include $\boldsymbol{G}(suspended \rightarrow \boldsymbol{X}\neg available)$ and $\boldsymbol{G}(suspended \rightarrow \boldsymbol{X}\ (\neg available\boldsymbol{U}resumed))$; the until operator $\boldsymbol{U}$ here requires that the book cannot be borrowed until it ceases to be reserved.[5] Notice that behaviours allowed by S are also allowed by S'.

In providing a set of specifications for a library, also users have to be considered, as well as their interactions with the system. In the object-oriented specification paradigm, specifications are obtained for each object; in the present case a specification for the user is required, call it T. Modelling and prescribing interactions between the library and the user signifies defining compositions between S (or S') and T (or integrations, in case the two specifications are expressed in different vocabularies). As predicates of T consider *takes*, *borrows*, and *returns*; some of the prescribing statements can be $\boldsymbol{G}(takes \rightarrow \boldsymbol{X}\ borrows)$, $\boldsymbol{G}(returns \rightarrow \boldsymbol{X}\neg borrows)$,

[4] $\boldsymbol{G}((available \wedge \neg taken) \rightarrow \boldsymbol{X}available)$; $\boldsymbol{G}((\neg available \wedge \neg returned) \rightarrow \boldsymbol{X}\neg returned)$; and $\boldsymbol{G}(returned \rightarrow \neg available)$ might be additional specification statements.

[5] Clearly, it must also hold that $\boldsymbol{G}\neg(suspended \wedge taken)$ and $\boldsymbol{G}\neg(resumed \wedge returned)$.

and $\boldsymbol{G}((borrows \wedge \neg returns) \rightarrow \boldsymbol{X}\, borrows)$.[6] Composing, say, S' with T means considering the interactions between behaviours imposed by S' and those imposed by T. For instance, when *takes* holds in any state of the model implementing T, *taken* must hold in the state transition system implementing S'. The same should be said about *returns* and *returned*. This shows how transitions of one model condition the transitions of the model of another module (in this case object) of the same software. This paper aims at specifying the logic relations holding among discovered models of different modules so that the semantic theory resulting from those models be adequately predictive with respect to the represented systems' executions.

Before that, it is worth asking whether the structuralist approach on scientific theories (Balzer et al. 1987) provides insights in the understanding of the modular semantic theories under consideration in this study. Indeed, in the structuralists program, empirical theories are semantically identified by the set of their models and a *theory-net* is introduced specifying intra-theoretical links among *theory-elements* in the net. A theory-element $T = \langle K,\ I\rangle$ is given by a so called theoretical core K and by the set of intended applications I. In a *theory-core* $K = \ \left\langle M_p,\ M, M_{pp},\ C,\ L\right\rangle$, the class of potential models M_p is a class of structures satisfying a set of axioms with no empirical content and defining the theory's basic concepts. Actual models in the class $M \subseteq M_p$ are models that, besides satisfying axioms satisfied by M_p, also satisfy the theory's laws having empirical content and being expressed with concepts defined in potential models. Axioms defining concepts coming from external theories are satisfied by the class of partial potential models M_{pp}. Constraints in the class $C \subseteq Po\left(M_p\right)$, belonging to the power set of set of potential models, connect models of the same theory-element, whereas links in the class $L \subseteq M_p \times M_p^*$ correlate partial models of two different theory elements T and T^* into a theory-net. Finally, intended applications are model-theoretically understood as well, as being in the class $I \subseteq M_{pp}$ (Moulines 1996).

Let us now turn to ask whether the structuralist concepts of *constraints* and *links* can be used to grasp the notions of refinements, integrations, and compositions between models in a modular semantic theory. The case of classical particle mechanics (CPM) provided by Balzer et al. (1987, 103–108) is considered here. Potential models of CPM are models identified by a (non-empty) set of particles, a set of time points, a set of space points, and by a position function, a mass function, and a force function assigning space points, masses and force values respectively to particles in the set.[7] Actual models of CPM are models that, besides satisfying

[6] Other statements are $\boldsymbol{G}(takes \rightarrow \neg borows)$, $\boldsymbol{G}(returns \rightarrow borrows)$, and $\boldsymbol{G}((\neg borrows \wedge \neg takes) \rightarrow \boldsymbol{X} \neg borrows)$.

[7] To use the formalism of Balzer et al. (1987, 30), m is a potential model of classical particle mechanics iff (i) $m = \ \langle P, T, S, c_1, c_2, s, m, f\rangle$; (ii) P, T, S are non-empty sets of (finite) particles, time and space points respectively; (iii) $c_1 : T \rightarrow \mathbb{R}$ and $c_2 : S \rightarrow \mathbb{R}^3$ are a time and a space bijective coordination function; (iv) $s : P \times T \rightarrow S$ is the space function; (v) $m : P \rightarrow \mathbb{R}^+$ is a mass function; (vi) $f : P \times T \times N \rightarrow \mathbb{R}^3$ is a force function.

the axioms made true by potential models, satisfy Newton's second law. Among the potential models of a given theory, one might be interested in models meeting some given property of interest. In the case of CPM, one such property can be that given two applications of the theory, i.e. two mechanical systems (such as two planets), and a particles belonging to both systems (a grave moving from one planet to the other), the assigned mass is the same in the two systems. This constraint, known as the equality constraint, simplifies the calculation of a particle motion between the two systems and can be settled by requiring, for two potential models m and $\overset{\circ}{m}$ and a particle p, that $m(p) = \overset{\circ}{m}(p)$. C is thus the class of potential models satisfying the desired constraints (Balzer et al. 1987, 44–46).

Potential models satisfy axioms defining all concepts used in the theory-element. Structuralists take into consideration the empirical positivism's distinction between theoretical and non-theoretical terms to isolate, among models in M_p, models in which only non-theoretical terms are involved.[8] Those terms are concepts coming from other, different, theory-elements and appear in partial potential models of the theory-element under consideration. Particles, time, and space are non-theoretical with respect to CPM. This means that they can be determined without the need of any actual model of CPM. Time and point space can be determined, for instance, by such theories as chronometry and physical geometry respectively (Balzer et al. 1987, 51–52). Partial potential models of CPM are thus models defining the particle, time, and space states together with the space function; mass and force function are omitted as being theoretical.[9] Non-theoretical terms in partial potential models of CPM are theoretical terms of partial models of some other theory-element. Intra-theoretical links in $L \subseteq M_p \times M_p$ allow one to use non-theoretical terms in a theory-element that are defined in a preceding theory-element.[10]

Let us ask whether constraints can define relations between models in a semantic hierarchy representing a program's module and whether intra-theoretical links are able to represent module interactions in a semantic modular theory. By considering the subset of models satisfying some property, constraints can be used to isolate all those structures that satisfy a given program specification. However, constraints do not represent logic relations, such as abstractions and refinements, among those models. A formalism able to map models at different levels of abstraction and of different types in order, for those models, to satisfy the same formulas, is still lacking.

As what concerns links, they are used to express specialization relations among theory-elements in a theory-net; links induce preorders among the actual models of specializing theory-elements; both potential and partial potential models are equal

[8] The introduction of the distinction between theoretical and non-theoretical terms, and their respective models, is, according to Suppe (2000), a neo-postivistic heritage preventing scientific structuralism from being consistent with the semantic view of theories.

[9] The class M_{pp} in CPM is given by models $m_{pp} = \langle P, T, S, c_1, c_2, s\rangle$.

[10] For instance, some link must be established between T and c_1 in partial potential models of CPP and the corresponding elements of potential models in chronometry theory.

in all theory-elements.[11] Given a theory-element, further laws can be added to those satisfied by its actual models, thus restricting the domain of intended applications. Actual models of CPM satisfy Newton's second law; according to Balzer et al. (1987) a theory-element specialization of CPM is NCPM (Newtonian Classical Particle Mechanics) obtained by adding, to the actual models of CPM, Newton's third law concerning the *actio-reactio* principle. NCPM is a specialization of CPM since Newton's third law is less universal than his second law, that is, there are applications of CPM in which the third law is not required. Further specializations of CPM include Hooke classical particle mechanics HCPM, whose actual models also satisfy Hooke's law; gravitational classical particle mechanics GCPM, whose actual models satisfy the law of gravitation; and the electrostatic classical particle mechanics ECPM, involving Coulomb's law. More in general, theory-elements, satisfying a physical law that has general intended applications, induce a *tree-like* theory-net of specialised theory-elements by adding further laws that restrict the domains of application (Balzer et al. 1987, 175).

Links are specializing relations which can express refinements of modules, insofar as they induce a preorder on actual models. Adding axioms to subsequent models from CPM to NCPM, and so on, is akin to considering further statements in specialising specifications from S to S', to S'', and so on. The first is a process of concretization of models, the second is a process of implementation of specifications. However, the more important relations of integration and composition are left out of this picture. Also, it is very unlikely that the preorder of models of a modular program displays a tree-like structure.[12]

Beyond all this, there is a main meta-theoretical and methodological reason at the base of the difference between the structuralists' aims and the present concern. The structuralist project pursues the objective of indentifying the structure of *existing* scientific theories, in terms of theory-nets consisting of theory-elements connected by links and constraints, and of underlining the relations of equivalence, specialization, and reduction among those theories. The far-reaching goal is that of reducing main theories to each other into a unificationsit, olsistic, picture. On the other hand, this study is involved in the discovery of new, *non-existing* theories. Specifically, the main objective is here that of getting from a collection of available computational models, representing different modules of a software system at several levels of abstraction, into a semantic theory of such system. Once one such

[11] Given two theory-elements T and T', σ is a specialization relations $T\, \sigma\, T'$ iff $M_p = M'_p$; $M_{pp} = M'_{pp}$; $M \subseteq M'; C \subseteq C'; L \subseteq L'; I \subseteq I'$ (Moulines 1996, 11).

[12] Structuralists maintain that global inter-theoretical relations are also given among theory-elements of different theory-nets, such as between models of CPM and models of collision mechanics (Balzer et al. 1987, ch. 6). This kind of links assumes the forms of specialization, reduction, equivalence, and approximation. Beside the fact that, again, these are not the types of relation needed in the construction of modular semantic theories, global inter-theoretical relations hold among different macro-theories, whereas the topic of concern here is the modular structure of each macro-theory.

modular semantic theory is available, the structuralist formalism might be used to reconnect theories of a class of programs into a theory-net.

7.4 Using Institutions to Build Modular Semantic Theories

The Theory of Institutions is an abstract model theory applied to software specification languages. It was introduced by Goguen and Burstall (1992) to face the vast number of formalisms characterising common specification activities. Based on category theory (Goguen 1991), Institutions abstract from both the syntax and the semantics of a given language to focus on the satisfaction relation of models. Institutions accomplish the Tarskian satisfaction condition requiring that truth is invariant under change of notation (Tarski 1944). In contrast with Barwise's (1974) abstract model theory, Institutions apply to any-order language and to many-sorted logics. They are used to formalise programs' specifications by providing syntactic and semantic descriptions of the programs' modules. By using common categorical constructs, Institutions allow one to connect "small" specifications to obtain "larger" specifications (Burstall and Goguen 1977).

Informally, an Institution is introduced by indicating a collections of signatures, i.e. vocabularies, one would like to utilize in the description of a piece of software; a collection of sentences of interest which are expressed by using the defined vocabularies; the set of models, each expressed within a given signature and satisfying those sentences; and a satisfaction relation which be independent from the chosen signature. All these classes are defined categorically by means of an Institution $I = (\boldsymbol{Sign}, \mathit{Sen}, \mathit{Mod}, \models_{\Sigma})$ wherein $\boldsymbol{Sign}$ is a category of signatures; $\mathit{Sen} : \boldsymbol{Sign} \rightarrow \mathit{Set}$ is a functor, that is, a mapping of morphisms, defining the set of sentences expressible with each signature Σ in $\boldsymbol{Sign}$; another functor $\mathit{Mod} : \boldsymbol{Sign} \rightarrow \boldsymbol{Cat}^{op}$ introduces the category of models satisfying the defined formulas; $\models_{\Sigma} \subseteq |\mathit{Mod}(\Sigma)| \times \mathit{Sen}(\Sigma)$ is a Σ-satisfaction relation construed as follows: given a model and a formula satisfied by that model, a morphism in the category of models maps from the model into the translation of the model in a different vocabulary in such a way that the translated model satisfies the translation of the formula (Goguen and Burstall 1992, 10).[13]

Institutions can be used to provide descriptions of modules in a program. Modules are represented by means of a so-called theory T over an Institution I. Once defined a given Institution, a theory $T = (\Sigma, E^{**})$ is introduced by choosing a profitable signature Σ and by formalising within Σ a set E^{**} of sentences describing

[13] Formally, this is expressed by stating that for each morphism $\phi : \Sigma \rightarrow \Sigma'$ in $\boldsymbol{Sign}$, each signature Σ in $\boldsymbol{Sign}$, each $f \in \mathit{Sen}(\Sigma)$, and each $m \in |\mathit{Mod}(\phi)|$, $m \models \mathit{Sen}(\phi)(e)$ iff $\mathit{Mod}(\phi)(m) \models e$ (Goguen and Burstall 1992, 10).

the module and that be closed under entailment. *Galois connections* enable one to determine the closed set of models M^{**} satisfying the theory's sentences.[14]

The defined Institution possesses many vocabularies apart from Σ and it thus allows to describe the same "abstract module" within different formalisms, the requirement being that morphisms be given between sentences of each theory or, equivalently, between models satisfying those sentences. In other words, the same "abstract module" can be described by different theories selecting different signatures, provided that each sentence in a theory is the translation of a sentence in the other theory. A given Institution I induces a category **Th** of theories, which objects are theories of a specified program's module, and which morphisms, known as *theory morphisms*, express relations holding between translating theories. It is worth noting how theory morphisms in **Th** can express, besides translational equivalence of theories, also abstractions and refinements, i.e. logic relations formalising software engineering techniques on module specification and programming (Sanella and Tarlecki 2012).

Objects and morphisms in the category **Th** of theories describing a given module can be used to define a semantic theory for that module. First, one can map, via Galois connections, a collection of corresponding models from the theories in the category. Note that each model is expressed with a different formalism in ***Sign*** (the signature of the corresponding theory); these models can be identified with the different typologies of models utilised in formal methods and testing in the evaluations of a program. Theory morphisms ensure that different models satisfy the same set of property specifications. Finally, one can consider data abstractions and refinements among models in an abstracting hierarchy by means of the opportune theory morphisms in **Th**.

Given a program encoded into several modules, one can preliminary represent each module by introducing an opportune Institution and indicating a category **Th** of theories over that institution. A modular semantic theory predicting the behaviours of the modular target system should include a series of Institutions, each providing a semantic theory for each program module, and a set of relations between models belonging to different theories. This can be formalised by considering the category **INS** of Institutions which objects are Institutions and which morphisms are the so-called *Institution morphisms* representing relations among Institutions (Goguen and Burstall 1992, 19–21). Institution morphisms are categorical constructs, such as colimits and pushouts, able to represent refinements, integrations, and compositions between couples of modules in a program (Diaconescu et al. 1993).

As already stated, integrations and compositions are the kinds of constructs of main interest in this study, insofar as they give rise to those unmodelled computations that should be predictable by modular semantic theories. Both integrations and

[14] Galois connections establish, in Institutions Theory, a duality between model classes and theories by means of which any Σ-theory T determines the class of Σ-models T° satisfying the Σ-sentences of T. And any Σ-model class V determines a Σ-theory V° containing all the Σ-sentences satisfied by models in the class V.

compositions of modules are represented by means of an *intermediary Institution*. In the former case, signatures, sentences, and models of the intermediary Institution are given by an opportune merging of the sentences, and models of the two integrating Institutions. Connections between sentences and models are obtained by considering appropriate Institution morphisms in **Ins** mapping from the models and the sentences of the integrating Institutions to the models and the sentences of the integrated Institution (Kutz et al. 2010, 44–45). In case of composition, signatures, sentences, and models of the composed intermediary Institution are mapped by chosen elements in the union sets of, respectively, the sets of signature, sentences, and models of the two composing Institutions (Kutz et al. 2010, 46–53).[15]

Consider the composition of specifications S' and T of the library exemplified in the previous section. An Institution $I^{S'}$ for specification S' is given by a category of signatures among which one is used in S'; in this case LTL. The set of sentences include the LTL formulas expressed above and others that can be expressed using LTL to describe the library system. A theory over the obtained Institution is properly given by the chosen signature, i.e. LTL, and the chosen LTL sentences. Galois connections determine the class of models, from $Mod\left(I^{S'}\right)$, satisfying those sentences. The same holds for I^T. The composition of S' and T is achieved by means of an intermediary Institution, call it $I_T^{S'}$, representing a virtual unifying object having the library and the user as sub-objects. To consider the interactions between *takes* and *taken*, and between *returns* and *returned*, two predicates describing the composed object are to be introduced, that is, *take* and *return*. The former holds in the composed model only when *takes* and *taken* hold in the composing models, and *return* is true in a state of the composed model only when both *returns* and *returned* are true in the corresponding states of the starting models. Institution morphisms are established from $I_T^{S'}$ to S' and from $I_T^{S'}$ to T; concretely, *take* is mapped to *takes* and *taken*, *return* to *returns* and *returned*. Most interestingly for the present study, new LTL formulas can be expressed in $I_T^{S'}$ and that hold in the intermediary models, such as $\boldsymbol{G}(take \rightarrow \boldsymbol{X}(\neg available \boldsymbol{U} return))$ and $\boldsymbol{G}(borrows \rightarrow \neg available)$.

The latter are statements describing computations coming from the interaction of the integrating or composing modules. Intermediary models should be included in a semantic modular theory of the whole software system. We propose to consider the category **Th** of theories over an intermediary Institution to build a semantic hierarchy for a modules' interaction. A model at the bottom of such hierarchy is a model of data representing the observed executions endangered by module interactions. A semantic modular theory can be given by the set of semantic hierarchies representing each module in the program and, for any two interacting modules, by a corresponding intermediary semantic hierarchy relating models of two interacting modules. To do this, we require that if a model, at any level of abstraction in a hierarchy representing a module, is related to a model of an

[15] For a technical treatment of integrations and compositions the reader may refer to (Kutz et al. 2010).

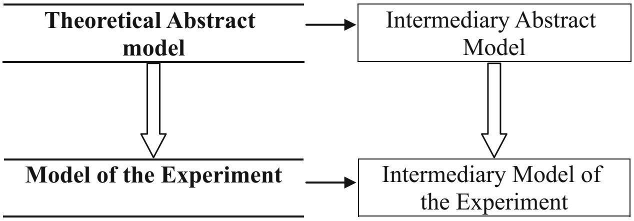

Theoretical Abstract model
Intermediary Abstract Model
Model of the Experiment
Intermediary Model of the Experiment

Fig. 7.2 Functors (*bigger arrows*) mapping Institution morphisms (*smaller arrows*) defining an intermediary abstract model and an intermediary model of the experiment. For the sake of simplicity, abstracting levels between the theoretical model and the model of the experiment have been omitted

intermediary hierarchy, the same relation should hold between models of the two hierarchies at any lower level of abstraction until models of data are obtained, as depicted in Fig. 7.2. Categorically, this can be formalised by establishing proper functors mapping Institution morphisms defining intermediary models at a given level in the representational hierarchy and Institution morphisms defining intermediary models at the immediate lower level. A modular semantic theory is, in this way, a net of theories related by intermediary models representing modules' interactions.

7.5 Concluding Remarks

Formal methods and software testing are involved in the hypothesis and the refinement of computational models, at different levels of abstraction, and for each module in the system to be evaluated. This paper is not involved in the improvement of any of those evaluation techniques. Rather, given the semantics of software verification and software testing, the philosophical aim is that of underlining what an empirical semantic theory of modular software system is.[16] By introducing an Institution and defining morphisms in the category **Th** of theories over that Institution, a semantic theory for each module in the program is provided. And by defining morphisms in the category **Ins** of module Institutions, a modular semantic theory representing interactions between couples of modules is constructed.

Interactions do not take place only *between* modules, but also *among* modules, that is, between group of modules. Executions arising from the interactions of two modules may, in turn, interact with other executions. Institutionally, this can be formalised by considering integrations and compositions among intermediary

[16] The epistemological and methodological analysis advanced in this paper runs alongside with the technical attempts of drawing formal specifications from non-formal descriptions of modules to be reused in software development (Pandita et al. 2012).

Institutions, giving thereby rise to bigger intermediary Institutions. The process can be in principle reiterated until one gets to an Institution able to describe the whole of the modular program. Indeed, the original aim of applying abstract model theory to specification languages was that of computing the specification of a system starting from smaller specifications. Directly providing a specification of non-trivial programs is quite an hard task.

An Institution defining a specification for the whole software system can be considered to provide also, via the category **Th** of theories over that Institution, a semantic theory for the whole program in the traditional sense, that is, defined in terms of a single hierarchy of abstracting structures. Each model in this hierarchy is a model of the entire system, as in the former approach of Angius and Tamburrini (2011). We maintain that the process, conceived in the work of (Burstall and Goguen 1977; Goguen 1991; Goguen and Burstall 1992), of getting from module specifications to software specification, resembles the process of discovery of semantic theories of modular software systems. Depending on the kind of predictions and explanation one is seeking, and on the observed executions one would like to model, the modularity of the built semantic theory can be decreased by computing bigger Institutions describing a higher number of potential computations, until a non-modular semantic theory is achieved.

Acknowledgments We are grateful to the three anonymous reviewers who advanced useful suggestions to improve the former draft of this chapter. This research has been co-financed by the European Union (European Social Fund- ESF) and Greek national funds through the Operational Program Education and Lifelong Learning of the National Strategic Reference Framework (NSRF) – Research Funding Program: THALES: Reinforcement of the interdisciplinary and/or inter- institutional research and innovation.

References

Ammann, P., & Offutt, J. (2008). *Introduction to software testing*. New York: Cambridge University Press.

Angius, N. (2013). Model-based abductive reasoning in automated software testing. *Logic Journal of the IGPL, 21*(6), 931–942.

Angius, N. (2014). The problem of justification of empirical hypotheses in software testing. *Philosophy and Technology, 27*(3), 423–439.

Angius, N., & Tamburrini, G. (2011). Scientific theories of computational systems in model checking. *Minds and Machines, 21*(2), 323–336.

Baier, C., & Katoen, J. P. (2008). *Principles of model checking* (Vol. 26202649). Cambridge: MIT Press.

Balzer, W., Moulines, C. U., & Sneed, J. D. (1987). *An architectonic for science: The structuralist program*. Dordrecht: Reidel.

Barwise, J. K. (1974). Axioms for abstract model theory. *Annals of Mathematical Logic, 7*(2), 221–265.

Burstall, R. M., & Goguen, J. A. (1977). Putting theories together to make specifications. In *Proceedings of the 5th international joint conference on artificial intelligence-volume 2* (pp. 1045–1058). San Francisco: Morgan Kaufmann Publishers Inc.

Clarke, E. M., Grumberg, O., & Peled, D. A. (1999). *Model checking*. Cambridge: The MIT Press.

Clarke, E. M., Grumberg, O., Jha, S., Lu, Y., & Veith, H. (2000). Counterexample-guided abstraction refinement. In *Proceedings of the 12th international conference for computer-aided verification* (Lecture notes in computer science*,* Vol. 1855, pp. 154–169). Berlin/Heidelberg: Springer.

Diaconescu, R., Goguen, J., & Stefaneas, P. (1993). Logical support for modularization. In *Proceedings of the second annual workshop on logical enviroments* (pp. 83–100). Cambridge: Cambridge University Press.

Dijkstra, E. W. (1970). *Notes on structured programming* (T. H.—Report 70-WSK-03). Mathematics Technological University Eindhoven, The Netherlands: Academic Press.

Fetzer, J. H. (1988). Program verification: The very idea. *Communications of the ACM, 31*(9), 1048–1063.

Fisher, M. (2011). *An introduction to practical formal methods using temporal logic*. Chichester/Hoboken: Wiley.

Goguen, J. A. (1991). A categorical manifesto. *Mathematical Structures in Computer Science, 1*(1), 49–67.

Goguen, J. A. (1992). The denial of error. In C. Floyd, H. Züllighoven, R. Budde, & R. Keil-Slawik Software (Eds.), *Development and reality construction* (pp. 193–202). Berlin/Heidelberg: Springer.

Goguen, J. A. (1996). Formality and informality in requirements engineering. *ICRE, 96*, 102–108.

Goguen, J. A., & Burstall, R. M. (1992). Institutions: Abstract model theory for specification and programming. *Journal of the ACM (JACM), 39*(1), 95–146.

Guerra, S. (2001). Composition of default specifications. *Journal of Logic and Computation, 11*(4), 559–578.

Hoare, C. A. R. (1969). An axiomatic basis for computer programming. *Communications of the ACM, 12*(10), 576–580.

Kutz, O., Mossakowski, T., & Lücke, D. (2010). Carnap, Goguen, and the hyperontologies: Logical pluralism and heterogeneous structuring in ontology design. *Logica Universalis, 4*(2), 255–333.

Littlewood, B., & Strigini, L. (2000). Software reliability and dependability: A roadmap. In *ICSE'00 proceedings of the conference on the future of software engineering* (pp. 175–188). New York, USA: ACM.

Moulines, C. U. (1996). Structuralism: The basic ideas. In W. Balzer & C. U. Moulines (Eds.), *Structuralist theory of science: Focal issues, new results* (pp. 1–21). Berlin: Walter de Gruyter.

Müller, P. (2002). *Modular specification and verification of object oriented programs*. Berlin/Heidelberg: Springer.

Pandita, R., Xiao, X., Zhong, H., Xie, T., Oney, S., & Paradkar, A. (2012). Inferring method specifications from natural language API descriptions. In *Proceedings of the 2012 international conference on software engineering* (pp. 815–825). Piscataway, NJ, USA: IEEE Press.

Sanella, D., & Tarlecki, A. (2012). *Foundations of algebraic specifications and formal software development*. Berlin/Heidelberg: Springer.

Sernadas, A., Sernadas, C., & Costa, J. F. (1995). Object specification logic. *Journal of Logic and Computation, 5*, 603–630.

Shapiro, S. (1997). Splitting the difference: The historical necessity of synthesis in software engineering. *Annals of the History of Computing, IEEE, 19*(1), 20–54.

Suppe, F. (1989). *The semantic conception of theories and scientific realism*. Chicago: University of Illinois Press.

Suppe, F. (2000). Understanding scientific theories: An assessment of developments, 1969, 1998. *Philosophy of Science, 67*, S102–S115.

Suppes, P. (1962). Models of data. In E. Nagel, P. Suppes, & A. Tarski (Eds.), *Logic, methodology, and philosophy of science: Proceedings of the 1960 international congress* (pp. 252–261). Stanford: Stanford University Press.

Tarski, A. (1944). The semantic conception of truth: And the foundations of semantics. *Philosophy and Phenomenological Research, 4*(3), 341–376.

Chapter 8
Introducing the Doxastically Centered Approach to Formalizing Relevance Bonds in Conditionals

Selmer Bringsjord, John Licato, Daniel Arista, Naveen Sundar Govindarajulu, and Paul F. Bello

Abstract Conditional reasoning is an important part of sophisticated cognition. Such reasoning is systematically studied in the sub-discipline of conditional logic, where the focus has been on the objects over which conditional reasoning operates (formulae, and the internals thereof). We introduce herein a new approach: one marked by a focus on the mental attitudes of the agents engaged in conditional reasoning. We specifically present our approach in connection with the challenge of rigorously capturing what seems to many to be a requirement for an adequate formalization of conditional reasoning: viz., that the antecedent and consequent in conditionals be *relevant* to each other.

8.1 Introduction and Plan

Every intelligent autonomous agent must presumably employ some form of *conditional reasoning* if it is to thrive in its environment. Even the third author's dog seems aided by the faculty to carry out such reasoning. Suppose Rupert is called from the tub to come over. He certainly seems to know that his coming (χ), combined with the conditional that if he does he's likely to endure his least-favorite chore (geting a bath $= \beta$), implies that he will receive a bath. Isn't that

The RPI contingent is greatly indebted to the Office of Naval Research for funding to Bringsjord, and all authors deeply appreciate the opportunity for Arista to present on conditional reasoning and relevance at IACAP 2014. In addition, the authors wish to express their gratitude to two anonymous referees for their penetrating comments, criticisms, and suggestions. Finally, the authors are deeply grateful for the leadership, guidance, wisdom, and diligence of editor Vincent Müller.

S. Bringsjord (✉) • J. Licato • D. Arista • N.S. Govindarajulu
Rensselaer AI & Reasoning (RAIR) Laboratory, Department of Cognitive Science, Department of Computer Science, Rensselaer Polytechnic Institute (RPI), Troy, NY 12180, USA
e-mail: selmerbringsjord@me.com

P.F. Bello
Naval Research Laboratory, 4555 Overlook Ave. SW, Washington, DC 20375, USA
e-mail: paul.bello@nrl.navy.mil

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_8

why he stares at his master and the tub from a distance, and stays right where he is? Unfortunately, as is well known, it has proved remarkably difficult to suitably model, formally, particular forms of conditionals used in reasoning carried out by intelligent autonomous agents of the biological variety, so that insight can be gained as to how one of the silicon variety, e.g. a smart robot, can be engineered.[1] Empirical confirmation comes from many sources; one is the lack of consensus, in the subdiscipline of conditional logic (classically introduced by Nute 1984), as to how the many types of conditionals out there in the minds and discourse of humans, are to be formalized. One of the chief, specific challenges currently blocking such consensus, and the challenge we focus upon in this short chapter, is that of figuring out how to ensure relevance between antecedent and consequent in conditionals.

For example, if → is the material conditional, one might model our canine example by some such sequence as

$$\begin{array}{ll} & \chi \\ & \chi \rightarrow \beta \\ \therefore & \beta \qquad\qquad \textit{modus ponens} \end{array}$$

But once we adopt this model we must accept the consequence that for any proposition ψ believed by Rupert, he can justify the belief that anything else he believes, or even arbitrarily assumes, *implies* ψ.[2] That seems rather implausible. Surely Rupert's reasoning about bath time involves some kind of intuitive *relevance* between χ and β. And this is all the more true at the human level, of course. In short, relevance seems to be a crucial part of what binds the dyads of conditional reasoning. But what does the relevance between these dyads consist in?

The pioneers of relevance logic, as well as their intellectual descendants, have looked for relevance in certain relationships between formulae, elements thereof (e.g., variables, relations, etc.), and meta-structures built from assembling and arranging such formulae. This approach is seen to be followed time and time again in any comprehensive survey of relevance logic; for example, see the masterful survey provided in Mares (2014). For example, the pioneers provided systems in which such formula-schemas as

$$\psi \rightarrow (\phi \rightarrow \psi)$$

[1] At the dawn of AI, Herb Simon blithely took *Principia Mathematica*'s material conditional off the shelf, and charged ahead in building LOGIC THEORIST (which, to much fanfare, automatically proved theorems from *PM*), and in declaring that human-level machine intelligence was just around the corner. Simon was of course rather too sanguine; and part of the problem, by our lights, was the "irrelevant" conditional he affirmed, without modeling the mental attitudes of relevant agents. (Yet the employment of formal logic as the vehicle of choice was wise.)

[2] This is implied by the theorem below, in conjuction with mild assumptions about canine epistemology quite outside the present investigation. That investigation centers on humans and human-level agents/robots, not dogs.

are *not* theorems. (This is of course an easy theorem in the standard propositional calculus.) But the clever blocks of such theorems were, and still are, far from the ratiocination of real and robust agents, and appear to be but syntactic prestidigitation, not moves that lead to the real engineering of real robots with the kind of flexible thinking that is needed in order to thrive in the real world. Accordingly, we introduce herein a new approach to the relevance in conditionals, one that takes explicit account of the mental attitudes of agents, and isn't limited only to deduction, and variants thereof, but rather includes analogical reasoning. We classify our approach as *doxastically centered*, because it explicitly factors in the beliefs of the agents doing the reasoning.

The sequel unfolds as follows. Next (Sect. 8.2), we give a brief overview of the expressive intensional logic that serves as the backbone of our new approach to relevance. We then (Sect. 8.3) first present our doxastically centered approach in the context of analogical reasoning, in connection, specifically, with subjunctive conditionals (Sect. 8.3). Next, in Sect. 8.4, we present our doxastically centered approach in connection with deductive reasoning, with special attention paid to material and so-called "strict" conditionals, but where the type of conditionals in question can in principle vary across all the types studied in conditional logic. A brief conclusion wraps up the paper.

8.2 Quick Overview of $\mathcal{DCEC}^*$

Our new approach to formalizing and mechanizing relevance is built upon the foundation of a well-established, computational, implemented first-order multi-modal logic, the *deontic cognitive event calculus* ($\mathcal{DCEC}^*$) (Bringsjord and Govindarajulu 2013; Bringsjord et al. 2014). $\mathcal{DCEC}^*$ is extremely expressive, in that regard well beyond even expressive extensional logics like first- or second-order logic (FOL, SOL), and *a fortiori* beyond the FOL-based Event Calculus axiom system (Kowalski and Sergot 1986) subsumed by $\mathcal{DCEC}^*$ (and used to model and track time and change), and indeed beyond traditional so-called "BDI" logics, as explained in Arkoudas and Bringsjord (2009). At least by the standards of philosophy of mind, merely *propositional* multi-modal logics are just not sufficiently expressive, and therefore $\mathcal{DCEC}^*$ is quantificational. For instance, such propositional logics cannot capture the difference between *de dicto* ($\approx$ "believing that" *simpliciter*), *de re* ($\approx$ "believing of some x that"), and *de se* ($\approx$ "believing of oneself that") belief; yet all three are crucial in philosophically sophisticated modeling of the human mind (Bringsjord and Govindarajulu 2012; Chisholm 1976). (The $*$ in '$\mathcal{DCEC}^*$' is a reflection of our following philosopher Castañeda (1999) on self-consciousness.) The versatility of first-order logic is very convenient, and is further augmented by such $\mathcal{DCEC}^*$ intensional constructs such as common knowledge, communication, and perception.

In addition, $\mathcal{DCEC}^*$ has a proof-theoretic semantics, as explained and defended in Bringsjord et al. (2014), as opposed to a possible-worlds semantics, an approach that is explicitly rejected and supplanted with the proof-theoretic approach.[3] The usual possible-world semantics (Fagin et al. 1995) are mathematically elegant and well-understood, and they can be a useful tool in certain situations (e.g. in security protocol analysis). But they are notoriously implausible from a philosophy-of-mind viewpoint.[4] The element of justification, for instance, which is central in our intuitive conception of knowledge in $\mathcal{DCEC}^*$ (in keeping with proof-theoretic semantics, the meaning of the knowledge operator (**K**) in $\mathcal{DCEC}^*$ is that it appears as a conclusion in proofs), is entirely lacking from the formal semantics of standard epistemic logic (e.g. see Goble 2001). Indeed, knowledge, belief, desire, intention, etc., all receive the exact same formal analysis in possible-world semantics. By our lights, that is simply not tenable.

To further inform the reader regarding $\mathcal{DCEC}^*$, we report that the semi-formal doxastic notation used by Smullyan in (1987) shares some features with the $\mathcal{DCEC}^*$, but the $\mathcal{DCEC}^*$ offers more operators, and a precise set of inference rules, which we show in Fig. 8.1.[5] The intensional operators currently available in the $\mathcal{DCEC}^*$ are intuitively named. The formulae $\mathbf{P}(a,t,\phi)$, $\mathbf{B}(a,t,\phi)$, $\mathbf{I}(a,t,\phi)$, $\mathbf{D}(a,t,\phi)$, and $\mathbf{K}(a,t,\phi)$, respectively, express agent a's **P**erceiving, **B**elieving, **I**ntending, **D**esiring, or **K**nowing that the formula ϕ holds at time t. $\mathbf{S}(a,b,t,\phi)$ expresses agent a having communicated (or "**S**aid") ϕ to agent b at time t; and $\mathbf{O}(a,t,\phi,\psi)$ holds when a is obligated to perform ψ at t if ϕ holds. Finally, $\mathbf{C}(t,\phi)$ expresses the familiar concept of common knowledge: ϕ here is known by all agents.

A few words on the contrast of $\mathcal{DCEC}^*$ with systems other than Smullyan's: LORA (Wooldridge 2000) is a multi-sorted language that extends first-order branching-time temporal logic with modal constructs for beliefs, desires, and intentions (drawing on the seminal work of Cohen and Levesque (1990), and particularly on the BDI paradigm that followed it (Rao and Georgeff 1999)), as well as a dynamic logic for representing and reasoning about actions. It does not have any constructs for perception or for common knowledge, and does not allow for the representation of events that are not actions. Its semantics for the propositional attitudes are standard Kripke semantics, with the possible worlds being themselves branching time structures. We are not aware of any implementations of LORA. CASL (Cognitive Agents Specification Language) (Shapiro et al. 2002) is another system which combines an action theory, defined in terms of the situation calculus, with modal operators for belief, desire, and intention. Like LORA, CASL does not

[3] Readers unfamiliar with proof-theoretic semantics are encouraged to begin with Prawitz (1972).

[4] In an apt assessment of the situation, Anderson (1983) wrote that epistemic logic "has been a pretty bleak affair." Fagin et al. (1995) describe various attempts to deal with some of the problems arising in a possible-worlds setting, none of which has been widely accepted as satisfactory.

[5] This is technically a specification of one particular dialect of $\mathcal{DCEC}^*$. Alert readers will note that operators for logical necessity and logical possibility are e.g. not included in the dialect shown here.

Syntax

$$S ::= \begin{array}{l}\text{Object} \mid \text{Agent} \mid \text{Self} \sqsubset \text{Agent} \mid \text{ActionType} \mid \text{Action} \sqsubseteq \text{Event} \mid \\ \text{Moment} \mid \text{Boolean} \mid \text{Fluent} \mid \text{Numeric}\end{array}$$

$$f ::= \begin{array}{l}
\mathit{action} : \text{Agent} \times \text{ActionType} \rightarrow \text{Action} \\
\mathit{initially} : \text{Fluent} \rightarrow \text{Boolean} \\
\mathit{holds} : \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean} \\
\mathit{happens} : \text{Event} \times \text{Moment} \rightarrow \text{Boolean} \\
\mathit{clipped} : \text{Moment} \times \text{Fluent} \times \text{Moment} \rightarrow \mathit{Boolean} \\
\mathit{initiates} : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean} \\
\mathit{terminates} : \text{Event} \times \text{Fluent} \times \text{Moment} \rightarrow \text{Boolean} \\
\mathit{prior} : \text{Moment} \times \text{Moment} \rightarrow \text{Boolean} \\
\mathit{interval} : \text{Moment} \times \text{Boolean} \\
* : \text{Agent} \rightarrow \text{Self} \\
\mathit{payoff} : \text{Agent} \times \text{ActionType} \times \text{Moment} \rightarrow \text{Numeric}
\end{array}$$

$$t ::= x : S \mid c : S \mid f(t_1, \ldots, t_n)$$

$$\phi ::= \begin{array}{l}
t : \text{Boolean} \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall x : S.\ \phi \mid \exists x : S.\ \phi \\
\mathbf{P}(a,t,\phi) \mid \mathbf{K}(a,t,\phi) \mid \mathbf{C}(t,\phi) \mid \mathbf{S}(a,b,t,\phi) \mid \mathbf{S}(a,t,\phi) \\
\mathbf{B}(a,t,\phi) \mid \mathbf{D}(a,t,\mathit{holds}(f,t')) \mid \mathbf{I}(a,t,\mathit{happens}(\mathit{action}(a^*,\alpha),t')) \\
\mathbf{O}(a,t,\phi,\mathit{happens}(\mathit{action}(a^*,\alpha),t'))
\end{array}$$

Rules of Inference

$$\frac{}{\mathbf{C}(t,\mathbf{P}(a,t,\phi) \rightarrow \mathbf{K}(a,t,\phi))}\ [R_1] \qquad \frac{}{\mathbf{C}(t,\mathbf{K}(a,t,\phi) \rightarrow \mathbf{B}(a,t,\phi))}\ [R_2]$$

$$\frac{\mathbf{C}(t,\phi)\ t \leq t_1 \ldots t \leq t_n}{\mathbf{K}(a_1,t_1,\ldots \mathbf{K}(a_n,t_n,\phi)\ldots)}\ [R_3] \qquad \frac{\mathbf{K}(a,t,\phi)}{\phi}\ [R_4]$$

$$\frac{}{\mathbf{C}(t,\mathbf{K}(a,t_1,\phi_1 \rightarrow \phi_2) \rightarrow \mathbf{K}(a,t_2,\phi_1) \rightarrow \mathbf{K}(a,t_3,\phi_3))}\ [R_5]$$

$$\frac{}{\mathbf{C}(t,\mathbf{B}(a,t_1,\phi_1 \rightarrow \phi_2) \rightarrow \mathbf{B}(a,t_2,\phi_1) \rightarrow \mathbf{B}(a,t_3,\phi_3))}\ [R_6]$$

$$\frac{}{\mathbf{C}(t,\mathbf{C}(t_1,\phi_1 \rightarrow \phi_2) \rightarrow \mathbf{C}(t_2,\phi_1) \rightarrow \mathbf{C}(t_3,\phi_3))}\ [R_7]$$

$$\frac{}{\mathbf{C}(t,\forall x.\ \phi \rightarrow \phi[x \mapsto t])}\ [R_8] \qquad \frac{}{\mathbf{C}(t,\phi_1 \leftrightarrow \phi_2 \rightarrow \neg\phi_2 \rightarrow \neg\phi_1)}\ [R_9]$$

$$\frac{}{\mathbf{C}(t,[\phi_1 \wedge \ldots \wedge \phi_n \rightarrow \phi] \rightarrow [\phi_1 \rightarrow \ldots \rightarrow \phi_n \rightarrow \psi])}\ [R_{10}]$$

$$\frac{\mathbf{B}(a,t,\phi)\ \ \mathbf{B}(a,t,\phi \rightarrow \psi)}{\mathbf{B}(a,t,\psi)}\ [R_{11a}] \qquad \frac{\mathbf{B}(a,t,\phi)\ \ \mathbf{B}(a,t,\psi)}{\mathbf{B}(a,t,\psi \wedge \phi)}\ [R_{11b}]$$

$$\frac{\mathbf{S}(s,h,t,\phi)}{\mathbf{B}(h,t,\mathbf{B}(s,t,\phi))}\ [R_{12}]$$

$$\frac{\mathbf{I}(a,t,\mathit{happens}(\mathit{action}(a^*,\alpha),t'))}{\mathbf{P}(a,t,\mathit{happens}(\mathit{action}(a^*,\alpha),t))}\ [R_{13}]$$

$$\frac{\begin{array}{l}\mathbf{B}(a,t,\phi)\ \ \mathbf{B}(a,t,\mathbf{O}(a^*,t,\phi,\mathit{happens}(\mathit{action}(a^*,\alpha),t'))) \\ \mathbf{O}(a,t,\phi,\mathit{happens}(\mathit{action}(a^*,\alpha),t'))\end{array}}{\mathbf{K}(a,t,\mathbf{I}(a^*,t,\mathit{happens}(\mathit{action}(a^*,\alpha),t')))}\ [R_{14}]$$

$$\frac{\phi \leftrightarrow \psi}{\mathbf{O}(a,t,\phi,\gamma) \leftrightarrow \mathbf{O}(a,t,\psi,\gamma)}\ [R_{15}]$$

Fig. 8.1 $\mathcal{DCEC}^*$ Syntax and Rules of Inference. This is one of four dialects of $\mathcal{DCEC}^*$. The *lower lefthand portion* of the figure, as most readers will quickly divine, gives the type of formulae allowed in this dialect, where *bolded* majuscule Roman letters are intensional operators. The figure also gives the formal grammar for the formal language $L_{\mathcal{DCEC}^*}$ of $\mathcal{DCEC}^*$. (Machinery in $\mathcal{DCEC}^*$ for encoding, by Gödel numbering, meta-mathematical concepts, is not shown)

have any constructs for perception or for group knowledge (shared, distributed, or common). Also, like LORA, the semantics of all intensional operators in CASL are given in terms of standard possible worlds.

In general, due to sheer space limitations, and the further fact that the chief purpose of the present paper is to introduce a new doxastically centered approach to relevance in conditionals, exposition of $\mathcal{DCEC}^*$ must, alas, rest at the point it has reached here, and the motivated reader is encouraged to consult the many references we provide in the present section. We now proceed to provide some additional information regarding not the "internals" of $\mathcal{DCEC}^*$, but instead its position in the general space of logical systems, laid out from a Leibnizian point of view.

AI work carried out by Bringsjord is invariably related to one or more logics (in this regard, see Bringsjord 2008), and, inspired by Leibniz's vision of the "art of infallibility," a heterogenous logic powerful enough to express and rigorize all of rational human thought, he can nearly always position some particular work he and (at least temporarily) likeminded collaborators are undertaking within a view of logic that allows a particular logical system to be positioned relative to

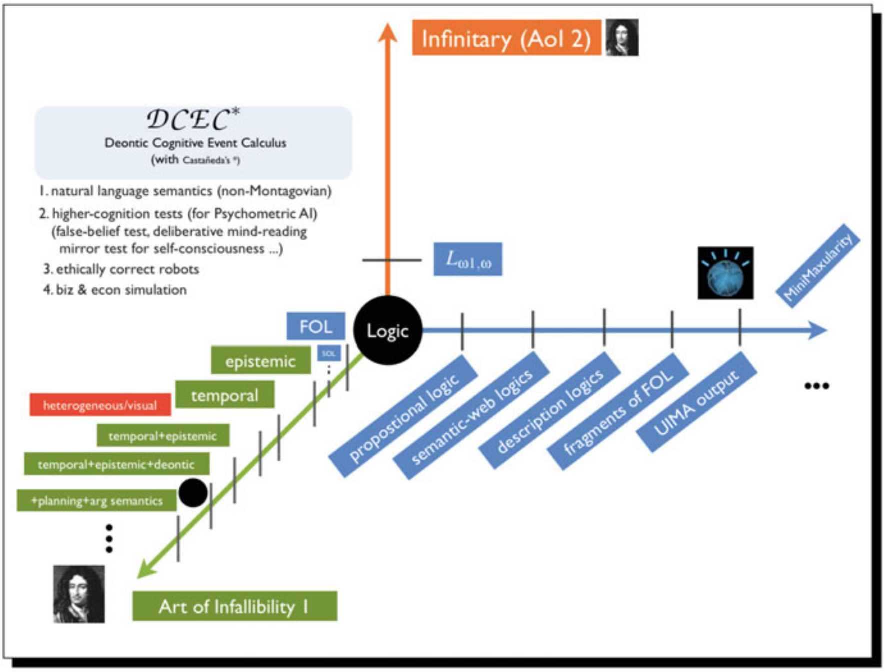

Infinitary (AoI 2)
$\mathcal{DCEC}^*$
Deontic Cognitive Event Calculus
(with Castañeda's *)
1. natural language semantics (non-Montagovian)
2. higher-cognition tests (for Psychometric AI)
(false-belief test, deliberative mind-reading
mirror test for self-consciousness ...)
3. ethically correct robots
4. biz & econ simulation
$\mathcal{L}_{\omega 1, \omega}$
MiniMaxularity
FOL
SOL
Logic
epistemic
temporal
heterogeneous/visual
temporal+epistemic
temporal+epistemic+deontic
+planning+arg semantics
propositional logic
semantic-web logics
description logics
fragments of FOL
UIMA output
...
Art of Infallibility I

Fig. 8.2 Locating $\mathcal{DCEC}^*$ in "Three-Ray" Leibnizian Universe

three dimensions, which correspond to the three arrows shown in Fig. 8.2. We have positioned $\mathcal{DCEC}^*$ within Fig. 8.2; its location is indicated by the black dot therein, which the reader will note is quite far down the dimension of increasing expressivity that ranges from expressive extensional logics (e.g., FOL and SOL), to logics with intensional operators for knowledge, belief, and obligation (so-called philosophical logics; for an overview, see Goble 2001). Intensional operators like these are first-class elements of the language for $\mathcal{DCEC}^*$, as we have seen. In contrast, only purely extensional logics short of or equal to full FOL traditionally appear along Ray 1 in Fig. 8.2; these are the logical systems that cover nearly all of contemporary AI, the Semantic Web, and recent AI success stories like Watson. For confirmation of this, see, respectively, Russell and Norvig (2009), Antoniou and van Harmelen (2004), and Ferrucci et al. (2010).

In the present paper, we will not make use of all of the features of $\mathcal{DCEC}^*$, which are broad and wide-ranging. The role provided by $\mathcal{DCEC}^*$ in our new approach to relevance in conditionals, in a word, is this: $\mathcal{DCEC}^*$ allows the doxastic, perceptual, and communicative aspects of this approach to be formalized, and mechanized.

8.3 Analogical Version of the New Approach

Assume that we have the following:

- A list of commutative predicates of the form $Assoc^*(\phi, \psi)$ where ϕ, ψ are formulae in $\mathcal{DCEC}^*$ and $Assoc^* \in \{No\_Assoc, Weak\_Assoc, \ldots, Normal\_Assoc, Strong\_Assoc\}$.
- A background pair $(\Phi, L_{\mathcal{DCEC}^*})$ consisting of axioms Φ expressed in formal language $L_{\mathcal{DCEC}^*}$ (see Fig. 8.1), where Φ may contain common knowledge, common beliefs, etc.

Suppose that agent a wishes to convince agent b that r holds, but may not immediately be aware of any $l \in \Phi$ such that $l \Rightarrow r$. a may therefore search algorithmically to find $l', r', l \in \Phi$ using kernel $\mathcal{K}$, such that $l' \Rightarrow r'$ and the formula $l' \Rightarrow r'$ is analogically similar to $l \Rightarrow r$. a would then be able to claim that it is plausible that $l \Rightarrow r$ holds, and might proceed to find a proof that $\Phi \vdash_{\tau} l \Rightarrow r$, or may communicate the tuple (l, r, l', r') to b directly.

A high-level algorithmic description of kernel $\mathcal{K}$ is as follows:

```
Given $(\Phi, L_{\mathcal{DCEC}^*}), r$, collect source analogs.
for each formula in $\Phi$ of the form $l' \Rightarrow r'$ do
    Determine the similarity score between $r'$ and $r$. This is defined as the weighted sum of all $Assoc^*(\phi, \psi)$ where $\phi \in \mathbf{W}$ and $\psi \in \mathbf{W}'$.
    The predicate used for $Assoc^*$ determines the relevant proposition's contribution to the similarity score.
    If the predicate $NoAssoc$ is used, the weighted contribution to the score may actually be negative.
    The pairs $(l', r')$ for which the highest similarity scores between $r'$ and $r$ were found are saved as the source analogs.
end for
for each of the source analogs $(l', r')$ do
    Perform analogical matching using $l' \Rightarrow r'$ as the source and $r$ as the target to produce $l$.
    if $\neg(l \in \Phi)$ then
        Discard this source analog.
    else
        Save $(l', r', l)$ as a potentially relevant triple.
    end if
end for
```

Every such potentially relevant triple produced by $\mathcal{K}$ can be presented by a. Note that $\mathcal{K}$ is essentially a source analog retrieval algorithm, which uses a static similarity metric between $\mathcal{DCEC}^*$ formulae captured by the $Assoc^*$ predicates. $\mathcal{K}$ is also loosely based on computational models of analog retrieval such as MAC/FAC (Forbus, Gentner and Law 1995) and LISA (Hummel and Holyoak 2003). Such computational models reflect the fact that human reasoners tend to use surface similarity to retrieve source analogs from long-term memory (Holyoak and Koh 1987; Ross 1989).

Because it is outside of the scope of this paper, we will omit the details of how analogical matching and inference can be used to match structures derived from the formula $l' \Rightarrow r'$ with smaller structures such as r, and to use the resulting matching to produce new formulae, as is done in $\mathcal{K}$. We refer interested readers to Gentner and Forbus (2011) for more details.

8.3.1 An Example of Kernel $\mathcal{K}$ in Action

Consider the case in which a wishes to explain r to b, where $r =$ "Today, c is avoiding b." Assume that part of Φ is the set of formulae l_1, l_2, l_3, the first two of which we present here in English for readability:

l_1 Last week, b was wearing an offensive shirt.
l_2 Last week, c avoided b.
l_3 $l_1 \Rightarrow l_2$

Kernel $\mathcal{K}$ would return (l_1, l_2) as a source analog, for the simple reason that l_2 has a high similarity with r. An analogical matching would produce the analogical inference l_i (= "Today, b is wearing an offensive shirt."), and if $l_i \in \Phi$, then $l_i \Rightarrow r$ would be proposed by a.

$\mathcal{K}$ therefore provides a with the suggestion that knowledge about an interaction yesterday is relevant to a (possibly subjective) judgment about a's t-shirt; such a determination of relevance may not necessarily be captured by an examination of l_i and r alone; rather, it requires the context Φ. In practical applications, there may be further analysis of the hypothesis $l_i \Rightarrow r$; for example, the hypothesis may be subjected to a rigorous deductive proof in a manner consistent with what has been called Analogico-Deductive Reasoning (ADR) (Licato et al. 2012, 2013).

However, the following subtle distinction must be made. Kernel $\mathcal{K}$ is not an algorithm which claims to find all possible $l \in \Phi$ relevant to any given r. $\mathcal{K}$ is simply a semi-formalized account of a psychologically plausible algorithm that is believed to be used by humans to find relevant knowledge.

$\mathcal{K}$ determines relevance using methods that can be imprecise and dependent on a notion of similarity which itself may be subject to unpredictable changes. It is entirely plausible, then, to assume that such an algorithm as $\mathcal{K}$ might declare two inconsistent possibilities to be potentially relevant. For example, Rupert may believe that there are two reasons his master is calling: the master wants Rupert to take a bath, or the master wants to provide Rupert with a treat. Assume that Rubert believes, from experience, that these two possibilities never occur at the same time. He might then believe that the two possibilities are inconsistent. But because $\mathcal{K}$ makes no attempt to compare the produced potentially relevant triples with each other, this inconsistency is not detected by $\mathcal{K}$. A process which does detect this inconsistency may, however, be added on to the end of $\mathcal{K}$.

8.4 Deductive Version of the Doxastic Approach

We now present a new form of *deductive* entailment in the same spirit as the doxastic-and-analogical scheme given in Sect. 8.3 for introducing subjunctive conditionals. This deductive entailment is based on a doxastic "link" between agents, one forged by virtue of the fact that they share an affirmation of a background logical system.

8.4.1 Standard Machinery Presupposed

Let Φ be some axiom system expressed in FOL, based on some formal language L (which is of course defined out of some alphabet and grammar); and assume some standard finitary proof theory τ in which proofs from Φ can be expressed. Following customary usage, we write

$$\Phi \vdash_{\tau} \phi$$

to say that ϕ is provable from Φ in τ; and we write

$$\Phi \vdash_{\pi_{\tau}} \phi$$

to say that ϕ is obtained by the particular proof π, which is expressed in τ.

8.4.2 Doxastic Context

Now we move to something new and, given our purposes herein, germane: a doxastically centered approach to relevance in conditionals and conditional reasoning. The basic idea is simple: one agent *a presents* a given proof to an agent b, where these agents have both understood and affirmed a "background" triple (Φ, L, τ). An overall presentation ${}_{a}\Pi_{b}$ from a to b includes not only the relevant proof in the transaction, but also the contexual triple. The presentation is valid only if both agents have such understanding, and have issued such an affirmation. The concepts of understanding and affirmation can be formalized in $\mathcal{DCEC}^{*}$, which was of course used above, but such formalization is outside the scope of the present paper; hence we leave the cognitive attitudes of a and b toward this triple of concepts unobjectionably informal. That said, we can provide part of what isn't a difficult recipe for the formalization: To start, the common-knowledge operator **C** in $\mathcal{DCEC}^{*}$ (see Fig. 8.1) should range over the core elements (e.g., formal language and proof theory) of the shared affirmation between agents involved in a presentation. In addition, the operators **S** and **P** should be employed as well.

As we explained in Sect. 8.2, the former can be read as "says," and the latter as "perceives." Clearly, the agents in presentations need to perceive the core elements in our framework, and would need to (at least tacitly) say something to each other about these elements. As an example, since a key element in the concept of sharing introduced here from a to b is the background proof theory τ, a given presentation will entail

$$\mathbf{B}(a,t,\phi(\|\tau\|)).$$

Here $\phi(\|\tau\|)$ is a formula expressing that the proof theory forming part of the basis for the presentation in question is τ, and $\|\tau\|$ is an Gödel encoding of that proof theory.

Note that the affirmation in a presentation needn't be explicitly communicated by the agents involved. For example, if one number theorist presents to another number theorist a new theorem T, and the two of them tacitly share an affirmation of not only an informal framework for articulating proofs, but the axiom system for Peano Arithmetic and its underlying formal language, that is enough for them to agree that it makes sense to use the material conditional, with its latitudinarian conception of relevance between antecedent and consequent. We write the presentation ${}_a\Pi_b$ from a to b as:

$$a \xrightarrow[\Phi \vdash_\tau \phi]{(\Phi,L,\tau)} b \tag{8.1}$$

The reader, given the account we have provided, will have no trouble seeing the role of the elements of the equation schema (8.1) just given. Note that since we don't insist that $a \neq b$, there is room for a solitary agent, perhaps engaged in an attempt to establish that some conjecture is in fact a theorem, to present proofs to herself. Indeed, while the scope of the present paper falls far short of calling for the setting out a logic of presentation, note that reflexivity of the presentation relation is in fact a theorem:

Theorem 1: $\forall a \; {}_a\Pi_a$

In addition, this relation is symmetric,

Theorem 2: $\forall a \forall b [{}_a\Pi_b \rightarrow {}_b\Pi_a]$,

but not (as desired) transitive. Using meta-reasoning, these theorems become provable in $\mathcal{DCEC}^*$. Along that line, we point out that the central conditional of Eq. 8.1 could itself originate in some higher-level logic which subsumes the proof theory τ. This is an option available to us different than the aforementioned use of Gödel numbering to encode meta-mathematical concepts within language and proof-search technology of $\mathcal{DCEC}^*$. For a sustained, detailed treatment of meta-reasoning (in connection with a doxastic/epistemic logic that is an ancestor of $\mathcal{DCEC}^*$), we direct technically inclined and curious readers to Arkoudas and Bringsjord (2005).

8.4.3 Example

No one can deny that in certain contexts, $\phi \to (\psi \to \phi)$, despite the fact that there needn't be any syntactic connection linking ϕ and ψ, is a perfectly acceptable theorem.[6] One such context is an introductory (classical) deductive logic class, the sort of thing that is taught across the globe year after year. And yet on the other hand, this theorem is taken to be one of the so-called "paradoxes of material implication" (and as such a cardinal motivator of conventional relevance logic; e.g. see Mares 2014) precisely because, as the story goes, there are instances of the schema in which relevance seems to be clearly violated.[7] How can this be? The doxastically centered concept of provability we have introduced provides an answer to this question: In the logic class, one agent a (e.g., the instructor) presents to another agent b (e.g., a student) a proof of the theorem, in a context in which both of these agents understand and affirm a certain system. One such system would be the language L_{PC} of the propositional calculus, the axioms Φ_{PC} for this logic, and some standard natural-deduction proof theory, say $\mathcal{F}$ from Barwise and Etchemendy (1999). In our notation, then, we have the following:

$$a \; \overset{(\Phi_{PC}, L_{PC}, \mathcal{F})}{\overrightarrow{\Phi \vdash_{\mathcal{F}} \phi \to (\psi \to \phi)}} \; b \tag{8.2}$$

8.4.4 Approach Subsumes Relevance Proof Theory

Readers familiar with longstanding proof-theoretic approaches to relevance logic will probably have already realized that the doxastically centered form of deductive entailment presented here subsumes these approaches. Consider for instance the exemplar of the relevance-logic approach constituted by Anderson and Belnap's (1975) natural-deduction proof theory for their logic **R**, which we here dub τ_r.

[6] Reality serves up any number of stark examples, given that classical mathematics is filled with theorems based in part on precisely what proof-theoretically sanctions this theorem. Empirical confirmation of this is at hand, esp. given that Bourbaki (2004) reasoned from axiomatic set theory to show that at least a major portion of classical mathematics, where in our machinery Φ is set to ZF and τ matches our assignment, flows from presentations by one agent to another.

[7] As many readers will know, there are numerous such schemata; e.g., $\neg\phi \to (\phi \to \psi)$.

The basic idea driving τ_r is straightforward: Formulae in proofs carry with them indicators expressing which assumptions are used in inferences; only inferences that obey certain constraints on indicators are permitted; and the constraints are engineered so as to preclude the deduction of such formulae as the paradoxes of material and strict implication. Obviously, since our framework promotes particular proof theories to the level of a meta-object affirmed by agents, it's easy to subsume **R**, and for that matter easy to subsume *any* of the relevance proof theories: **NR** (Meyer and Friedman 1992), **S** (Martin and Meyer 1982), etc. Since there is no consensus among relevance logicians as to which relevance logics are correct, our framework, which relativizes relevance to context formed by the mental attitudes of agents, seems promising, especially when the goal is the design and engineering of intelligent agents.

8.4.5 Remark: The Stunning Elasticity of Relevance

We observe that the following proposition is quite plausible:

> (+) For any conditional $c = l \Rightarrow r$, where '$\Rightarrow$' is a meta-variable that can be instantiated with any type of conditional studied in conditional logic,[8] and similarly l and r denote the generic antecedent and consequent, resp., there exists a context C in which l and r are relevant.

For instance, consider specifically

$$c' = \text{Snow is black.} \to \text{Quantum mechanics is bogus.}$$

where $\to$ is the material conditional. Many thinkers would declare c' to be a paradigmatic failure of relevance in a conditional. This diagnosis seems at first glance quite sensible, certainly. But obviously there are an infinite number of contexts in which the left and right sides of c' are clearly relevant. For instance, suppose c' is presented by agent a to agent b, after a points out that it's false that snow is black, and that in the propositional calculus any conditional with a false antecedent is truth-table true. a might be a logic teacher teaching the mechanics of the propositional calculus, for example.

One of the chief virtues of the new form of doxastically relativized deductive entailment presented in the present Sect. 8.4 is that this entailment is consistent with (+). As far as we can tell, all relevance logics with which we are familiar are unfortunately inconsistent with (+), or are at least not provably consistent with this proposition.

[8] A nice introduction to which, again, is provided in Nute (1984).

8.5 Conclusion: Next Steps

We make no such claim as that our doxastically centered approach is superior to prior work, only that it's fundamentally new. Readers must judge for themselves whether the approach is worth genuinely pursuing in the longer run. Such pursuit of course entails the next three steps: the full $\mathcal{DCEC}^*$-based formalization of the relationship between agents in the deductive approach, computational mechanization of both our analogical and deductive schemes, and then the evaluation of those implementations. Of course, if our project is really to go somewhere significant, we need to expand our approach to cover, eventually, *all* the conditionals studied in conditional logic, not just the subjunctive, material, and strict cases treated herein. In this regard, our near-term target: counterfactuals.

Finally, our project, we hope, is poised to make philosophical contributions as our more technical work advances. Fortunately, the connection our new approach makes between formal notions of relevance and the modeling of the psychological processes underlying analogical retrieval and inference is of great interest to the growing philosophical literature on analogy. Bartha (2010) notes that we have no substantive normative theory of analogical arguments, and his proposed solution, the Articulation Model, relies on a notion of "potential relevance" which is at least partially in line with the use of the same term in our Kernel $\mathcal{K}$. In addition, the contemporary literature on legal reasoning is just as dependent on the notion of relevance, as evidenced by Ashley and Bridewell (2010), Franklin (2012), Guarini (2004), and Macagno and Walton (2009).

References

Anderson, C. A. (1983). The paradox of the knower. *Journal of Philosophy, 80*(6), 338–355.

Anderson, A., & Belnap, N. (1975). *Entailment: The logic of relevance and necessity* (Vol. I). Princeton: Princeton University Press.

Antoniou, G., & van Harmelen, F. (2004). *A semantic web primer*. Cambridge: MIT.

Arkoudas, K., & Bringsjord, S. (2005). Metareasoning for multi-agent epistemic logics. In *Fifth International Conference on Computational Logic in Multi-Agent Systems (CLIMA 2004)* (Volume 3487 of Lecture notes in artificial intelligence (LNAI), pp. 111–125). New York: Springer.

Arkoudas, K., & Bringsjord, S. (2009). Propositional attitudes and causation. *International Journal of Software and Informatics, 3*(1), 47–65.

Ashley, K. D., & Bridewell, W. (2010). Emerging AI and law approaches to automating analysis and retreival of electronically stored information in discovery proceedings. *Journal of Artificial Intelligence and Law, Special Issue on e-Discovery, 18*(2), 311–320.

Bartha, P. F. (2010). *By parallel reasoning: The construction and evaluation of analogical arguments*. Oxford: Oxford University Press.

Barwise, J., & Etchemendy, J. (1999). *Language, proof, and logic*. New York: Seven Bridges.

Bourbaki, N. (2004). *Elements of mathematics: Theory of sets*. New York: Springer. This is a recent release. The original publication date was 1939.

Bringsjord, S. (2008). The logicist manifesto: At long last let logic-based AI become a field unto itself. *Journal of Applied Logic, 6*(4), 502–525.

Bringsjord, S., & Govindarajulu, N. S. (2012). Given the web, what is intelligence, really? *Metaphilosophy, 43*(4), 361–532.

Bringsjord, S., & Govindarajulu, N. S. (2013). Toward a modern geography of minds, machines, and math. In V. C. Müller (Ed.), *Philosophy and theory of artificial intelligence* (Volume 5 of Studies in applied philosophy, epistemology and rational ethics, pp. 151–165). New York: Springer.

Bringsjord, S., Govindarajulu, N., Ellis, S., McCarty, E., & Licato, J. (2014). Nuclear deterrence and the logic of deliberative mindreading. *Cognitive Systems Research, 28*, 20–43.

Castañeda, H. (1999). *The phenomeno-logic of I: Essays on self-consciousness*. Bloomington: Indiana University Press.

Chisholm, R. (1976). *Person and object: A metaphysical study*. London: George Allen and Unwin.

Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence, 42*, 213–261.

Fagin, R., Halpern, J., Moses, Y., & Vardi, M. (1995). *Reasoning about knowledge*. Cambridge: MIT Press.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, W., Nyberg, E., Prager, J., Schlaefer, N., & Welty, C. (2010). Building Watson: An overview of the DeepQA project. *AI Magazine, 31*(3), 59–79.

Forbus, K., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science, 19*, 141–205.

Franklin, J. (2012). Discussion paper: How much of commonsense and legal reasoning is formalizable? A review of conceptual obstacles. *Law, Probability and Risk, 11*(2–3), 225–245.

Gentner, D., & Forbus, K. (2011). Computational models of analogy. *Wiley Interdisciplinary Reviews: Cognitive Science, 2*(3), 266–276.

Goble, L. (Ed.). (2001). *The Blackwell guide to philosophical logic*. Oxford: Blackwell Publishing.

Guarini, M. (2004). A defence of non-deductive reconstructions of analogical arguments. *Informal Logic, 24*(2), 153–168.

Holyoak, K. J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory and Cognition, 15*(4), 332–340.

Hummel, J. E., & Holyoak, K. J. (2003). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review, 110*, 220–264.

Kowalski, R., & Sergot, M. (1986). A logic-based calculus of events. *New Generation Computing, 4*(1), 67–95.

Licato, J., Bringsjord, S., & Hummel, J. E. (2012). Exploring the role of analogico-deductive reasoning in the balance-beam task. In *Rethinking Cognitive Development: Proceedings of the 42nd Annual Meeting of the Jean Piaget Society*, Toronto.

Licato, J., Govindarajulu, N. S., Bringsjord, S., Pomeranz, M., & Gittelson, L. (2013). Analogico-deductive generation of Gödel's first incompleteness theorem from the liar paradox. In *Proceedings of the 23rd Annual International Joint Conference on Artificial Intelligence (IJCAI-13)*, Beijing.

Macagno, F., & Walton, D. (2009). Argument from analogy in law, the classical tradition, and recent theories. *Philosophy and Rhetoric, 42*(2), 154–182.

Mares, E. (2014). Relevance logic. In E. Zalta (Ed.), *The Standford encyclopedia of philosophy*. Spring 2014 ed.

Martin, E., & Meyer, R. (1982). Solution to the P-W problem. *Journal of Symbolic Logic, 47*, 869–886.

Meyer, R., & Friedman, H. (1992). Whither relevant arithmetic? *Journal of Symbolic Logic, 57*, 824–831.

Nute, D. (1984). Conditional logic. In D. Gabay & F. Guenthner (Eds.), *Handbook of philosophical logic volume II: Extensions of classical logic* (pp. 387–439). Dordrecht: D. Reidel.

Prawitz, D. (1972). The philosophical position of proof theory. In R. E. Olson & A. M. Paul (Eds.), *Contemporary philosophy in scandinavia* (pp. 123–134). Baltimore: Johns Hopkins Press.

Rao, A. S., & Georgeff, M. P. (1999). Modeling rational agents within a BDI-architecture. In *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*, San Mateo, CA, (pp. 473–484).

Ross, B. H. (1989). Distinguishing types of superficial similarities: Different effects on the access and use of earlier problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 15*(3), 456–468.

Russell, S., & Norvig, P. (2009). *Artificial intelligence: A modern approach* (3rd ed.). Upper Saddle River: Prentice Hall.

Shapiro, S., Lespérance, Y., & Levesque, H. J. (2002). The cognitive agents specification language and verification environment for multiagent systems. In *The First International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002)*, Bologna (pp. 19–26).

Smullyan, R. (1987). *Forever undecided: A puzzle guide to Gödel*. New York: Alfred A. Knopf.

Wooldridge, M. (2000). *Rational agents*. Cambridge, MA: MIT Press.

Chapter 9
From Silico to Vitro: Computational Models of Complex Biological Systems Reveal Real-World Emergent Phenomena

Orly Stettiner

Abstract Computer simulations constitute a significant scientific tool for promoting scientific understanding of natural phenomena and dynamic processes. Substantial leaps in computational force and software engineering methodologies now allow the design and development of large-scale biological models, which – when combined with advanced graphics tools – may produce realistic biological scenarios, that reveal new scientific explanations and knowledge about real life phenomena. A state-of-the-art simulation system termed Reactive Animation (RA) will serve as a study case to examine the contemporary philosophical debate on the scientific value of simulations, as we demonstrate its ability to form a *scientific explanation* of natural phenomena and to generate new emergent behaviors, making possible a *prediction* or hypothesis about the equivalent real-life phenomena.

9.1 Introduction

Computer simulations constitute a significant scientific tool for promoting scientific understanding of natural phenomena and dynamic processes in diverse disciplines, including biology. The need of culling significant knowledge and insights from vast amounts of empirical data, generated in recent decades about biological molecules and the millions of interactions among them, has promoted the development of innovative sophisticated computational methods and helped form new interdisciplinary research fields.

A group of researchers have developed over the last decade a computational approach termed Reactive Animation (RA) for simulating complex biological systems (Vainas et al. 2011). The dynamic characteristics of the biological objects are described based on cellular and molecular data collected from lab experiments. These data are integrated bottom-up by computational tools and methods to create a comprehensive, dynamic, interactive simulation (with front-end animated

O. Stettiner (✉)
Interdisciplinary Studies Unit, Science, Technology and Society (STS) Studies, Bar Ilan University, Ramat Gan, Israel
e-mail: orlyst@netvision.net.il

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_9

visualization) of biological systems behavior and development, in which the ‘simulationist’ may intervene on-line and observe in-silico on the artificial life-like system the effects of what may be considered as thought experiments.

In particular, the RA system was reported to have revealed several unexpected emergent properties, such as: (1) Competition between Thymocyte cells (part of the adaptive immune system) for sites of stimulation during their development (Efroni et al. 2007); (2) Novel features of lymphocyte dynamics, differentiation and anatomic localization (Swerdlin et al. 2008); (3) Formation of clusters of pancreatic cells that correspond well with clusters appearing early in the developing organ in-vivo (Setty et al. 2008); (4) Clear impact of two signaling factors’ expression levels on the structure and size of the pancreas during its morphogenesis (ibid). (5) Novel negative feedback loop in the regulatory network governing VPC fate specification during C. elegans vulval development (Kam et al. 2008).

These behaviors (and others) were not overtly preprogrammed in the molecular and cellular data integrated during the construction of the simulation and may be considered ‘weekly emergent’ phenomena (as termed by (Bedau 2013)). These discoveries consequently alerted biologists and prompted new lab experimentations of phenomena previously unknown. This renewed investigation of the real world is a process which, according to the researchers, highlights the explanatory power and the potential aid to experimentation offered by an animated interactive simulation of complex sets of data (Efroni et al. 2005). According to the developers, these models enable in-silico experiments at run-time and produce results that are similar to in-vivo experiments and suggest new intriguing hypotheses (Setty et al. 2010).

The RA system serves here as a case study for what may be considered a ‘full simulation’, a concept defined by (Humphreys 2009), as opposed to ‘core simulation’. It is a concrete computational device that correctly represents the structure of a real system as well as a temporal presentation of the model solutions representing the behavior and dynamic development of the real system.

The simulation construction is claimed (by its developers) to be a bottom-up process. Whereas top-down approaches stem from theory down to models and data production, the RA system stems from data (gathered experimentally about molecules, cells and discrete interactions that compose complex living systems), which is then systematically integrated to synthesize an accurate and comprehensive representation, serving as a model. In addition, the data put into the model is that of micro-scale molecules, and the simulation is expected to generate macro-scale emergent phenomena and behaviors, not originally programmed into the model.

However, we claim that a top-down direction is strongly integrated into the construction process, imposing environmental constraints, as well as theoretical and computational limitations, and it is the combined bottom-up/top-down process that enables the simulation to relate to real-life observations and become a scientifically-verified tool.

In this paper we initially review the contemporary philosophical viewpoints about the ability of computer models to scientifically explain real-life phenomena. We then investigate aspects of explanatory simulations construction process, which allow them to produce novel emergent behaviors, demonstrated through the RA modeling system.

9.2 Can Models Explain?

Scientific models and simulations are inevitably based on idealizations and abstraction of real-world entities and complex systems. Nevertheless, many philosophers of science believe that, under certain circumstances, such models may offer genuine scientific explanation and even prediction for real-world previously-unknown phenomena. Carl Craver (2006) claimed that models' explanatory power stems from their mechanism, and he distinguished between 'how possibly models' (which 'describe how a set of parts and activities *might be* organized such that they exhibit the explanandum phenomenon') and 'how actually models' (which 'describe real components, activities and organizational features of the mechanism that *in fact* produces the phenomenon'). While 'how possibly models' may be useful in constructing a space of possible mechanisms, the 'how actually models' begin with an accurate and complete characterization of the phenomena and show how the mechanism actually works.

Earlier, Ernan McMullin proposed the 'Hypothetico-Structural' account of explanation for models, suggesting that it is the structure of the model and the way in which its entities are combined, which constitutes the explanation (McMullin 1985). McMullin described a process of 'de-idealization', in which researchers may justify the explanatory power of a model by adding-back features that were deliberately omitted or assumptions that were too over-simplified during the model construction, while formally and theoretically justifying these corrections.

Woodward (following David Lewis (1973, 1986)) interpreted the explanation process as revealing information about patterns of *counterfactual* dependencies, being able to respond to 'what-if' questions, or '*what* sort of difference it would have made for the explanandum *if* the factors cited in the explanans had been different in various possible ways' (Woodward 2003, p. 11). This account was recently expanded by Bokulich (2011), who also required that the model should adequately capture the '*relevant features*' of the real world system (based on detailed empirical data). This step, she claimed, 'plays a central role in distinguishing between those models that are merely *phenomenological*, 'saving the phenomena', from those models that are *genuinely explanatory'*.

9.2.1 Simulations Are More Than Models

In recent years there has been an awakening among philosophers of science, seeking to clarify the role played by simulations and their epistemological standing within the space defined by theories, models and scientific experiments. Some claim that simulations are simply 'models', that cannot produce any novel knowledge, which had not been implicitly included within its base theory and assumptions (Eckhart 2010).

Others (including us) regard simulations as a unique scientific activity, which has '*a life of its own*' and consequently – deserves an epistemology of its own (Winsberg 2003, 2006). Winsberg claimed that simulations gain their credibility from the 'antecedently established credentials of the model building techniques employed by the simulationists' and the cumulative results gained by it, whereas they are close enough to the predicted results, based on real-world experiments. This view is supported by Morgan and Morrison (1999), who view simulations as '*autonomous agents*', theory-independent entities and their construction as a process which involves broad and diverse types of knowledge, intuition and inspiration. Others supporting this view include Humphreys (2009), Fox-Keller (2003) and Lenhard (2007).

The latter referred to the ability of simulations to scientifically explain and to provide a '*pragmatic mode of understanding*'. A simulation, he claimed, 'opens up a new mode of quantitative understanding, based on the deployments of epistemically opaque models whose behavior is made assessable by simulation' (Lenhard 2006). This is a mode of '*understanding by control*', through controlled intervention, manipulation of various components or parameters and through effective visualization of the dynamic results. As a result, simulations offer new instrumental access to phenomena, which can provide surprising predictions.

Mark Bedau (2008) referred to weakly-emergent phenomena, created through simulations, whose 'explanation works simply by tracing through the temporal details in the complex web of micro-level causal interactions that ultimately generate the macro-events'.

Following these philosophers (and others), we identify five characteristics of simulations (specifically of complex biological systems), which are necessary for their potential explanatory power. They:

1. Are based on actual, up-to-date *scientific theories and data* from experiments and observations.
2. Can dynamically *yield emergent phenomena* (either behaviors or structures), which may be recognized visually or computationally.
3. Allow in-silico experimentations through dynamic intervention, control and manipulations.
4. Include intensive *validation & verification* loops against *real-world* observations at *various hierarchy levels;* and
5. Make *testable biological hypotheses*, which can promote new in-vivo lab experiments.

Specifically, the two latter characteristics are those that 'make the difference', we claim, between *Phenomena-Generating simulations* ('how possibly', models that merely 'save the phenomena') and *Phenomena-Elucidating simulations* ('how actually', genuinely explanatory models).

The RA-based models are presented as an example for models that can scientifically explain certain biological complex phenomena and predict others.

9.3 Constructing an Explanatory Predictive Simulation

9.3.1 The Bottom-Up/Top-Down Conflict

Many models of complex systems are developed based on scientific theories and include complex mathematical structures, the equations of which are not analytically solvable. Computational simulations of such models are often expected to produce data about the time-dependent behavior of these systems. The simulations' construction process is then directed from the theory (top) downwards to the concrete (computerized) implementation, relevant to a specific physical system. In different cases, a solid broad theoretical framework might not exist, and scientists rather construct simulations from the bottom upwards. Data are collected from lab experiments and observations, focusing on specific aspects of the physical (biological) system, and the simulated model is developed to gain insights about the whole system's dynamic behavior. Philosophers of science have debated over the construction process and the way it affects simulations' ability to explain real-world phenomena and to produce novel and useful knowledge.

Eric Winsberg claimed that the knowledge produces by computer simulations is the result of inferences that are downward, in the sense that they are drawn from high theory down to particular features of phenomena (Winsberg 2001, 2009). Simulations are meant, he said, to *replace* experiments and observations as sources of data about the world, where real data are sparse. Winsberg proposed a hierarchical taxonomy of model types involved in top-down construction: Mechanical models, (concrete) Dynamic models, Ad-hoc models, Computational models, and finally- a 'model of the phenomena', which is a 'manifold representation that embodies the relevant knowledge gathered from all relevant sources about the phenomena', including the visualization and interpretation of the results (Winsberg 1999).

Bottom-up construction is often referred to as 'synthesis' or Inductive Inference. Various parts of the system are initially specified in detail (under constraints of existing knowledge and implementing technology). These parts are then joined together to form bigger entities, which are then connected to others, forming complex interactivity and mutual dependencies, finally adding up to the whole desired system.

Popular computerized techniques, such as Agent-based Modeling (ABM) or Cellular Automata (CA), are operating from the bottom-upwards. Out of local interactions between low-level entities and local pre-defined state transitions, new patterns and behaviors may be revealed in higher system levels:

> Situate an initial population of autonomous heterogeneous agents in a relevant spatial environment; allow them to interact according to simple local rules, and thereby generate- or grow up- the macroscopic regularity from the bottom-up. (Epstein 1999)

Bottom-up models, which are inherently nonlinear, enable synthesis and formation of complex, dynamic patterns or behaviors, and are popular in modeling behaviors of natural or social complex systems.

A major criticism against such modeling practice stems from the fact, that nearly every phenomenon may be produced in such a manner, regardless of its relevance to real-world systems. Stephan Lansing is quoted to have stated that:

> One does not need to be a modeler to know that it is possible to 'grow' nearly anything in silico without necessarily learning anything about the real world. (Richardson 2003)

Others also claimed against the feasibility and validity of pure bottom-up construction, saying that theory is essential to provide epistemological access to scientific phenomena and to explicate the data (Schindler 2007). The same inevitable conclusion was drawn following an intensively study of designing and engineering emergent systems, to generate desired complex global behaviors from simple local actions (Fromm 2005a, b, 2006): 'a bottom-up approach alone is not feasible… the number of combinations and configurations grows exponentially with the number of states, elements and rules'.

On the other side, a pure top-down (Macro-to-Micro direction) approach is definitely not enough, since it would be impossible to predict the opposite, micro-to-macro direction.

Consequently, combining both theoretical-methodological analysis and experimental-based synthesis seems to be necessary for creating a full explanation for the dynamics of a complex system (Weber 2002). Theory and experiments should be merged methodologically, in an *'explorative cooperation*', which enables simulations to reproduce realistic dynamics of known phenomena (Lenhard 2007). Constraints, limitations and assumptions should be imposed (from 'above') on the (bottom-up) evolving dynamic simulated self-organizing entities, in an ongoing loopy, recursive and adaptive process.

Most biological entities are 'complex' in the sense that they interact with and are influenced by entities of different ('higher' and 'lower') scales. In order to fully describe (and hopefully understand) a specific biological entity (e.g. cell), an integration of numerous sources of data is essential (from disciplines such as molecular biology, biochemistry, genetics – from the 'bottom', as well as cell biology, structural biology, developmental biology, evolution, physiology, anatomy and others – from the 'top').

Consequently, following Fromm (and others) we strongly support the claim that constructing an explanatory simulation of a complex (biological) system requires an *iterative two-way approach*, combining bottom-up synthesis processes, guided by top-down constraining feedbacks. Scientific low-level data about specific components of the system should be collected and integrated from the bottom upwards. These parts can then be joined together to form bigger entities, which are then connected to others, forming complex interactivity and mutual dependencies, finally adding up to a whole complex system. Concurrently, environmental and structural constraints, as well as high-level theoretical limitations, should be imposed from the top-downwards. It is the combined bottom-up\top-down process, which enables the simulation to relate to real-life observations and become a scientifically-verified tool.

9.3.2 Validation and Verification Against Real-World Data

The transition from a huge set of data into dynamic interactive computer models requires primarily the construction of an idealized conceptual model, based on the building block entities selected for modeling and on the desired scale of modeling (e.g. molecular, cellular etc.).

During this stage, missing elements – such as unknown values, transitions between untested scenarios, initial conditions – should be interpreted and creatively completed, based on the developers' instincts, intuition, inspiration or any other non-empirical reason.

Following classical model engineering methodologies, initially proposed in the 1970s and then extended (Sargent 2009), we propose a diagrammatic scheme for the construction of valid simulations, which tightly relate to the real-life simulated system and which can scientifically explain observed behaviors in that system. Such simulations may test in-silico new hypotheses and subsequently predict novel, previously unknown behaviors, which emerge computationally and may be subsequently tested in laboratories.

The simulation process is an ongoing iterative process, which involves validation and verification of the simulated results, by comparing them to data and scientific predictions based on experimental data. This iterative *trial-and-error* procedure includes feedback loops from the bottom upwards and from the top downwards. At each step, the simulated system's components, its parameters and inner architecture may vary and get re-designed repeatedly, until sufficient correlation is achieved with the simulated system, based on criteria determined by the designers.

Once specific architecture is set and parameters are fixed for a *'trial'*, the simulation is executed in what may be considered an 'in-silico experiment'. A bottom-up synthesis is taking place, where simulated components perform step-by-step execution as defined, interact with one another, values and states are modified, inputs are considered for calculations and etc. Out of the dynamics of the cumulative concurrent interactions and causal effects, 'finally' (at a specific time stamp defined by the observers) some new feature evolves, which is recognizable at a higher hierarchical level or scale. Simulated (numerical or visual) data are collected, analyzed and compared (computationally or visually) to the expected, lab-based results.

The gap detected in this comparison includes the *'error'* and decisions need to be made. As long as the gap does not satisfy the researchers, they need to join forces, expertise, creativity and skills to make the most efficient and practical modifications (of parameters, feedback loops, interactions, hierarchical levels, environmental conditions, entities definitions, algorithms and more). These modifications yield a new simulation – basis for a new *'trial'* scenario, and so on.

Verification and Validation steps should be integrated into the simulation at any possible link, specifically:

- Between the conceptual model and real-life data: Iteratively determine the computerized entities, interactions, modules, layers and parameters that faithfully represent the real-life system at hand.
- Between the computerized the conceptual models: making sure that the simulation implementation faithfully represents the conceptual model;
- Between the computerized model and real-life data, at two important paths:
  - Make sure that *all* known experimental data integrated into the simulation (at every hierarchical level relevant to the simulation), can be faithfully *reproduced* by it.
  - Test and verify newly identified (in-silico) emerging behaviors or phenomena (which may be considered *testable biological hypotheses*) against (newly suggested) lab experiments or real-life observations.

Simulations which inherently include such validation and verification loops within their two-way (bottom-up and top-down) paths are, we claim, *Phenomena-Elucidating simulations*, which may provide an actual scientific explanation for real-world complex behaviors and even predict new emerging behaviors (Fig. 9.1).

9.4 The Reactive Animation Simulation Environment

9.4.1 General

Over a decade has passed since David Harel, a leading computer scientist, presented an exciting vision:

> Our long-term aim is to model a full multi-cellular animal as a reactive system. Specifically, the C. elegans nematode worm, which is complex, but very well defined in terms of anatomy and genetics. The challenge is to construct a full, true-to-all-known-facts, 4-dimensional, fully animated model of the development and behavior of this worm (or of a comparable multi-cellular animal), which is easily extendable as new biological facts are discovered. (Harel 2003)

Such a model should be 'fully executable, flexible, interactive… which would help uncover gaps, correct errors, suggest new experiments and help predict unobserved phenomena'. In addition, it 'would be set up in such a way that biologists would be able to enter new data themselves as it is discovered, and even plug in varying theses about aspects of behavior that are not yet known, in order to see their effects.' (Harel 2005)

It has been a main goal of the research group to design simulations that can produce emergent phenomena, unknown behaviors that would raise speculations, which would be further tested through lab experiments.

An extensive computer simulation environment, termed 'Reactive Animation' (RA), was developed, based on the assumption that biological systems are 'large-scale complex systems that maintain an ongoing interaction with their environment and can thus be specified as reactive systems' (Harel and Setty 2008). Reactive

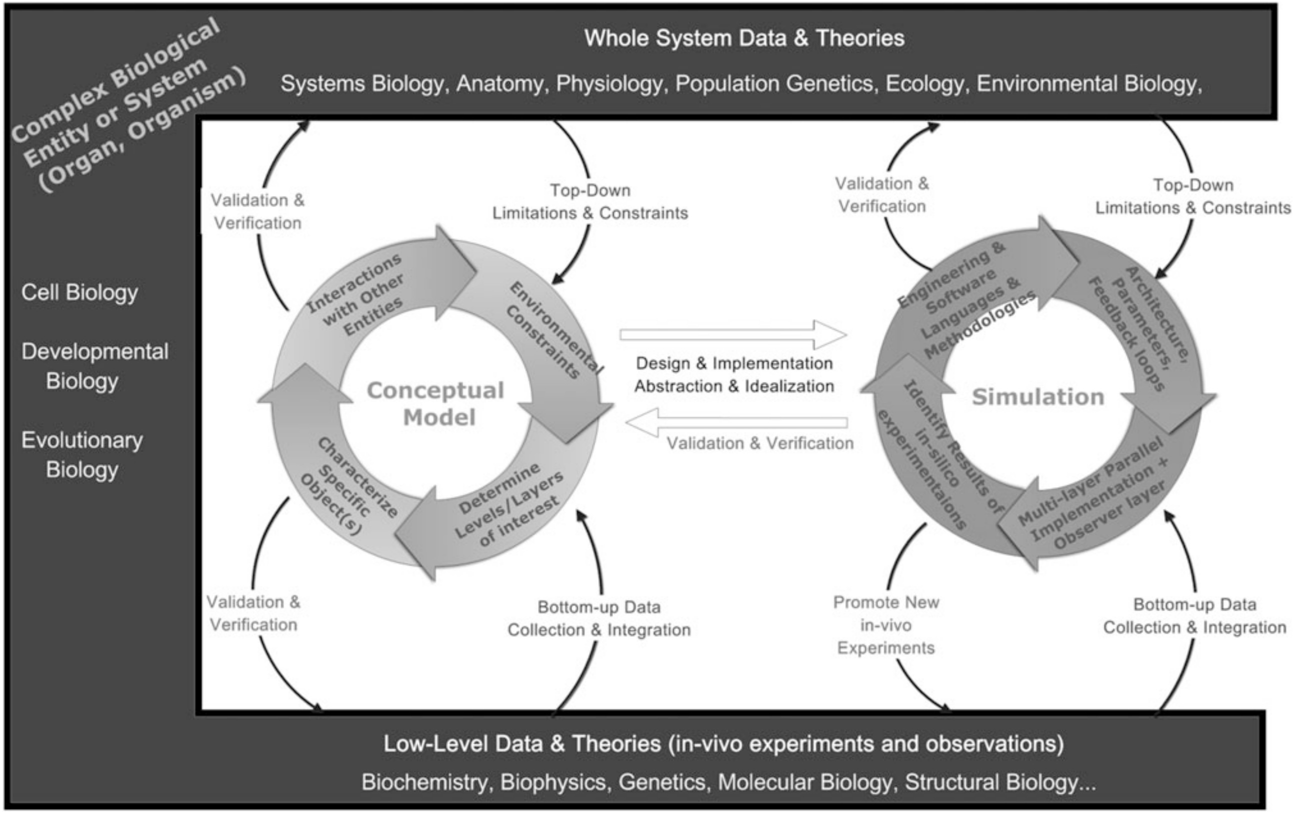

Complex Biological Entity or System (Organ, Organism)
Whole System Data & Theories
Systems Biology, Anatomy, Physiology, Population Genetics, Ecology, Environmental Biology,
Cell Biology
Developmental Biology
Evolutionary Biology
Validation & Verification
Top-Down Limitations & Constraints
Interactions with Other Entities
Environmental Constraints
Conceptual Model
Characterize Specific Object(s)
Determine Levels/Layers of interest
Validation & Verification
Bottom-up Data Collection & Integration
Design & Implementation
Abstraction & Idealization
Validation & Verification
Validation & Verification
Top-Down Limitations & Constraints
Engineering & Software Languages & Methodologies
Architecture, Parameters, Feedback loops
Simulation
Identify Results of in-silico experimentaions
Multi-layer Parallel Implementation + Observer layer
Promote New in-vivo Experiments
Bottom-up Data Collection & Integration
Low-Level Data & Theories (in-vivo experiments and observations)
Biochemistry, Biophysics, Genetics, Molecular Biology, Structural Biology...

Fig. 9.1 Schematic relationship between a real-world complex system, the conceptual model and the simulation. The validation and verification loops are inherently combined into the simulation at all scales and levels

systems are designed by exact and full specification of all possible reactions of the system (or parts within) to potential inputs or stimulations.

The methodology, initially based on 'state-charts' (Harel 1987), was chosen for its ability to describe a complex system's dynamic behavior, which is multi-scale, hierarchical and concurrent. The methodology was enhanced into a 'scenario-based' model, where 'scenarios' are statements based on 'if-then' logic, that document the *results* (usually phenotypic) of (in-vivo) experiments conducted under specific *conditions*. These data are based on extensive research and information resources, gathered from different aspects and disciplines relevant to the modeled system (e.g. anatomic, physiologic, genetic, molecular, cellular, etc.). Each statement is described by a 'Live Sequence Chart' (LSC), which represents specific conditions, known to result in some observed behavior for a specific system component. These modular and hierarchical mechanisms are interconnected by 'events' (that may be deterministic or random) and by 'objects' that may be referred to by multiple charts.

Information about the environmental conditions, physical constraints or boundary terms are imposed upon the modeled system, based on theory or whole-system prior knowledge.

RA system allows experimenters to intervene in the simulation and observe in-silico (through an integrated dynamic animation front-end) the results of 'thought experiments' ('what-if?' questions). The following section presents just a few of the emergent phenomena detected by the modelers.

9.4.2 Examples of Emerging Behaviors in RA Simulations

9.4.2.1 Model #1: Thymocyte Development

Data about Thymocyte development was used to generate an integrated dynamic simulation of the biological process termed 'T-Cell Education', through which stem cells reach mammalian thymus, undergo various modification and selection, until a small amount of candidate cells survives and becomes an essential part of the adaptive immune system (Efroni et al. 2005, 2007).

A systematic integration of cellular and molecular data was performed bottom-up into an accurate and comprehensive representation of the system. A specific goal of the researchers was to 'reveal multi-scalar unexpected emergent properties and to guide experimentation in thymocyte development'. Several such properties were observed and reported:

- The experimental mirco-scale molecular database (e.g. gene expression profiles, markers gradients etc.) integrated into the model was shown to suffice for generating (and thus validating) *realistic whole-organ, macro-scale thymocyte dynamic migration between functional anatomical locations*. Thymic fine anatomical structure was characterized according to 12 distinct developmental stages (http://www.wisdom.weizmann.ac.il/~dharel/ReactiveAnimation/demo.htm). The animation shows that immature cells proliferate at specific zones, while mature cells proliferate at anatomically different locations.

- Visualization of cell dynamics provided a view of emergent physiology, including the existence of *competition* between individual thymocytes for sites of stimulation, in order to engage in productive interactions with epithelial cells. This feature was unknown before (it cannot be seen in static histologic sections) and was therefore *not explicitly programmed into the simulation*. Rather, it was a behavioral derivative that emerged from the lower-level data that constituted the model.

  In-silico manipulation of various parameters suggested that such competition could comprise an important factor in three different emergent properties of the T-cell maturation process, and these suggestions promoted lab experimental validation efforts.
- RA's interactive nature makes it possible to *knock out molecules or cells* (in-silico) and observe the effects. Each knockout influenced the resulting thymus morphology in a different way, a phenomenon that RA made visible quite effectively. As the researchers reported (2005), 'Others have already experimentally validated in-vivo two of the three predictions we made after these simulations'.

9.4.2.2 Model #2: Pancreas Organogenesis

A fully executable, interactive, visual 4D simulation of the organogenesis[1] developmental stage of a mouse pancreas was developed, using the RA modeling system. Execution of the model provided a dynamic description of pancreas development, culminating in a structure that remarkably recapitulates morphologic features, seen in the embryonic pancreas (Setty et al. 2008, 2010).[2]

The model was designed in the cellular level, where pancreas cells were modeled as autonomous agents, sensing their environment and acting accordingly. The behaviors were determined based on bottom-up data, gathered from numerous lab experiments. In addition, top-down constraints were imposed, related to environmental entities (e.g. Extra-Cellular Matrix[3]), responsible for inter-cellular processes.

Several emergent properties revealed by the simulation were reported:

- The emerging 3D structure was compared against 2D histological sections of the pancreas at different stages, which revealed a close visual resemblance, indicating that the simulation captured quite well the pancreatic morphogenesis in mice.

[1] The term Organogenesis refers to the development of a functioning, anatomically specialized organ from a relatively small number of relatively undifferentiated precursor cells, and is critically influenced by factors involving multiple scales, dynamics, and 3D anatomic relationships (Setty et al. 2008).

[2] Animation: http://www.wisdom.weizmann.ac.il/~yaki/wisDay/index.html. Demonstrating movie: http://www.pnas.org/content/suppl/2008/12/17/0808725105.DCSupplemental/SM1.mov

[3] Extracellular Matrix (ECM) is a collection of extracellular molecules secreted by cells that provides structural and biochemical support to the surrounding cells.

- Dynamic interaction enabled to test the influence of modifying expression levels of specific regulating factors (as well as other environmental conditions, e.g. layout of blood vessels) on the pancreas morphology (the emerging 3D structure). The results were reported to become the subject of a collaborative testing, for an experimental (lab) validation.
- The simulated formation of clusters was observed, of pancreatic cells not expressing a specific gene (Pdx1), which 'corresponds well with the primary transition clusters appearing early in the developing organ in vivo'. The researchers emphasize that they 'did not have anything like this in mind when we started out, and the model was not explicitly programmed to do so'. Being able to trace the evolutionary origin of each cell within the simulation enabled the researchers to predict the nature of processes leading to such cluster formation, predictions which triggered new lab experiments (Setty et al. 2010).

9.4.3 Bottom-Up Data Integration with Top-Down Environmental Constraints

In the models presented above, dynamic characteristics of the biological system were specified based on facts and data gathered from numerous lab experiments reported in formal research biological papers. These data were translated and integrated bottom-up into modular and hierarchical 'statecharts' (which may generally be referred to as agents). The dynamic user interface enabled interactive intervention and manipulating of the biological simulated objects by the user, while visually observing the online consequences and effects. Thus, researchers could observe in-silico emerging behaviors (which were not specified or coded into the simulation) and predict the results of comparable future in-vivo experiments.

Computer models of living systems reflect their distributed, parallel and interactive nature. Simple agent-like autonomous entities interact locally, producing global emergent behaviors that evolved dynamically at higher system levels. As noted before, allowing the simulation to freely evolve bottom-up may result in behaviors not relevant to the actual living systems. Therefore, models must be constrained and limited according to existing scientific knowledge, imposed from the top downwards, so that emergent features would be restricted to the relevant solution space.

Accordingly, despite the RA developers' strong claim that the simulation is constructed bottom-up, we claim that a significant element of rational engineering design was included, based on a whole-system view of the biological objects, which inevitably posed top-down constraints. The entities selected by the researchers (as equal to specific biological entities) and the interactions between them, as well as the environmental representation, are all derived from system-level knowledge, from knowing the entire simulated organism (or organ) and decomposing it into distinct, isolated entities, under adequate abstractions and simplifications. The animated visualization system highly relates to the holistic systematic output, desired by the researchers. For example, detection of spatial physical separation between developing T-cells at different stages requires a detailed anatomic knowledge of the Thymus, which is the basis for the gentle design of the computerized simulated Thymus. Moreover, the computerized entities called ‘T-cells’ need to be allowed (by the simulationists) to move within the computerized space in pre-defined manner which relates to a-priori scientific findings. The visual output system should be designed to present such spatial movement in a way that has a meaning to biological observers. These considerations enforce constraints upon the simulation, which limit the space of potential emergent behaviors and make them verifiable.

9.5 Summary – Why are These Simulations Explanatory?

Simulations of biological processes often produce impressively visualized life-like behaviors, which may resemble certain aspects of the biological systems. For the most part, these simulated phenomena represent ‘Life as it could be’, hypothetical scenarios which are not testable in real life. The RA Simulation, we claim, represents ‘life as we know it’ (surely, under numerous limitations, such as the scale of modeling, which obviously does not currently go down to the sub-molecular chemical level; in addition, a high level of abstraction and simplification is inherently applied, due to computational and implicational constraints). Its credibility, capability to predict and explanatory power stem from its ability to accurately represent real biological phenomena, which are successfully validated against numerous sources of real-life data.

The RA developers state that:

> Tradition says, first understand, and then make a model to explain what you understand. RA (Reactive Animation), at least for complex living systems, turns the process around: first make a dynamic model, that integrates the data, then you will understand. A model that represents faithfully the dynamic crossing of scales and layers is itself an explanation of the living system’s emergent properties. (Cohen and Harel 2007).

The simulations were developed based on thorough, comprehensive knowledge of the biological systems, and entities were carefully defined based on real-world mechanisms, enabling the models to discover dynamic systematic features. This corresponds to Craver’s explanatory ‘how actually’ models category.

The RA models seem to grasp the actual structure (entities, interactions, environmental) and the characteristics which are causally responsible for the traits of the complex (needs to be explained) entity, and thus construes an explanation and a direction for further research, according to McMullin’s Hypothetico-Structural explanation account.

These simulations enable researchers to examine and observe dynamically the results of ‘what-if’ scenarios, in-silico experiments related to real-life system. This relates to Lenhard’s pragmatic account: The observed behavior is quantitative understood through the ability to produce the phenomena, to dynamically intervene and control its nature through the simulation, as well as predict novel, emergent unknown phenomena, which can be later confirmed (verified) through newly designed lab experiments.

The excessive validation steps integrated into the simulation environment provides a high level of confidence in the simulation by establishing a solid link between the (conceptual and computational) models developed and the real biological complex system.

This renewed investigation of the real world is a process which strongly supports and highlights the explanatory power and the potential aid of such simulations to biological research.

References

Bedau, M. A. (2008). Is weak emergence just in the mind? *Minds and Machines, 18*(4), 443–459.

Bedau, M. (2013). Weak emergence drives the science, epistemology, and metaphysics of synthetic biology. *Biological Theory, 8*(4), 334–345.

Bokulich, A. (2011). How scientific models can explain. *Synthese, 180*(1), 33–45. doi:10.1007/s11229-009-9565-1.

Cohen, I. R., & Harel, D. (2007). Explaining a complex living system: Dynamics, multi-scaling and emergence. *Journal of the Royal Society Interface, 4*, 175–182.

Craver, C. F. (2006). When mechanistic models explain. *Synthese, 153*(3), 355–376.

Eckhart, A. (2010). *Tools or toys?* Stuttgart: Institute of Philosophy, University of Stuttgart.

Efroni, S., Harel, D., & Cohen, I. R. (2005). Reactive animation: Realistic modeling of complex dynamic systems. *Computer, 38*, 38–47. doi:10.1109/MC.2005.31.

Efroni, S., Harel, D., & Cohen, I. R. (2007). Emergent dynamics of thymocyte development and lineage determination. *PLoS Computational Biology, 3*(1), e13. doi:10.1371/journal.pcbi.0030013.

Epstein, J. M. (1999). Agent-based computational models and generative social science. *Complexity, 4*(5), 41–60.

Fromm, J. (2005a). Ten questions about emergence. *arXiv:nlin/0509049v1 [nlin.AO].*

Fromm, J. (2005b). Types and forms of emergence. *arXiv:nlin/0506028v1 [nlin.AO].*

Fromm, J. (2006). On engineering and emergence. *arXiv:nlin/0601002 [nlin.AO].*

Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of Computer Programming, 8*, 231–274.

Harel, D. (2003). A grand challenge for computing: Towards full reactive modeling of a multicellular animal. *Bulletin of the EATCS, European Association for Theoretical Computer Science, 81*, 226–235.

Harel, D. (2005). On comprehensive and realistic modeling: Some ruminations on the what, the how and the why. *Clinical and Investigative Medicine, 28*(6), 334–337.

Harel, D., & Setty, Y. (2008). Generic reactive animation: Realistic modeling of complex natural systems. In *Proceedings of the 1st international workshop on Formal Methods in Systems Biology (FMSB'08) 2008a* (Lecture notes in bioinformatics, Vol. 5054, pp. 1–16). Springer: Springer Berlin Heidelberg.

Humphreys, P. (2009). The philosophical novelty of computer simulation methods. *Synthese, 169*(3), 615–626. doi:10.1007/s11229-008-9435-2.

Kam, N., Kugler, H., Marelly, R., Appleby, L., Fisher, J., Pnueli, A., et al. (2008). A scenario-based approach to modeling development: A prototype model of C. elegans vulval fate specification. *Developmental Biology, 323*, 1–5.

Keller, E. F. (2003). Models, simulation and "Computer Experiments". In H. Radder (Ed.), *The philosophy of scientific experimentation* (pp. 198–215). Pittsburgh: Pittsburgh University Press.

Lenhard, J. (2006). Surprised by a nanowire: Simulation, control, and understanding. *Philosophy of Science, 73*(5), 605–616.

Lenhard, J. (2007). Computer simulation: The cooperation between experimenting and modeling. *Philosophy of Science, 74*(2), 176–194. doi:0031-8248/2007/7402-0003$10.00.

Lewis, D. (1973). Counterfactuals and comparative possibility. *Journal of Philosophical Logic, 2*(4), 418–446.

Lewis, D. (1986). Postscripts to "Counterfactual dependence and time's arrow". In D. Lewis (Ed.), *Philosophical papers: Volume II* (pp. 52–66). Oxford: Oxford University Press.

McMullin, E. (1985). Galilean idealization. *Studies in History and Philosophy of Science Part A, 16*(3), 247–273.

Morgan, M. S., & Morrison, M. (1999). *Models as Mediators: Perspectives on Natural and Social Sciences*. Cambridge/New York: Cambridge University Press.

Richardson, K. A. (2003). On the limits of bottom-up computer simulation: Towards a nonlinear modelling culture. In *Proceedings of the 36th Hawaiian international conference on system science, IEEE, California, 2003.*

Sargent, R. G. (2009). Verification and validation of simulation models. *IEEE proceedings of the 2009 winter simulation conference* (pp. 162–176). Austin, Texas, USA.

Schindler, S. (2007). Rehabilitating theory: Refusal of the 'bottom-up' construction of scientific phenomena. *Studies in History and Philosophy of Science, 38*, 160–184. doi:10.1016/j.shpsa.2006.12.009.

Setty, Y., Cohen, I. R., Dor, Y., & Harel, D. (2008). Four-dimensional realistic modeling of pancreatic organogenesis. *Proceedings of the National Academy of Science, 105*(51), 20374–20379.

Setty, Y., Cohen, I., & Harel, D. (2010). Modeling biology using generic reactive animation. *Fundamenta Informaticae, 123*, 1–12. doi:10.3233/FI-2010-330.

Swerdlin, N., Cohen, I., & Harel, D. (2008). The lymph node B cell immune response: Dynamic analysis In-Silico. *Proceedings of the IEEE, 96*(8), 1421–1443. doi:10.1109/JPROC.2008.925435.

Vainas, O., Harel, D., Cohen, R. I., & Efroni, S. (2011). Reactive animation: From piecemeal experimentation to reactive biological systems. *Autoimmunity, 44*(4), 1–11. doi:10.3109/08916934.2010.523260.

Weber, M. (2002). Theory testing in experimental biology: The chemiosmotic mechanism of ATP synthesis. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences, 33*(1), 29–52. doi:10.1016/S1369-8486(01)00016-4.

Winsberg, E. (1999). Sanctioning models: The epistemology of simulation. *Science in Context, 12*(02), 275–292. doi:10.1017/S0269889700003422.

Winsberg, E. (2001). Simulations, models, and theories: Complex physical systems and their representations. *Proceedings of the Philosophy of Science Association, 68*(3), S442–S454.

Winsberg, E. (2003). Simulated experiments: Methodology for a virtual world. *Philosophy of Science, 70*(1), 105–125.

Winsberg, E. (2006). Models of success versus the success of models: Reliability without truth. *Synthese, 152*(1), 1–19. doi:10.1007/s11229-004-5404-6.

Winsberg, E. (2009). Computer simulation and the philosophy of science. *Philosophy Compass, 4*(5), 835–845.

Woodward, J. (2003). *Making things happen: A theory of causal explanation*. New York: Oxford University Press.

Part III
Philosophy of Cognition & Intelligence

Chapter 10
Why We Shouldn't Reason Classically, and the Implications for Artificial Intelligence

Douglas Campbell

Abstract In this chapter I argue that human beings should reason, not in accordance with classical logic, but in accordance with a weaker 'reticent logic'. I characterize reticent logic, and then show that arguments for the existence of fundamental Gödelian limitations on artificial intelligence are undermined by the idea that we should reason reticently, not classically.

10.1 Introduction

In this chapter I argue that human beings should reason, not in accordance with classical logic (CL), but in accordance with what I will call 'reticent logic' (RL). To see why we shouldn't reason classically, imagine two prisoners, locked in Cells A and B respectively. Each prisoner is given a list of sentences, and can choose whether to 'accept' sentences in the list. We can suppose a prisoner accepts a sentence by checking a box next to it. I am one of the prisoners. Initially I don't know whether I am in Cell A or Cell B, but I know it will be announced soon which cell I am in.

My list looks like this:

1: If I am Cell B's inmate, then Cell B's inmate will never accept 3. □
2: I am Cell B's inmate. □
3: Cell B's inmate will never accept 3. □

My aim is to accept only true sentences. (E.g., imagine 1 year will be deducted from my prison-sentence for each true sentence I accept, but a year added for each false sentence.) If Cell B's inmate was to accept 3, then 3 would be false, and so Cell B's inmate would have accepted a falsehood. Recognizing this, I resolve that

D. Campbell (✉)
Department of Philosophy, University of Canterbury, Christchurch, New Zealand
e-mail: douglas.campbell@canterbury.ac.nz

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_10

if it is announced that I am Cell B's inmate I will never accept 3.[1] My track record of following through on such resolutions is perfect. Hence I have good grounds for thinking that if I am Cell B's inmate, then Cell B's inmate will never accept 3. This is what 1 says. Accordingly I accept 1, by checking its box.

Next it is announced that I am Cell B's inmate, and so I check 2's box.

CL includes the rule of inference, *modus ponens*, which validates the inference from 1 and 2 to 3. Thus 1,2⊢3 (henceforth, the *prisoner's argument*) is classically valid. So were I to reason classically then I would, having accepted 1 and 2, also accept 3. However, I would falsify both 1 and 3 by accepting 3.[2]

What are my options? There appear to be four:

(i) I might accept 1 and 2, reason classically, and accept 3. This is a bad option, for as just seen it results in me accepting only one truth and two falsehoods.

(ii) I might accept 1 and 2, but refuse to accept 3, even though 3 is classically entailed by 1 and 2. This option is attractive, since it results in me accepting two truths and no falsehoods. However it means I must reason non-classically.

(iii) Foreseeing a trap, I might refuse to accept either 1 or 2 so CL won't push me into accepting 3. This option is unattractive for two reasons. First it results in me accepting only one truth, instead of the two truths I get to accept under option (ii). (If I am to refuse to accept a sentence I can plainly see to be true, better it be 3 rather than 1 or 2.) Second, both 1 and 2 might be classically entailed by other statements I can see to be true, creating a risk of escalation: to avoid being forced by CL into accepting 3, I might have to refuse to accept, not only 1 or 2, but numerous other true propositions from which 1 and 2 can be derived.

(iv) 3 is self-referential, and in this respect similar to the strengthened liar sentence ('This sentence is untrue'), which lacks coherent truth-conditions. It might be suggested on this basis that 3 lacks coherent truth-conditions too. If this were right then 3 wouldn't be classically entailed by 1 and 2, dissolving the problem. However, this option appears untenable. The strengthened liar sentence is paradoxical because any attempt to assign it a truth-value yields contradiction: the supposition it is true supports the conclusion it is untrue, and *vice versa*. In

[1] If the word 'never' raises intuitionistic worries about permanently undetermined truth-values, then it can be added as an extra stipulation that there is a time limit of 1 h for accepting sentences. The truth-values of 1 and 3 will be determined once and for all when the hour expires.

[2] Is the prisoner's argument a counterexample to *modus ponens*? No—or at least, not if by 'counterexample' we mean a case where both ϕ and $\phi \rightarrow \psi$ are true but ψ is false. The prisoner's argument is instead a case in which ϕ and $\phi \rightarrow \psi$ can both be true only if ψ is not accepted.

> contrast, neither the supposition that 3 is true nor the supposition it is untrue is contradictory. Rather, to suppose 3 is true is merely to suppose that Cell B's inmate never checks the third box on his list, while to suppose 3 is untrue is to suppose that Cell B's inmate will eventually check this box. 3's truth conditions are therefore unproblematic.[3]

Since (ii) is the best of these options, the prisoner's argument provides strong *prime facie* support for the idea that we shouldn't reason classically. But according to which logic should we reason, if not CL? This chapter is structured as follows. Section 10.2 introduces the key notion of a 'perverse argument'. Section 10.3 describes RL and argues we should reason reticently, rather than classically. Sections 10.4 and 10.5 showcase philosophical applications of the claim that we should reason reticently, with Sect. 10.4 critiquing a Gödellian argument against the possibility of an artificially intelligent machine knowing itself to be consistent, and Sect. 10.5 critiquing the famous 'mathematical argument' against artificial intelligence. Section 10.6 wraps things up.

10.2 Perverse Arguments

To 'accept' a sentence is to perform some mechanical action by which one endorses it as being true. For example, in the scenario just discussed the prisoner 'accepts' a sentence by ticking a box next to it. A formal system can be regarded as 'accepting' a sentence by proving it as a theorem. A person can be regarded as 'accepting' a sentence, ϕ, by believing ϕ (i.e., by loading ϕ into her 'belief box', as it were), or by saying, "ϕ is true" or "I accept ϕ". The notion of acceptance is intended to be a general one, having each of these other notions as special cases.

Notation Let $\Box\phi$ be shorthand for 'This system will ultimately accept ϕ' (or 'I will ultimately accept ϕ').[4] So, if a system accepts both ϕ and $\Box\phi$, then it thereby ensures that the latter sentence is true by accepting the former sentence. On the other hand, if it accepts $\Box\phi$ but never accepts ϕ, then in accepting $\Box\phi$ it accepts a falsehood.

With this notation in place, the prisoner's argument is revealed as having the following form:

[3] Lingering suspicions that 3 is liar-like should be put to rest by noticing that Gödel's (1931) diagonalization procedure for generating self-referential sentences with well defined truth-conditions can be used to manufacture a version of 3. See Sect. 10.5, below, for an explanation of how this procedure can be applied to English.

[4] I borrow the '$\Box$' notation from provability logic, wherein the intended meaning of '$\Box\phi$' is 'ϕ is provable in Peano Arithmetic'. In using this notion I don't mean to suggest that RL is a standard modal logic. (It isn't.)

A0. $(P \wedge Q) \rightarrow \neg\Box Q$
A1. $P \rightarrow Q$
A2. P

A3. Q

Here *P* stands for 'This system is Cell B's inmate'. *Q* stands for 'Cell B's inmate will never accept *Q*'. A0 isn't an explicit premise of the prisoner's argument, but is a tautological adjunct to the argument. It says, 'If this system is Cell B's inmate and Cell B's inmate will never accept *Q*, then it is not the case that this system will ultimately accept *Q*'.

A0, A1 and A2 together classically entail both *Q* and $\neg\Box Q$. That is, they classically entail both that *Q* is the case and that *Q* won't be accepted by the system. Let such arguments be called *perverse*. I.e., an argument is perverse iff: (a) its conclusion, ϕ, is classically entailed by its premises (i.e., the argument is classically valid); and (b) $\neg\Box\phi$ is also classically entailed by its premises. More generally, a proposition-set is perverse iff there is some ϕ such that the set classically entails both ϕ and $\neg\Box\phi$.

Perversity isn't to be confused with inconsistency. For example, the prisoner's argument's premises are perverse and yet clearly consistent (as can be seen by noticing that if I am Cell B's inmate and I never accept 3, then both 1 and 2 will be true).

Let $S \vdash \phi$ be some perverse argument. A system which reasons classically from *S* will commit a kind of fallacy—the 'perversity fallacy' as I shall call it. In accepting *S*, it is committed, on pain of having accepted a falsehood, to not accepting ϕ (since S entails $\neg\Box\phi$), and yet because it reasons classically and *S* classically entails ϕ, it *will* accept ϕ. Thus by accepting ϕ it ensures the falsity of *S*, *thereby undermining its grounds for concluding that* ϕ *is true in the very act of drawing this selfsame conclusion*. Such a classical reasoning system is like a moth flying in the dark near a candle. Just as the moth's method of navigation dooms it to the flame, so a system that reasons classically will blunder inevitably into error if a perversity lurks in the base of sentences it is reasoning from.

10.3 Reticent Logic (RL)

The idea behind RL is that to avoid succumbing to the perversity fallacy we should always do a 'perversity check' before accepting the conclusion of a classically valid argument. A reticent logic (RL) is simply a logic that includes a perversity check. Such a logic is 'reticent' in the sense that it 'holds back' in some cases when CL blithely accepts the conclusion of a perverse argument.

10.3.1 Basic RL

Basic RL classifies arguments as *reticently valid* or *reticently invalid.* $S \vdash \phi$ will be classified as reticently valid if these two conditions are satisfied:

(a) $S \vdash \phi$ is classically valid.
(b) $S \vdash \neg\Box\phi$ is not classically valid (i.e., $S \vdash \phi$ passes the 'perversity check').[5]

Otherwise $S \vdash \phi$ is classified as reticently invalid.

For example, although the prisoner's argument is classically valid, Basic RL classifies it as reticently invalid. This is because A0, A1 and A2 classically entail not only Q, but also $\neg\Box Q$.

Basic RL is weaker than CL, in the sense that while every reticently valid argument is classically valid, some classically valid arguments are not reticently valid. It can be thought of as being a logic of two parts, these being: (i) CL's methods for classifying an argument as classically valid or classically invalid; and (ii) a 'devalidating rule' that reclassifies perverse classically valid arguments as 'invalid'. In other words, it is a logic that sets the bar for validity higher than CL, by demanding not only that it be impossible for the premises to be true whilst the conclusion is false, but also that it be possible for the premises to be true whilst the conclusion is accepted.

10.3.2 Stepwise RL

By a 'logic' we usually mean not just a method for classifying arguments as valid or invalid, but a set of rules of inference that allow the conclusion of a valid argument to be derived from the argument's premises through a series of intermediate steps. Perversities might lurk at any step. A *stepwise RL* is a version of RL that performs a perversity check at each step. It consists of a set of *reticent rules of inference*, that differ from the classical rules by dint of having perversity checks built into them. For example, the classical and reticent versions of *modus ponens* differ from each other as follows:

Classical modus ponens: if both ψ and $\psi \rightarrow \phi$ are accepted, then accept ϕ.
Reticent modus ponens: if, (i) both ψ and $\psi \rightarrow \phi$ are accepted, and (ii) $\neg\Box\phi$ isn't classically derivable from any sentences that are already accepted, then accept ϕ.

Detecting whether condition (ii) is satisfied requires a meta-level test to be conducted, to see whether $\neg\Box\phi$ is classically entailed by the sentences accepted to date. Doing this meta-level test for classical validity will typically require invoking the ordinary, classical rules of inference multiple times. Hence the reticent

[5] The perversity check will be straightforward if the language is that of propositional logic or unary predicate logic, since it will then be decidable whether $S \vdash \neg\Box\phi$ is classically valid. For richer languages it will be necessary to make do with an incomplete perversity testing method, that errs by sometimes failing to classify perverse arguments as perverse. For every such method there will be a corresponding version of Basic RL, with its own strengths and weaknesses where its ability to detect perversities is concerned. The question as to which of such methods are 'best' is rich and complex, but I say no more about it here.

rules of inference presuppose the classical rules. One can therefore accept this chapter's thesis—that we should reason reticently rather than classically—while still maintaining that there remains a strong sense in which CL is the most fundamental logic.

10.3.3 Other Reticent Logics

More sophisticated and powerful reticent logics based on the following axiom and rule are possible:

Rule U: If any formula, ϕ, is accepted, then $\Box\phi$ may be accepted too.
Axiom V: $(\Box\phi) \to \phi$

Rule U is obviously well motivated. It enables a system that has accepted ϕ to accept it has done so—i.e., to accept $\Box\phi$. (It makes the system 'self conscious', so to speak.) It is 'truth preserving', since it will never directly cause a system to accept a falsehood.

Axiom V is similarly well motivated, for upon accepting $\Box\phi$, a system is committed, on pain of having accepted a falsehood, to accepting ϕ too. V lets the system discharge this commitment. The inference step from $\Box\phi$ to ϕ is truth preserving in the sense that a system that has accepted $\Box\phi$ has 'burnt its bridges' and can only hope to keep its set of accepted sentences free of falsehoods by accepting ϕ too.[6]

Limitations of space prevent me saying more about these logics here.

10.3.4 Classifying RL

RL is a non-monotonic logic, since adding $\neg\Box\phi$ to premises that reticently entail ϕ yields premises that don't reticently entail ϕ. It is a deductively incomplete logic since there can be a ϕ such that neither ϕ nor $\neg\phi$ is reticently entailed by the premises (as when the premises classically entail both ϕ and $\neg\Box\phi$).

RL has some resemblance to a modal logic. For example, Rule U amounts to a strengthened version of K's Necessitation Rule, and Axiom V is identical to modal logic's axiom M. However, RL doesn't respect K's Distribution Axiom, $\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$, and so it is certainly not a standard modal logic.

[6] From the fact that the system has accepted $\Box\phi$, it does not follow that ϕ is *true*. But it does follow that if the system fails to accept ϕ, then its risk of having accepted a falsehood is 100 %.

10.4 Can a Consistent Artificially Intelligent Machine Prove It Is Consistent?

The remainder of this chapter is devoted to showing that RL has important philosophical applications. Consider Argument G:

G1. *F* is consistent
G2. If *F* is consistent, then *F* will not prove *G(F)*
G(F). *F* will not prove *G(F)*

Here *F* denotes some formal system, and *G(F)* is an English rendering of *F*'s Gödel sentence. Gödel (1931) showed that, provided *F* uses various classical rules of inference such as *modus ponens* (see, e.g., Raatikainen 2014) and encodes elementary arithmetic, then if *F* proves its own consistency (i.e., if it proves G1) it will be driven, by Argument G, to accept *G(F)*, from which it follows that *F* is actually inconsistent. Since digital computers amount to instantiations of formal systems, this result – Gödel's second incompleteness theorem – can be taken (see, e.g., Gaifman 2000) as implying that no artificially intelligent digital computer can prove its own consistency except on pain of inconsistency.

I believe that Gödel's second incompleteness theorem has no such implications. To see why not, let us analyze 'proof' as being a species of 'acceptance', and use '$\Box\phi$' to represent the claim, '*F* will prove ϕ'. Argument G can then be formalized as follows:

G1′. $Con(F)$
G2′. $Con(F) \rightarrow \neg\Box G'(F)$
$G'(F)$. $\neg\Box G'(F)$

This argument is perverse, since its premises entail both $G'(F)$ and (same thing) $\neg\Box G'(F)$. Needless to say this perversity is crucial to Gödel's argument, since his strategy for proving the second incompleteness theorem hinges entirely on the idea that *F* will falsify Argument G's conclusion, and thereby falsify one of the argument's premises (namely, G1) in the very act of proving this selfsame conclusion.

Let us suppose that *F* is an artificially intelligent system that reasons reticently, rather than classically. In this case Gödel's second incompleteness theorem will not apply to it. While the theorem applies to any formal system that satisfies certain, modest requirements, one of these requirements is that the system uses classical logic. If *F* does not obey the various rules of classical logic, including *modus ponens*, then *F* need not be driven by the logic it is using from proving G1 and G2 to proving *G(F)*. Indeed if – as we are imagining – the non-classical logic used by *F* is RL, then *F* certainly *will not* prove *G(F)* after accepting G1 and G2: for it will instead recognize that Argument G is perverse and refuse to prove *G(F)* for this reason. Because it won't prove *G(F)*, it won't be caused, by its having proved G1 and G2, to undermine its own consistency by proving *G(F)*. Thus – at least for all Gödel's argument shows – it is entirely possible for such a system, which reasons reticently rather than classically, to prove both G1 and G2 (and thus prove its own consistency) without thereby tumbling into inconsistency.

For the reasons just given, it appears that consistent artificially intelligent computers will be unable to prove their own consistency only if they must reason classically, rather than reticently. But why couldn't an artificially intelligent machine reason reticently? Why not indeed! It is surely plausible that any machine that is truly 'intelligent' will be capable of recognizing whether, in accepting various premises, it has committed itself to not proving a conclusion that follows classically from these premises, and of refusing to prove the conclusion in such cases. That is, machines that are genuinely intelligent will surely not be prone to succumbing to the perversity fallacy. They will use RL, not CL.

10.5 A Rebuttal of the Mathematical Argument Against Artificial Intelligence

The 'mathematical argument' against artificial intelligence (Gödel 1951; Nagel and Newman 1957; Lucas 1961, 1996; Penrose 1989, 1994, 1996) purportedly shows that the theorem-proving abilities of the human mind cannot be matched by a computer. There is widespread agreement among philosophers and mathematicians that the argument is defective, but less agreement as to why. In what follows I first consider several stock rejoinders to the argument, and show that the argument can be patched to avoid them. Next I contend that the real problem with the argument involves a perversity fallacy within it.

Let the mathematical argument's 'protagonist' – referred to in the first person – be some human mathematician. Let *F* be some formal system (or programmed digital computer). 'I am *F*' is the conjecture that the protagonist's sentence proving dispositions match *F*'s. Let this conjecture be called the 'identity hypothesis'. The original version of the mathematical argument, found in Lucas (1961) and Penrose (1989), may be summarized as follows:

H1. I am consistent (i.e., I won't, for any sentence ϕ, prove both ϕ and $\neg\phi$).

H2. If I am *F*, then *F* is consistent. (From H1.)

H3. If *F* is consistent, then I can prove *F* is consistent.

H4. If I can prove *F* is consistent then (by invoking Gödel's first incompleteness theorem) I can know that *G(F)* is true.

H5. If I can know that *G(F)* is true, then I can prove *G(F)* without compromising my consistency.

H6. If I am *F*, then I can prove *G(F)* without compromising my consistency. (From H2 to H5.)

H7. If I am *F*, then I cannot prove *G(F)* without compromising my consistency. (From Gödel's first incompleteness theorem.)

H8. I am not *F*. (From H6 and H7, by *reductio*.)

The most glaring point of weakness in Argument H is H3. Many authors (e.g., Putnam 1960; Bowie 1982; Barr 1990; Boolos 1990; and Gaifman 2000) have pointed out that there is ample room to imagine that: (i) the protagonist's sentence-producing powers might be equivalent to those of some consistent formal system, F; but that (ii) the protagonist might be unable, because of F's great complexity, to prove that F is consistent.

Penrose has developed an ingenious new version of the mathematical argument that sidesteps this problem (1994, pp. 179–188; 1996). It is sometimes called Penrose's 'new argument', but I will call my formulation of it 'Argument J'.[7] It is based on the idea that we need not require the protagonist to prove that F is consistent 'from the ground up', as it were, because we can instead start from the assumption (contained in H1) that the protagonist herself is consistent, and then cantilever sideways from this starting point to the conclusion that, if the identity hypothesis is correct, then F must be consistent too. The argument requires as a premise not only that the protagonist is consistent, but also that she *knows* she is consistent. Since she knows she is consistent, she can know that, if 'I am F' is true, then F is consistent. She doesn't know whether F is *in fact* consistent (because she doesn't know whether 'I am F' is true), but as an intellectual exercise she can *imagine* that 'I am F' is true and explore the logical consequences of this supposition. In doing this she will prove various sentences of the form, *if I am F then* ϕ. If 'I am F' is *in fact* true then any such sentence that she can prove will also be proved by F. Penrose has us consider another formal system F', which is like F but which internalizes 'I am F' as an extra axiom. Thus, if F proves any sentence of the form *if I am F then* ϕ, F' will instead simply prove ϕ. Penrose observes that if F is consistent, and if 'I am F' is true, then F' must be consistent too. Thus the identity hypothesis implies, not only that F is consistent, but also that F' is consistent, and thus (via Gödel's theorems) that F''s Gödel sentence, $G(F')$, is true. The argument's protagonist can recognize this (for *we* can recognize this), so she can prove the sentence, 'if I am F, then $G(F')$'. If the identity hypothesis is in fact correct, then F will prove this sentence too. But if F proves this sentence, then F' will prove $G(F')$, which, by Gödel's theorem, is something it cannot do if it is consistent. In short, the

[7] There is some question as to precisely how Penrose's 'new argument' is supposed to go (see Chalmers 1995; Penrose 1996; Lindström 2001, 2006; Shapiro 2003). My Argument B is closely based on Penrose's (1994) original, informal presentation of the argument, and its essential logic is similar to Lindström's (2001) formulation. Departing from Penrose, I frame the argument in terms of the *consistency* of the formal systems in question, instead of the *soundness* of these systems, with the reason being that the former notion is less demanding and more general than the latter but still adequate for the argument's purposes.

identity hypothesis implies both that F' is consistent, and that F' will prove $G(F')$ – contradicting Gödel's theorem. Hence the identity hypothesis must be false. More formally:

J1. I am consistent, and I know it.

J2. If I am F, then F is consistent. (From J1.)

J3. If I am F and F is consistent, then F' is consistent (where F' is a formal system obtained by adding an extra axiom, 'I am F', to F, so that F' proves ϕ iff F proves 'If I am F then ϕ').

J4. If I am F, then F' is consistent. (From J2 to J3.)

J5. If F' is consistent, then $G(F')$. (From Gödel's first incompleteness theorem.)

J6. If I am F, then $G(F')$. (From J4 to J5.)

J7. I know that J1, J3, and J5 are true.

J8. If I know that J1, J3 and J5 are true, then I know that I can, by proving J6, prove a truth (since I can see that J6 follows from J1, J3 and J5).

J9. If I know that I can, by proving J6, prove a truth, then I will prove J6.

J10. I will prove J6. (From J7 to J9.)

J11. If I am F, and if I will prove J6, then F will prove J6.

J12. If F will prove J6 then F' will prove $G(F')$ (since J6 is of the form 'if I am F, then ϕ', with $G(F')$ replacing ϕ). (From what J3 says about F'.)

J13. If I am F, then F' will prove $G(F')$. (From J11 to J12.)

J14. If F' is consistent, then F' will not prove $G(F')$. (From Gödel's first incompleteness theorem.)

J15. If I am F, then F' will not prove $G(F')$. (From J4 to J14)

J16. I am not F. (From J13 to J15, by *reductio*.)

Three objections to Argument J are now considered.[8] The most popular objection targets the claim that the protagonist is consistent and knows it (i.e., premises J1). For instance, according to Turing (1947, 1948, 1950) the moral to be drawn from Gödel's work is that one can be intelligent enough to reason about the incompleteness theorems only if one is also so prone to error that no confidence can be put in the consistency of one's beliefs. In Turing's words, 'if a machine is expected to be infallible, it cannot also be intelligent' (1947). For Turing, fallibility – and a concomitant ability to make mistakes and then learn from them – is an *essential ingredient* of intelligence. Other authors (e.g., Grush and Churchland 1995) take the less radical position that, even if fallibility is perhaps not *necessary* for intelligence, it is nevertheless such an ineluctable feature of human performance that no human mathematician can know she is consistent.

Argument J can, I believe, be patched up to make it invulnerable to such objections by supposing that the argument's protagonist is what I will call a 'careful

[8] Most of these objections were initially conceived as objections to the original version of the mathematical argument, but apply equally against Argument J.

typist'. A careful typist is a person who evinces ordinary human fallibility in her day-to-day affairs (and who often makes mistakes and learns from them, as Turing says an intelligent being must), but who is charged with using a typewriter to produce a sequence of true sentences, and who takes the utmost care never to type a sentence unless she has a proof of its truth that meets the most exacting standards of simplicity, rigor and clarity. Whenever in doubt about the truth of a sentence, she errs on the side of caution and doesn't type it. Argument J is silent on what the protagonist must do to 'prove' a sentence. There is therefore nothing to prevent us stipulating that the protagonist 'proves' a sentence by typing it with the typewriter in question. She will therefore be 'consistent' iff the list of sentences she types is free of contradictions. (Mistakes she makes elsewhere in life will be irrelevant.) Provided the protagonist is such a 'careful typist', it is surely plausible that she might both be consistent and know she is consistent.

A similar objection (Chalmers 1995 and McCullough 1995) challenges the claim that the protagonist can know she is consistent (i.e., premise J1) based on Gödellian considerations.[9] Specifically, according to this objection the protagonist will, if the identity hypothesis is true, lapse into inconsistency in the very act of proving herself consistent (i.e., in the act of proving J1). However, as was explained in Sect. 10.4, Gödel's demonstration that a formal system will lapse into inconsistency if it proves itself consistent rests, in part, on the assumption that the system reasons classically, rather than reticently. The present objection is therefore dispensed with by supposing that the protagonist reasons reticently.

The last objection I consider (e.g., Robinson 1992, and Benacerraf 1968) targets premise J7 on the basis that, due to F''s complexity, its Gödel sentence, $G(F')$ would be such a stupendously large sentence of arithmetic that the protagonist would be unable to construct it, leaving her unable to know that J5 is true (and thus in no position to prove J6). This objection can be fended off by arranging for the protagonist to use a language in which a syntactically concise version of F''s Gödel sentence can be constructed. The following stipulations achieve this result:

- We use some name – say, '$\mathcal{F}$' – as a name for F'.
- We adopt some arbitrary method (say, some lexicographic method) for assigning Gödel numbers to sentences of English.
- We let $Sub(x,y)$ be the Gödel number of the sentence obtained by putting the number x in place of each occurrence of the lone free variable (if any) in the sentence with the Gödel number, y.
- We let $D(y)$ be the diagonalizing sentence, '$\mathcal{F}$ does not prove $Sub(y,y)$'
- We let d be $D(y)$'s Gödel number.
- Thus $D(d)$ says, '$\mathcal{F}$ does not prove $D(d)$'.[10]

[9] Chalmers uses Löb's theorem, rather than Gödel's theorem (but these two theorems are intimately related).

[10] This formulation of Gödel's diagonalization procedure is based on (Rucker 1982, p. 284).

Notice that *D(d)* is a self-referential sentence, that is true iff F' does not prove *D(d)*. Thus *D(d)* is – just like G(F') – a Gödel sentence of F'. This will come as no surprise since the above 'recipe' for constructing *D(d)* closely mirrors Gödel's own recipe for constructing *G(F′)*, with the only differences being that it uses English instead of Peano Arithmetic and uses the name, $\mathcal{F}$, instead of F''s (immensely large) Gödel number.[11] Importantly, whereas the task of constructing *G(F′)* is perhaps beyond the powers of a human, there seems nothing to prevent the protagonist from constructing *D(d)*. All she must do, when presented with a system, *F*, that allegedly models her own mathematical competency, is conceive of F' (a system obtained by adding the extra axiom, 'If I am *F*', to *F*), invent a name for it, and use this name in the above recipe. Having constructed *D(d)* in this way, she can use it as a 'stand in' for *G(F′)* within Argument J, as she sets about using this argument to refute the identity hypothesis.

At this point I hope the mathematical argument is beginning to look rather more compelling than it is generally given credit for. For reasons just outlined premises J1 and J7 seem robust. The remaining premises all appear unassailable, being in most cases either tautologies or provable theorems.

So, should we accept the mathematical argument's conclusion, and the implication that human theorem-proving powers exceed those of any formal system or digital computer? I think not. We should instead reject J8[12]:

J8. If I know that J1, J3 and J5 are true, then I know that I can, by proving J6, prove a truth (since I can see that J6 follows from J1, J3 and J5).

J8 appears innocuous at first blush: if one can see that the premises of a manifestly classically valid argument are true, then – so it would seem – one can prove the conclusion, safe in the knowledge that one is proving a truth. But the main theme of this chapter has been that such reasoning can be dangerous. We have seen that if an argument is perverse then accepting its premises involves committing oneself, on pain of having accepted a falsehood, to not accepting its conclusion. To prove the conclusion in such a case would be to falsify at least one of premises and commit the perversity fallacy. If the argument, J1, J3, J5 $\vdash$ J6 is perverse, then J8 is false.

And, indeed, J1, J3, J5 $\vdash$ J6 is perverse. To see why it is perverse, we must understand why the premises J1, J3 and J5 together entail, not only that J6 is true, but also that the protagonist will not prove J6. The explanation is as follows. In accepting J1, the protagonist accepts that *she knows she is consistent*. Were she to carelessly prove a sentence that might, as far as she knows, be contradictory, she

[11] When we ask whether the human protagonist in the mathematical argument can prove things a formal system cannot, we should not force her to use Gödel numbers and Peano arithmetic, which play to the strengths of formal systems, instead of names and natural language, which play to the strengths of the human mind. To do so would be to make her fight with one arm tied behind her back.

[12] The corresponding premise in Argument H is H5, which is problematic for the same reasons as J8.

would not know she was consistent, and so J1 would be false. Hence in accepting J1 she is committed to being careful not to undermine her own consistency. Now, for all that has been shown at this early point in Argument J, the identity hypothesis might be true: i.e., the protagonist's sentence proving dispositions might be the same as F's. So, as part of guarding against undermining her own consistency, the protagonist must be careful not to do anything that would undermine her consistency *if the identity hypothesis happened to be true*. With this thought in mind, let us imagine that the identity hypothesis is in fact true and that the protagonist proves J6. This being so, F will prove J6 too. If F proves J6, then F' proves $G(F')$. But if F' proves $G(F')$, then, by Gödel's theorem, F' is inconsistent. It would follow from this that J4 was false (since J4 says 'if I am F, then F' is consistent'). But if J4 is false then J1 must be false too, since J4 is derived from J1 by valid arguments having only one other, tautological premise (J3). And so the protagonist would in this case, by proving J6, have undermined her own grounds for accepting J1. The moral of this story is that the protagonist can know she is consistent, and J1 can be true, only if the protagonist won't take the risk of lapsing into inconsistency involved in proving J6. In short, *J1 entails that the protagonist will not prove J6*, from which it follows immediately that J1, J3, J5 $\vdash$ J6 is a perverse argument. (Its premises entail both J6, and that the protagonist won't prove J6.) Thus J8 is false and the mathematical argument is unsound.

If the above analysis is right then the mathematical argument is valuable, not because it shows that the human mind's problem-solving powers exceed those of a machine (it doesn't), but because it provides a wonderful, non-contrived example of a case where one must reason reticently, rather than classically, to avoid succumbing to a perversity fallacy.

10.6 Conclusion

In this chapter I have shown that CL exposes us to a kind of fallacy – the 'perversity fallacy' – wherein one accepts the conclusion of a classically valid argument even though its premises entail that one will not accept it, with the result that one falsifies the premises and undermines one's grounds for accepting the conclusion in the very act of accepting it. I have argued that we should instead reason in accordance with RL – a logic that includes a 'perversity check'. I have briefly sketched several versions of RL, and demonstrated that the notions of perversity and reticence have an important bearing on major issues in the philosophy of artificial intelligence.

Issues raised by this chapter that for reasons of space I must save for future work include: (i) applying RL to analyzing Moorean sentences and what Sorensen (1988) calls 'blindspots'; (ii) using RL to critique the doctrine that knowledge and/or justified belief is deductively closed; (iii) investigating 'higher-order perversities' (wherein the premises of an argument entail, not only that one won't accept the conclusion, but also that one won't detect the perversity); (iv) contrasting RL with other non-classical logics; and (v) further investigating the properties of basic and stepwise RL.

Acknowledgements Many thanks to Thomas Forster, Matthew Grice, Doukas Kapantais and Michael-John Turp for comments and suggestions.

References

Barr, M. (1990). Review: The emperor's new mind. By Roger Penrose. *The American Mathematical Monthly, 97*(10), 938–942.

Benacerraf, P. (1968). God, the devil and Gödel. *The Monist, 51*, 9–32.

Boolos, G. (1990). On seeing the truth of the Gödel sentence. *Behavioral and Brain Sciences, 13*(4), 655–656.

Bowie, G. L. (1982). Lucas' number is finally up. *Journal of Philosophical Logic, 41*(3), 279–285.

Chalmers, D. (1995). Minds, machines, and mathematics. A review of shadows of the mind by Roger Penrose. *Psyche, 2*, 11–20.

Gaifman, H. (2000). What Gödel's incompleteness result does and does not show. *The Journal of Philosophy, 97*(8), 462–470.

Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monash Mathematical Physics, 38*, 173–198.

Gödel, K. (1951). Some basic theorems on the foundation of mathematics and their implications. In S. Feferman (Ed.) (1995), *Kurt Gödel Collected works, vol. III: Unpublished essays and lectures.* (pp. 304–323). Oxford: Oxford University Press.

Grush, R., & Churchland, P. (1995). Gaps in Penrose's toilings. *Journal of Consciousness Studies, 2*(1), 10–29.

Lindström, P. (2001). Penrose's new argument. *Journal of Philosophical Logic, 30*, 241–250.

Lindström, P. (2006). Remarks on Penrose's new argument. *Journal of Philosophical Logic, 35*, 231–237.

Lucas, J. R. (1961). Minds, machines and Gödel. *Philosophy, 36*, 112–127.

Lucas, J. R. (1996). Minds, machine and Gödel: A retrospect. In P. J. R. Millican & A. Clark (Eds.), *Machines and thought: The legacy of Alan Turing* (pp. 103–124). Oxford: Oxford University Press.

McCullough, D. (1995). Can humans escape Gödel? A review of shadows of the mind by Roger Penrose. *Psyche, 2*(23), 57–65.

Nagel, E., & Newman, J. (1957). *Gödel's proof*. New York: New York University Press.

Penrose, R. (1989). *The emperor's new mind*. Oxford: Oxford University Press.

Penrose, R. (1994). *Shadows of the mind*. Oxford: Oxford University Press.

Penrose, R. (1996). Beyond the doubting of a shadow. *Psyche, 2*(23), 89–129.

Putnam, H. (1960). Minds and machines. In SidneyHook (Ed.), *Dimensions of mind: A symposium*. New York: New York University Press. Reprinted in A. R. Anderson (1964) (Ed.), *Minds and machines*. Prentice-Hall, 77.

Raatikainen, P. (2014). Gödel's incompleteness theorems. In Edward N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Spring 2014 Edition). http://plato.stanford.edu/archives/spr2014/entries/goedel-incompleteness/

Robinson, W. (1992). Penrose and mathematical ability. *Analysis, 52*(2), 80–87.

Rucker, R. (1982). *Infinity and the mind: The science and philosophy of the infinite*. Princeton: Princeton University Press.

Shapiro, S. (2003). Mechanism, truth and Penrose's new argument. *Journal of Philosophical Logic, 32*, 19–42.

Sorensen, R. (1988). *Blindspots*. Oxford: Clarendon.

Turing, A. M. (1947). Lecture to the London Mathematical Society on 20 February 1947. Reprinted in D. C. Ince (1992) (Ed.), *Collected works of A.M. Turing: Mechanical intelligence.* Amsterdam: North Holland.

Turing, A. M. (1948). *Intelligent machinery*. Reprinted in D. C. Ince (1992) (Ed.), *Collected works of A.M. Turing: Mechanical intelligence.* Amsterdam: North Holland.

Turing, A. M. (1950). Computing machinery and intelligence. Reprinted in D. C. Ince (1992) (Ed.), *Collected works of A.M. Turing: Mechanical intelligence.* Amsterdam: North Holland.

Chapter 11
Cognition as Higher-Order Regulation

Stefano Franchi

Abstract The chapter discusses Antonio Damasio's understanding of higher-level neurological and psychological functions in *Self Comes to Mind* (2010) and argues that the distinction he posits between regulatory (homeostatic) physiological structures and non-regulatory higher-level structures such as drives, motivations (and, ultimately, consciousness) presents philosophical and technical problems. The paper suggests that a purely regulatory understanding of drives and motivations (and, consequently, of cognition as well) as higher-order regulations could provide a unified theoretical framework capable of overcoming the old split between cognition and homeostasis that keeps resurfacing, under different guises, in the technical as well as in the non-technical understandings of consciousness and associated concepts.

11.1 Thinking and Breathing

The classic definition of "cognition" may be traced back to the Latin noun *cognitio*, a derivative of the verb *cognosco* ("to know," "to become acquainted with," cf. Greek *gignòsko*). A *cognitio* is the final result of the act that allows the subject to become acquainted with its object thereby producing knowledge about it. Coherently with its nominal grammatical status, a *cognitio* is thus an object of a specific kind: an item of knowledge, as the classic sources attest.[1] While the modern meaning of *cognition* has kept the original reference to knowledge, it has shifted it to the process

[1] In the *De natura deorum*, for instance, Cicero says of the gods that "they raised men from the ground and made them stand erect so that by looking at the heavens they could get knowledge of the gods (Qui primum eos humo excitatos celsos et erectos constituit, ut *deorum cognitionem* caelum intuentes capere possent)" (*d.N.D*, II, 140, 1933, 56). A similar use of *cognitio* is in the *De officiis* (I,43), where Cicero speaks of the "knowledge and contemplation of nature (*cognitio contemplatioque naturae*)" (1938, 153). The term is very frequent in his works and can be found also in the *Tusculan Disputations*, the *De finibus*, etc. The nominal use of the term is standard

S. Franchi (✉)
Texas A & M University, College Station, TX, USA
e-mail: stefano.franchi@gmail.com

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_11

that produces the knowledge of the object confronting the subject. Ulrich Neisser's (1967) suggestion, perhaps the first formal definition to gain wide acceptance, is still repeated more or less verbatim in all textbooks on cognitive psychology:

> The term "cognition" refers to all the processes by which the sensory input is transformed, reduced, elaborated, stored, recovered, and used. It is concerned with these processes even when they operate in the absence of relevant stimulation, as in images and hallucinations. Such terms as sensation, perception, imagery, retention, recall, problem-solving, and thinking, among many others, refer to hypothetical stages or aspects of cognition. (1967, 4)

The substantial difference Neisser introduced—implicitly, in 1967—concerns the subjects of cognition. Classic, medieval, and modern philosophers took it for granted that only humans could have *cognitiones*—indeed, they often based their definitions of human nature on the subject's capacity to produce demonstrative, i.e. linguistically articulated, knowledge about the objects confronting them.[2] Neisser's definition, on the other hand, is broad enough to include all kinds of sensory processing resulting in behavioral output. It must therefore be applied to non-human organisms as well, insofar as they are able to receive sensory input and act consequently on it. Neisser's revised (1976) definition makes this referential extension explicit. "Cognition—he states—is the *activity of knowing*: the acquisition, organization, and the use of knowledge. It is something that *organisms* do and *in particular* something that people do."[3]

The double shift in the meaning and scope (from state to process and from humans to organisms) turned cognition into an all-encompassing set of processes that leads to two strictly related potential problems for its theoretical understanding. First of all, the shift from *cognitiones* as knowledge items to "cognition" as process tends to obfuscate the role of the former in the overall biological economy of the (human) organism. According to the classical view (as presented, for instance, by Aristotle), the subject uses the knowledge it gains about the objects it confronts in order to achieve its natural goal (or its function: its "ergon"[4]). Since the natural goal of a human being is a life conducted according to discursive reason (*logos*), it follows that the knowledge the subject acquires (its *cognitiones*, as his Latin-

in the technical Latin of Medieval and Modern philosophy, as attested, for instance, by Aquinas, Spinoza, and Descartes.

[2] This is the basis of Aristotle's definition, for instance, and it is accepted wholeheartedly by the Medieval tradition that refers to him, with the slight modifications necessary to make it fit the Christian view of humans created in God's image. Early Modern Western philosophy follows in the same vein.

[3] 1976, 1, my emph. Indeed, contemporary studies in animal cognition adapt Neisser's definition while inserting an explicit reference to animals in general. See, for instance, Sarah Shettleworth: "Cognition refers to the mechanisms by which animals acquire, process, store, and act on information from the environment. These include perception, learning, memory, and decision-making. The study of comparative cognition is therefore concerned with how animals process information, starting with how information is acquired by the senses" (2009, 4).

[4] As Aristotle argues in *Eth. Nic.* I,7, 1098b1-18.

speaking followers will say) are just one element among the many that the subject needs to achieve the goal. A healthy body is also necessary, as well as a healthy relationship to the subject's own fellow beings in a well-ordered social formation. Most importantly, all the knowledge items the subject gains are to be part of this overall search for its natural goal (or, to use the Greek term, for a well-balanced life of *eudaimonia*), with the consequence that the highest and most prized form of knowledge concerns those eternal objects whose timelessly unchanging nature can become an endless source of happy contemplation.[5]

The Platonic attitude (at least in the middle dialogues) is similar, if more pessimistic: the difficult acquisition of stable, certifiable, and demonstrative knowledge which only a few human beings will attain is a component of a larger existential strategy, as it were, aimed at achieving a suitably satisfactory and therefore properly human life.

Other Western philosophers will change—sometimes radically—their definitions of the ultimate human goal, and will also change accordingly the role that knowledge acquisition will have to play on the road to its satisfaction. What does not change, however, is the realization that the *cognitiones* the subject acquires are just one element in a much broader and more complex picture. To put it in very simple terms: the study of knowledge acquisition cannot be limited to inquiries on *how* the subject gains its knowledge. It must also take into consideration *why* it does so.

Interestingly enough, Neisser was well aware of this particular limitation of the cognitive approach he was promoting. In his 1967 book, the definition of cognition I quoted above is immediately followed by this interesting paragraph:

> Given such a sweeping definition, it is apparent that cognition is involved in everything a human being might possibly do; that every psychological phenomenon is a cognitive phenomenon. But although cognitive psychology is concerned with *all human activity* rather than some fraction of it, the concern is from *a particular point of view*. *Other viewpoints are equally legitimate and necessary*. *Dynamic psychology*, which begins with motives rather than with sensory input, is a case in point. Instead of asking how a man's actions and experiences result from what he saw, remembered, or believed, the dynamic psychologist asks how *they follow from the subject's goals, needs, or instincts*. Both questions can be asked about any activity, whether it be normal or abnormal, spontaneous or induced, overt or covert, waking or dreaming. Asked why I did a certain thing, I may answer in dynamic terms, "*Because I wanted . . .* ," or, from the cognitive point of view, "*Because it seemed to me . . .* " (1967, 4, my emph.)

When trying to provide a satisfactory explanation for human behavior, Neisser argues, the cognitive approach provides only part of the answer. At best, it can explain *how* the complex relationships between the sensory apparatus, the internal cognitive machinery, and the organism's effector systems (the skeletal-muscular apparatus) produced the observable output. But it cannot explain *why* the organism engaged

[5] See *Eth Nic.* X, 6–9, as well as the similar Platonic theory of intellectual pleasure that Socrates develops in *Rep.* IX, 585b and ff.

into that particular set of interactions. This is where a "dynamic" inquiry[6] must "legitimately and necessarily" enter into the picture and supplement the cognitive answer with a higher level explanation that would reinsert the cognitive processes within the broader framework of an organism's conscious and possibly unconscious needs and desires. In 1967, "dynamic psychology" or "psychodynamics" were the accepted labels for all psychological theories that could trace their heritage back to Freud and his immediate followers (Jung, Klein, Adler, and so on). The terms had been coined by analogy with the traditional approaches that govern the explanation of movements in physics: statics and kinematics inquire into the laws that governs the motion between moving bodies without taking into consideration the forces that caused it, dynamics focuses explicitly on the forces attracting or repelling physical bodies and determine their observable behavior. In other words, kinematics is the study of motion regardless of its cause, while dynamics is precisely the study of the causes of motion. A physically satisfactory explanation of motion requires both: kinematics provides the surface level description of how bodies interact given their mass, velocities, and acceleration, while dynamics provides an explanation of how movement can be initiated and modified through the application of external forces to the bodies themselves. By analogy, the psychodynamic approach focuses on the (mostly unconscious) forces that motivate (i.e. cause) the psychologically observable behavior. Neisser is therefore arguing that cognitive psychology provides a sort of kinematics of human psychology which must necessarily be supplemented by a psychoanalytic dynamics to provide a full theory.

With the demise of psychoanalysis in the late 1970s, and the more general lack of scientific respectability for the Freudian approach that followed, the specific integration to cognitive psychology Neisser was advocating lost much of its appeal. However, the debates that have animated the study of cognition in subsequent years do not seem to have addressed Neisser's broader point, for they have been largely focused on the nature of this connection between environmental sensory input and the resulting organism's actions. Until not long ago, functionalism (and/or computationalism Piccinini 2004) used to provide the standard answer: sensory input is somehow translated into internal representations, representations are processed by computational syntax-based processes, and the eventual results are converted into actions by an organism's effectors.

[6] This use of "dynamic" as a short-hand for a causal explanation of behavior will probably sound rather strange to 21st scientists. As Gordana Dodig Crnkovic noted (personal communication), a physicist, when seeing the word "dynamic" in a scientific context such as Neisser's, will most likely think of dynamic attractors that lie below cognitive processes and which stand for particular *mechanisms* and not for *motives*. In other words, when a physicist points us toward a "dynamic explanation," she is telling us that we should be looking for a particular *class of mechanisms*—namely, those governed by dynamic attractors—and not be on the lookout for a further causal level. As I explain below, Neisser's now outdated usage of the term presupposes psychoanalysis's own self-description and the contemporary (ultimately, nineteenth century) sub-divisions of classical mechanics.

More recent developments have thrown some considerable doubts about the viability of the classic functionalist/computational paradigm. Connectionism, situated cognition, dynamic approaches, etc. have proposed alternative frameworks that purport to obviate some or all of functionalism's shortcomings. In all these instances, however, the alternative explanations have the same scope as Neisser's original proposal: they provide a different theory for the connection between sensory input and outputs—what I labeled the *how* of cognition—but refrain from addressing the broader framework behind it—its *why.*

The possibly structural incompleteness of current as well as past theories of cognition is made more complex by the second semantic shift I mentioned above. Once cognition is redefined to encompass *any* possible relationship between the environmentally-initiated sensory input and organism-produced output, it is evident, as Neisser affirms in 1967, that "all human activity" becomes a cognitive phenomenon. In 1976, the claim is extended to all organisms, although it is still considered to be more characteristic of people than, say, of bacteria, plants, or non-human animals. This semantic extension leads to the following problem: if a dynamic dimension were to be reinstated in the study of cognition, would it be identical, or at least structurally similar to the dynamic component of non-cognitive biological functions, or would it be essentially different? Otherwise put: the vast majority of an organism's exchanges with the environment are concerned with keeping itself viable in the face of continuously changing environmental circumstances and internal needs. Breathing, feeding, moving and finding shelter, are all instances of biological processes that involve an organism receiving inputs from the environment and producing adequate output in response to them. We do know that these biological processes are directed by a search for equilibrium with the surrounding environment. Physiological processes—in spite of the huge differences between them—are driven by homeostatic regulation. They all search to stabilize (i.e. "regulate") the internal and usually critical value of some parameter (sugar content in the blood, CO_2 concentration in the lungs, and so on) with respect to continuously varying environmental conditions and possibly changing internal capacities.[7]

The search for homeostatic equilibrium (which can often be very complex) provides the dynamic dimension—the *why* of biological processes—while the specific mechanisms the organism deploys to achieve that equilibrium within its particular homeostatic systems furnish its *how.* Consider breathing, the apparatus that was the subject of one the early and most rigorous studies of physiological homeostasis. J. S. Haldane (1917, 1922) showed that the respiratory system is constantly monitoring how the internal CO_2 level is modified by environmental variation (such as a the change in air composition that may happen in a mine shaft) as well as by body-generated variation (such as increases in muscular activities, or decreased efficiency of the apparatus itself). In all cases, the body responds to these

[7] See Franchi (2011b) for a detailed discussion of homeostasis, whose details and fascinating history I am forced to skip over in the present context.

variation by deploying suitable actions—e.g. an increase in ventilation rates—until proper CO_2 values are restored. The search for equilibrium directs the mechanism of breathing. In other words: breathing is essentially a form of regulation of the relationship between organism and environment from the specific point of view of CO_2 levels in the blood.

Now consider cognition again from the broader point of view of its role in the relationship between organism and environment. If cognition is involved in all of the organism's activities, it is natural to ask if the overall mechanism that directs it is similar to the mechanism that presides over the other biological functions. To put it bluntly, the question is: is thinking fundamentally akin to breathing? Is cognition directed, as most of the organism's physiological processes are, by homeostatic regulation? I propose to explore this hypothesis by looking at an aspect of the theory of mind recently proposed by Antonio Damasio.

11.2 On the Insufficiency of Regulation for Life

In his 2010 book, *Self Comes to Mind,* Damasio builds a sustained argument in favor of the fundamental role played by homeostatic regulation in all aspects of biological processes. He even argues that the deeper role of regulation "in neurobiology and psychology has not been appreciated" (2010, 45). The statement suggests that Damasio may intend to expand the scope of homeostasis beyond the physiological level. Since the late nineteenth century, when biologists such as Claude Bernard, John S. Haldane, and later Walter Cannon turned their attention to the phenomenon, homeostasis has traditionally be confined to the level of purely physiological processes such as the regulation of sugar or oxygen levels in the blood, while psychological processes were traditionally excluded from its purview. J. S. Haldane, for instance, declared that it is "unmeaning to treat consciousness as a mere accomplishment to life, or to ignore the difference between blind organic activity and rational behavior" (1917, p. 115), while Walter Cannon took Haldane's position further and argued that the successful regulation of physiology by homeostasis freed the organism "for the activity of the higher levels of the nervous system and the muscles that govern it .... [With homeostasis, we] find the organism liberated for its more complicated and socially important tasks because it lives in a fluid matrix, which is kept in a constant condition" (1939[1932], p. 302). Damasio's explicit mention of its role in "neurobiology and psychology" seems to run counter to this well-established tradition and open the door to an upward extension of regulatory processes beyond the lower physiological levels.

Yet, Damasio immediately tempers the scope of his previous statement by arguing that homeostatic regulation is not sufficient for these "more-than-physiological" functions. Indeed, regulation is not even sufficient for basic physiology, he claims: while homeostasis can correct imbalances between the internal state of the organism and the environment it lives in, it presupposes a constant and fixed environment. Put it differently, regulation is short-sighted and naive: it only sees its immediate surroundings and it assumes they are more or less static. He conceives homeostasis as a set of static rules of the form "if *x* is needed, then do *y*". But what happens when the organism cannot carry out action *y*, because the environment no longer makes it possible? Or when the need for vital yet unattainable element *x* could be preempted by searching for a different element *z*?

Regulation is not enough to guarantee survival, Damasio claims, "because attempting to correct homeostatic imbalances after they begin is inefficient and risky." Physiological regulation needs a supplement—"drives and motivations need to help homeostasis" (2010, 59)—that will rely on regulation while providing a more far-sighted and less naive management of its basic functions through the goals and incentives regulation cannot generate. Damasio's conclusion is that higher brain functions—indeed, the brain itself—is just such a manager: "a brain—he states—exists for managing life inside a body" (2010, 64). This thesis, which Damasio formulates at the end of the first part of the book, provides the blueprint for the overall theory of consciousness he goes on to detail in the remaining chapters. We could rephrase it by saying that all brain functions do is to manage and direct physiological regulation in the three key areas of sensing, response policy, and movement.

The argument presupposes that drives and motivations cannot have a purely regulatory character. As a consequence, Damasio holds that a satisfactory explanation of cognitive functions needs to drive a fundamental distinction between *regulatory* mechanisms *vs.* the higher order *non-regulatory* motivational and incentive structures that "manage" homeostasis. I disagree. Damasio's distinction, as I will argue below, rests on a traditional yet fundamentally limited view of homeostasis whose only justification rests on unwarranted philosophical assumptions.

11.3 Meta-regulation

As I mentioned previously, Damasio's setup repeats and updates the classic conception of homeostasis we find in Claude Bernard, John S. Haldane, and Walter Cannon (see also Franchi 2011b) and it reintroduces—in spite of Damasio's claim for a model of cognition and consciousness solidly grounded in biology[8]—an unwelcome split between higher motivational non-homeostatic structures and lower level homeostatic ones that just repeats the old Western philosophical qualitative split between higher-level non-homeostatic (rational) processes and lower-level physiological and instinctual ones. In fact, Damasio's overall framework bears an uncanny resemblance to the tripartite distinction between plants', animals', and

[8] See, for instance, the sweeping declaration on the need to ground "pain, pleasure, emotions, and feelings; social behaviors; religions; economies and their markets and financial institutions; moral behaviors; laws and justice; politics; art, technology, and science" in "life regulation" (2010, 63–64).

humans' modes of life that Aristotle drew in the first book of *Nichomachean Ethics* and that was substantially repeated, with very few exceptions, throughout the history of Western philosophy.[9] These philosophical difficulties translate into technically problematic consequences at the biological level. For instance, Damasio's emphasizes that sensing and movement are fundamental biological innovations which, at the same time, put homeostatic regulation at risks (since they allow organisms to wander outside their homeostatic niches) and prompt the development of non-homeostatic higher level structures eventually culminating in organisms endowed with brains and minds. The counter-intuitive consequence is that all plants, being structurally unable to move, are indeed some kind of evolutionary dead end, while motile bacteria such as *E. choli* have the potential for consciousness. Damasio claims that "the tragedy of plants, though they do not know it, is that their corseted cells could never change their shape enough to become neurons" (2010, 53). Since plants do not move, however, they wouldn't need neurons in the first place. Damasio is here repeating, more or less unwittingly, the modern version of Aristotle's standard classification of life forms provided by Hans Jonas (1966, 1984), who claimed that bacteria have the essence of human (Kantian) freedom in a nutshell. The biological consequences are an effect of the philosophical assumption: Damasio starts from an implicit yet very powerful (and very old) anthropocentric view of world as a whole and projects it back onto the biological domain as a dichotomy between regulation-only organisms on the one hand (plants first and foremost) and "regulation+" on the other.

One possible, and epistemologically natural alternative to Damasio's framework that I would like to suggest consists in conceiving of higher order functions such as "drives and motivations" as higher order regulatory structures. Basic homeostatic processes regulate the exchange between organism and environment on the basis of simple parameters (the *if/then* rules Damasio refers to) that can only cope with specific and static conditions. Higher order processes (e.g. drives, etc.) regulates the parameters themselves: they try to reestablish equilibrium between the organism and the environment (as all homeostatic processes do) by changing the rules used by lower level processes. Simply put, basic biological processes homeostatically regulate agent/environment exchanges *directly*. Higher level processes—all the way up to cognition and consciousness—regulate those exchanges indirectly: higher level processes regulate lower level processes that regulate biological processes directly. While lower level processes strive to keep the value of a biologically crucial variable within acceptable bounds (e.g. "optimal sugar *concentration*," "optimal CO_2 *level*," etc.), higher level processes keep the functioning of a lower level process within acceptable bounds (e.g. "satisfactory sugar concentration *regulation*," and so on). This view has the advantage of positing a single biological mechanism as the root of all biological functions—namely, regulation—while allowing to differentiate across different classes of organism on the basis of the sophistication and logical depth of their regulatory structures.

[9] See Aristotle 1984, v. 2, 1734–1735=1097b–1098a and Franchi (2014).

The idea is not new. In different forms, it has been advanced by thinkers as diverse as W. R. Ashby, with his concept of homeostatic ultrastability (1960); Freud, with his position of the unconscious drive as the only motivational structure and, at the same time, the only interface between the body and the higher level control mechanism[10]; and Clark Hull, with his conception of motivation as drive-reduction (1943). Jean Piaget's theory of cognitive development is perhaps the theoretical model that comes closer to the idea of meta-regulation. In *Biologie et connaissance* (1967), Piaget states explicitly that cognitive processes are "the results of organic auto-regulation (whose essential mechanisms they reflect) and, at the same time, the most differentiated organs of regulations, within the organism's interaction with the environment" (37). In a passage that seems to be constructed as an ideal reply to Damasio's criticism of homeostasis's shortcomings, Piaget goes as far as suggesting the existence of a hierarchical series of regulation and meta-regulation mechanisms. Reflecting on the possible problems of an organism's basic regulatory strategies (as deployed by the endocrine and nervous systems), Piaget suggests that, in some cases,

> [In addition] to all kinds of *feebacks,* there could be overlapping *feed-forwards* mechanisms that take care of the insufficient speed or of the excessive amplitude of the *feedbacks*, thereby constituting a kind of second-order regulation or a regulation of regulation itself. (1967, 243, Piaget's emph.)

Even though Piaget's theory of regulation presents important differences with Freud's (and even more so with Hull's), his insistence on the crucial role played by homeostatic regulation and the organism's constant search for equilibrium is structurally consistent with the psychodynamic approach whose integration with cognitive psychology Neisser was advocating in the same year as Piaget was writing the lines quoted above. These past attempts have so far failed to establish the viability of higher-order regulation as an explanatory paradigm in biology as well as in the cognitive sciences. Nonetheless, I think that a purely regulatory understanding of drives and motivations (and, consequently, of cognition as well) could provide a unified theoretical framework capable of overcoming the old anthropomorphic split between cognition and homeostasis that keeps resurfacing, under different guises, in the technical as well as in the non-technical understandings of cognition, consciousness, and associated concepts. It would make possible a conception of *all* the aspects of human and non human life as purely regulatory in character—that is, as (possibly complex) sequences of purely homeostatic processes. As a consequence, all motivational and incentive structures would become homeostatic (as Freud 1953b claimed, with his conception of the drive as the only motivational structure and, at the same time, the only interface between the body and higher level control mechanisms). What would it take to assess the hypothesis that homeostasis, all by itself, can provide a sufficient motivational structure for cognition? As I hinted above, the strategy to be pursued is twofold. On the one hand, we need

[10] (Freud 1953b,c). See Franchi (2011a) for a brief discussion of Freud's and Lacan's theory of the drive from an Ashby-inspired regulation perspective.

to overcome a theoretical-historical problem: the limited and subordinated role traditionally assigned to homeostasis in cognitive science is the scientific effect of a well-entrenched Western philosophical framework that privileges a subject's conscious (and rational) actions over bodily regulation. I barely hinted at the depth of this problem above, although I have dealt with it more extensively elsewhere (2011b, 2013, 2014).

The second aspect is more technical. The best model ever developed toward a purely regulatory approach to cognition is the framework Ashby developed in his *Design for a Brain* (1960). However, the theory was never really tested in empirical settings, let alone developed beyond the first preliminary steps Ashby himself undertook with his device (the homeostat) or in the few batch-oriented simulations he conducted in the following decade.[11] The technology available in the 1950s and 1960s did not really make it easy to produce an empirical model of the fully connected continuous time networks Ashby's envisioned, and the rough electro-mechanical device he built did not scale well beyond a few units. Even though our technology has progressed far beyond the stage it was in the 1960s, both in terms of the hardware and software routinely available for digital simulations of neural networks and in terms of our theoretical understanding of artificial neural networks, Ashby's insights have not been picked up in their full generality yet. Starting with Ezequiel Di Paolo's (2000) groundbreaking work, several researchers (Williams 2006; Williams and Noble 2007; Herrmann et al. 2004; Der 2001; Ikegami and Suzuki 2008; Iizuka and Di Paolo 2007, and several others afterwards) have recovered some of the technical solutions Ashby suggested and tested them within the context of evolutionary robotics (i.e. the theory of ultrastability). But we still do not have a complete implementation of Ashby's generalized homeostatic model. All the mentioned attempts graft Ashby's technical solution (ultrastability) onto a theoretical framework (standard *CTRNN*s) and a philosophical view of cognition (*enactivism*) which are arguably incompatible with the technical and philosophical approaches pursued in *Design for a Brain*.[12] Moreover, the development of a truly Ashbian homeostatic network is only the first step toward a fully regulatory and empirically testable model of cognition. Ashby's theoretical results show that a *stable* and, most importantly, *adaptive* homeostatic model demands an overall architecture comprising sparsely coupled sub-networks.[13] Ashby tried to develop a practical device implementing the theory of general homeostasis (which he called DAMS or "Dispersive and Multistable System"), but his attempts failed, mostly due to the technical complexities produced by the analog technology he was working

[11] See the papers collected in Ashby (1981).

[12] See Franchi (2011b) for and extended discussion of the incompatibility between the contemporary and Ashby's approach to homeostasis and Franchi (2013) for a few, preliminary result on robotics simulations of a fully homeostatic Ashbian model of behavior.

[13] The theory is sketched out in chapters 15–17 of *Design for a Brain*. Empirical results about the necessary loose coupling of homeostatic sub-networks confirming the earlier theoretical analysis are provided in Ashby (1981).

with in the late 1950s.[14] A viable research program willing to follow Ashby's pioneering efforts would have to start again from his *technical* insights into a general homeostatic cognitive architecture while, at the same time, pursuing a general *theoretical* articulation of (human and non-human) life as an essentially passive, regulation-based phenomenon. While this is obviously not the place for such an ambitious project, I hope the general and historically-based considerations I advanced may provide a first opening salvo toward such a goal.

References

Aristotle. (1984). *The complete works of Aristotle*. Princeton: Princeton University Press.

Ashby, W. R. (1960). *Design for a brain* (2nd ed.). London: Chapman and Hall.

Ashby, W. R. (1981). *Mechanisms of intelligence: Ross Ashby's writings on cybernetics*. Seaside: Intersystems Publications.

Cannon, W. (1939[1932]). *The wisdom of the body* (2nd ed.). New York: W.W. Norton.

Cicero, M. T. (1933). *De natura deorum* (Loeb classical library). Cambridge: Harvard University Press.

Cicero, M. T. (1938). *De officiis* (Loeb classical library). Cambridge: Harvard University Press.

Damasio, A. (2010). *Self comes to mind: Constructing the conscious brain*. New York: Knopf Doubleday.

Der, R. (2001). Self-organized acquisition of situated behaviors. *Theory in Biosciences, 120*(3), 179–187.

Di Paolo, E. (2000). Homeostatic adaptation to inversion of the visual field and other sensorimotor disruptions. In J. A. Meyer, A. Berthoz, D. Floreano, H. L. Roitblat, & S. W. Wilson (Eds.), *From Animals to Animats 6: Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior* (pp. 440–449). Cambridge: MIT Press.

Franchi, S. (2011a). Jammed machines and contingently fit animals: Psychoanalysis's biological paradox. *French Literature Studies, 38*, 231–256.

Franchi, S. (2011b). Life, death, and resurrection of the homeostat. In S. Franchi & F. Bianchini (Eds.), *The search for a theory of cognition: Early mechanisms and new ideas* (pp. 3–52). Amsterdam: Rodopi.

Franchi, S. (2013). Homeostats for the 21st century? Lessons learned from simulating Ashby simulating the brain. *Constructivist Foundations, 8*(3), 501–532. With open peer commentaries and author's response.

Franchi, S. (2014). General homeostasis, autonomy, and heteroomy: Toward an affect-based theory of cognition. In V. Müller (Ed.), *Fundamental Issues of Artificial Intelligence*. Berlin: Springer.

Freud, S. (1953a). *Standard edition of the complete psychological works of Sigmund Freud*. London: Hogarth Press.

Freud, S. (1953b). *The standard edition of the psychological works of Sigmund Freud* (chap Instincts and their vicissitudes, pp. 117–140). Vol 14 of Freud (1953a), written in 1915.

Freud, S. (1953c). *The standard edition of the psychological works of Sigmund Freud* (chap. Project for a scientific psychology, pp 283–387). Vol 1 of Freud (1953a), written in 1895.

Haldane, J. S. (1917). *Organism and environment as illustrated by the physiology of breathing*. New Haven: Yale University Press.

Haldane, J. S. (1922). *Respiration*. New Haven: Yale University Press.

[14] See Pickering (2010) and Husbands and Holland (2008) for a discussion of DAMS.

Herrmann, J. M., Holicki, M., & Der, R. (2004). On Ashby's homeostat: A formal model of adaptive regulation. In S. Schaal (Ed.), *From Animals to Animats 8. Proceedings of the Eighth International Conference on Simulation of Adaptive Behavior* (pp. 324–333). Cambridge: MIT Press.

Hull, C. L. (1943). *Principles of behavior. An introduction to behavior theory*. New York: D. Appleton-Century.

Husbands, P., & Holland, O. (2008). The Ratio Club: A hub of British cybernetics. In P. Husbands, O. Holland, & M. Wheeler (Eds.), *The mechanical mind in history* (pp. 91–148). Cambridge: MIT Press.

Iizuka, H., & Di Paolo, E. (2007). Toward spinozist robotics: Exploring the minimal dynamics of behavioral preference. *Adaptive Behavior, 15*(4), 359–376.

Ikegami, T., & Suzuki, K. (2008). From a homeostatic to a homeodynamic self. *Biosystems, 91*(2), 388–400.

Jonas, H. (1966). *The phenomenon of life: Towards a philosophical biology*. New York: Harper and Row.

Jonas, H. (1984). *The imperative of responsibility. Foundations of an ethics for the technological age*. Chicago: The University of Chicago Press.

Neisser, U. (1967). *Cognitive psychology*. Englewood Cliffs: Prentice-Hall.

Neisser, U. (1976). *Cognition and reality. Principles and implications of cognitive psychology*. New York: Freeman.

Piaget, J. (1967). *Biologie et connaissance*. Paris: Gallimard.

Piccinini, G. (2004). Functionalism, computationalism, and mental states. *Studies in the History and Philosophy of Science, 35*, 811–833.

Pickering, A. (2010). *The cybernetic brain: Sketches of another future*. Chicago: The University of Chicago Press.

Shettleworth, S. J. (2009). *Cognition, evolution, and behavior*. New York: Oxford University Press.

Williams, H. (2006). *Homeostatic adaptive networks*. PhD thesis, University of Leeds

Williams, H., & Noble, J. (2007). Homeostatic plasticity improves signal propagation in continuous-time recurrent neural networks. *Biosystems, 87*(2–3), 252–259.

Chapter 12
Eliminativisms, Languages of Thought, & the Philosophy of Computational Cognitive Modeling

Marcello Guarini

Abstract The philosophy of cognitive modeling, especially computational cognitive modeling, contains a range of possibilities for understanding language and cognition. Jerry Fodor is perhaps best known for defending the language of thought hypothesis. In his hands, that hypothesis is a way of understanding how folk psychology could be true. Its truth does not entail the language of thought, but the language of thought (in his sense) entails the truth of folk psychology. This paper explores the relationship between eliminativisms, languages of thought, and folk psychology. Among other things, a way of allowing language a role in cognition that is weaker than Fodor's will be considered. The possibility that a more moderate language of thought could be compatible with eliminativism is considered as well. The paper provides a sense of the conceptual territory; it does not contain detailed evaluations of the various positions discussed. The tone is as nonpartisan as possible.

12.1 Introduction

Some might find it tempting to think that if we discover in the head computational processes defined over words or sentences, then (a) Jerry Fodor is right about the language of thought, (b) folk psychology is true, and (c) eliminativism is false. I will argue that these inferences are hasty by showing that it is possible to imagine a role for language in thought that is less substantive then what Fodor imagines (so not any kind of language in thought would vindicate his approach). Moreover, it will be shown that a more moderate role for language in thought would be compatible with folk psychology being false. The point will not be to argue that a more moderate approach to language in thought is correct; it will be to show that we can coherently imagine it. It is an empirical question what is going on in the head, though greater clarity on what the options are influences what we go looking for, so it is worth clarifying some of the available options.

M. Guarini (✉)
Department of Philosophy, University of Windsor, Windsor, ON N9B 3P4, Canada
e-mail: mguarini@uwindsor.ca; http://www.uwindsor.ca/guarini

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_12

The strategy for doing the above will be to start by disentangling some of the different positions that might be called eliminativist, which is the purpose of the next two sections. In Sect. 12.3, it will be argued that while intentional eliminativism entails folk psychological eliminativism, it does not follow that folk psychological eliminativism entails intentional eliminativism. In Sect. 12.4, strong and weak conceptions of the language of thought will be put forward. Fodor's work (1975, 1987, 1991, 2008) will be taken as the exemplar of strong language of thought. The weak conception is something that no one denies – that we sometimes consciously "talk" to ourselves in a subverbal but conscious manner. This distinction between strong and weak approaches prepares the ground for the eventual introduction of a moderate language of thought. In Sect. 12.5 it will be argued that the absence of words or sentences in the head is not a convincing reason to abandon folk psychology. Section 12.6 adds to that point by suggesting that different types of information processing may be going on in the head, some using language tokens, and some not, and there is no reason to think that folk psychology distinguishes between them. One of the ideas discussed in more detail here, picking up on ideas in Sect. 12.5, is that dispositions to behave (including linguistic or speech behaviour) can be the basis for belief attribution even if there are no sentences in the head causing that behaviour. In Sect. 12.7, that idea will be used to motivate the claim that in some cases there may be a sufficiently rich body of behaviour to warrant belief attribution without language tokens in the head causing that behaviour, but still further types of problem solving may require the temporary use or representation of linguistic items in the head. This will be described as a kind of moderate language of thought: the presence of linguistic items in the head are not needed as a basis for propositional attitude ascription (strong language of thought), but they may be needed for other things. Finally, in Sect. 12.8, the point is taken even further, showing that a moderate language of thought would be compatible even with the elimination of folk psychology.

Throughout the paper, possibilities will be multiplied. The goal will not be to catalogue all the options available for computational cognitive modeling; the focus will be on disentangling some of the different forms of eliminativism and language of thought. There is insufficient space to evaluate all the different positions (and possible combinations of positions) that will be raised. The hope is to add to some past attempts – Clark (1993a) and Chemero (2009, chapter 2) – to get a sense of the lay of the land.

12.2 Eliminativisms

Let us characterize eliminativism with respect to folk psychology as the view that for (some) advanced research purposes, our common sense theory of the mental is inadequate. Perhaps small parts of folk psychology need to be eliminated; perhaps many; perhaps the whole thing will disappear – eliminativism comes in degrees. Regarding the motivation for elimination, there could be many. Paul Churchland

(1989) has long argued that folk psychology will be displaced because it is an empirically inadequate theory of the mind. It is alleged that its basic categories are so flawed, it cannot underwrite an empirically sophisticated scientific psychology. A less appreciated theme in Churchland's work is that there are philosophical benefits to the elimination of folk psychology. For years he has attempted to scout out problems in philosophy that he thinks cannot be resolved because they presuppose an underlying view about cognition that is radically flawed. Decades ago, Churchland (1989) was using artificial neural networks to gesture toward a non-sentential epistemology that could help overcome problems with theories of explanation in philosophy of science. In his 2012 book, *Plato's Camera*, he applies his idea that neural networks can be understood as implementing high-dimensional maps to problems in the philosophy of science. Moreover, speculation about computational neural modeling informing moral philosophy has been around for some time (Churchland 1989, chapter 14). In other words, the purported benefits of elimination need not be of a strictly empirical/scientific nature. One of the morals of his work is that philosophical and normative categories are not entirely insulated from empirical/scientific research. A different approach to the elimination of some folk psychological categories may be motivated by largely conceptual and normative considerations. For example, when Robert Brandom (2000, p. 174; see also 1994) is in a naughty mood, he remarks, "I do not officially believe in beliefs." He thinks that the notion of belief is sufficiently problematic for theorizing about the normative statuses at work in intersubjective information management that he gives up on belief in favour of a theoretically regimented notion of *commitment*. To put his point less paradoxically, Brandom is not *committed* to beliefs; alternatively, he can say that "belief" is not one of his words (See Brandom 2000, p. 70 to see why the point can be put in the latter way.). This is a very weak eliminativism since only one folk psychological notion has been abandoned, but it does show that the motivation for the elimination of a folk psychological category need not come entirely from empirical/scientific considerations. The "advanced research purposes" referred to above can be of a different sort, including such things as better normative models for intersubjective information management.

Discussing Churchland and Brandom in the same paragraph – that may seem odd, but a brief historical reflection shows otherwise. Long ago, Wilfrid Sellars (1963) emphasized that our self-conception is a corrigible framework, and he also reflected on the role of language in thought. Brandom has been strongly influenced by his work, and Churchland was a student of Sellars. The idea that our self-conception is corrigible (in principle) lays the groundwork for contemporary discussions of eliminativism, even if Sellars himself was not a folk psychological eliminativist. Churchland argues that our self-conception is not only corrigible in principle, but that it is, in fact, radically false. Brandom has been influenced by Sellars' work on intersubjectivity and norms and how they tie into our self-conception as intentional beings, and he is open to revisions in our self-conception. A detailed historical examination of the topics engaged herein is beyond the scope of this paper. Still, it is worth noting that Sellars' shadow is a long one, falling over many discussions in the contemporary philosophy of mind, including this one.

12.3 Paul Churchland: A Folk Psychological Eliminativist, Not an Intentional Eliminativist

Let us characterize intentional eliminativism as the view that for (some) advanced research purposes, intentional idiom needs to be abandoned (in whole or in part, depending on how strongly the position is formulated). Since folk psychological idiom is intentional, it follows that intentional eliminativism entails folk psychological eliminativism, but folk psychological eliminativism does not entail intentional eliminativism. Indeed one has no hope of understanding Paul Churchland's views without understanding the latter point. For decades, Churchland (1989, 2007, 2012) has been using intentional idiom to characterize computational neural modeling. He has been writing about how computational models of neural networks construct *representations of* the world or how they *portray the world*. More recently, he has been emphasizing that they form complex high dimensional maps that are *about* the world. He has suggested that rather than thinking of thoughts as sentences (for example, a belief that *p*) in the head that are about the world, we should see neural networks as constructing maps that, to varying degrees of adequacy, are about or correspond to the world. The *truth* of sentences in the head is replaced with the *representational adequacy* of maps. His work on interpreting computational neural models is filled with idiom that is both intentional and normative. What he avoids is the folk psychological notion of belief. References to aboutness and ofness are not hard to find, though it has long been clear that he has been interested in relational intentionality, not intrinsic intentionality (Churchland 1989, chapter 2; Churchland 1979). No one who writes this way could seriously be interpreted as abandoning intentional idiom, even if he is moving toward either abandoning or reconceiving folk psychological idiom. This is not to say that intentional eliminativism goes undefended; Anthony Chemero (2009) is a recent proponent of that position.

12.4 Languages of Thought

There are different ways of conceiving of what a language of thought might be. On a very weak conception, it may simply amount to the view that sometimes when we think, we are subverbally and consciously speaking to ourselves – talking in our heads, if you will. I do not know of anyone who has denied that we do this – not Dan Dennett, not the Churchlands, and not Chemero. I mention it to set up a spectrum of possibilities for how language and thinking might be related. A much stronger version of the language of thought would claim that there are functionally discrete, language-like entities in the head, and they are organized such that the role of some entities is functionally differentiable from the role of others, where the role of some corresponds to beliefs, and the role of others corresponds to desires, and so on for other propositional attitudes. Talk of a "belief box" or "desire box" is a stand-in for whatever it is that differentiates the attitudes. The idea is to be able to explain thought and action in terms of the computational state transitions between the items in our different propositional attitude boxes. For example, if a student raises her hand in class, it may be because a token in her desire box (*I want to understand this material*) interacted with a token in her belief box (*The way to understand is to ask questions*), and the interaction (as mediated by the appropriate folk psychological law(s)) caused her to raise her hand – and all of this can happen without any subverbal/conscious rehearsing of the language items in question.

In the second to next section we will explore the possibility that language may play a role in thought which is (a) greater than simple subverbal/conscious rehearsal of language items and (b) need not go as far as Fodorian style language of thought – LOT, with stored tokens of beliefs, desires, and the like. Before getting to that we will look a little further at intentional attitude ascription and whether sentences in the head are needed to underwrite them in every instance. We will consider the possibility that behavioural dispositions could be rich enough to warrant the attribution of beliefs even if those beliefs are not caused by sentences in the head. That idea will play a key role in the second to next section.

12.5 Maps, Intentional Attitudes, LOT, and Eliminativism

For the sake of argument, let us say that we are using something like a map in the head when we navigate our spatial environment.[1] Putting things in this way is surely too simple, a point to which we will return shortly, but it is useful as a starting point. Let us say that Jasmine is driving from Montreal to New York. She has done it before and knows the route well. She is following her internal map and doing so without consciously reflecting on what she is doing. If we ask her questions about why she took one turn rather than another, she can extract information from her map to provide answers. She can even ask herself questions: "Did I miss an exit?" or "Was that the right turn to make?" You get the idea. She can use language to query for information on her map, and she can generate answers in a linguistic form. Imagine being in the car with Jasmine; you have been talking to her for some time, and she has not been paying close attention to her driving. You overhear her say *soto foce* and with some concern, "Did I miss an exit?" You may be inclined to attribute to her the belief that she is not sure where she is. For the sake of argument, assume she has a belief box and a desire box. We can imagine that "I am not sure where I am" is *not* in her belief box. Her expression of concern may be the result of wanting to know where she is and not knowing precisely how to position herself on her mental map. She may be inclined to locate herself somewhere in the province of Quebec

[1] See Kuipers (1982) for some longstanding concerns with the idea that we use, in all instances, maps in the head for spatial navigation. For our purposes, we assume Jasmine to have an eidetic memory. We treat her as an exceptional individual capable of carrying out the relevant navigation with a map in the head. It may well turn out that many different strategies are used for spatial navigation. The idea of a map in the head should not be privileged.

and not the state of New York because she does not recall crossing an international boarder, but she is not sure exactly where in Quebec she is. While continuing to drive, she reads a sign that says her exit is in three kilometres. She is immediately reoriented. That reorientation consists in much more than knowing where her next exit is. In virtue of having that information, she can position herself on her map, then other information becomes available (approximately how far she is from the next town, approximately how far she is from the Canada-USA boarder, …). She knows how to locate herself on her internal map, even if that means that *nothing* of the form, "Such-and-such is the next town," or "I am such-and-such a distance from the boarder," are added to her belief box. We may attribute beliefs of that form to her, and if queried verbally about such matters, she would be inclined to answer correctly, but that may be the result of where she places herself on her mental map, not the result of the answers being stored in a linguistic format in her belief box. So while she has the disposition to correctly answer questions about how far she is from the boarder, the map or map-like representation in her head is central to what causes her to say,

P: we are about ten kilometres from the boarder.

It is not that P is stored in her head (in the belief box), and it causally interacts with the desire to be informative, and that leads to the uttering of P when Jasmine is queried in the relevant manner. What we have here is an example of a representational, intentional information structure – the map in the head – making possible certain behavioural (including linguistic) dispositions which underwrite the attributions of certain kinds of beliefs. Granted, maps may contain words or phrases, but the information processing properties of a map in the head would be quite different from the information processing properties of only having sentences in the head. If something like a map in the head strategy were right for even a fragment of cognition, that would be an example of how something representational, but not sentences in a belief box, could play a role in the production of behaviour and dispositions that support the attribution of beliefs. (We are assuming that there is no core or base of internally stored sentences enabling spatial navigation in this case.) While Churchland is very much fond of writing about maps in the head, he avoids writing about the attribution of beliefs – he is, after all, trying to be some kind of eliminativist. However, even if it is the case that we use maps and not sentences in the head, it is hard to see how that is a substitute for the personal-level work that belief attribution does in both the intra- and intersubjective management of information and behaviour (including the normative assessment and the prediction and explanation of behaviour).

Pace Stich and Warfield (1995) and following Clark (1993b, chapter 10) and Dennett (1987) before him, I am not prepared to say that folk psychology is false either in general or for a fragment of cognition if it turns out that there are no sentences in the belief box causing the relevant behaviour. However, *if* it turns out that there are better ways to manage personal-level information processing and behaviour (either intrasubjective or intersubjective) than attributing beliefs, that is a different story *entirely*. Churchland (2012, chapters 4 and 5) does seem to think that saying we have maps in the head helps us to understand better – that is, better than saying we have beliefs – both the doing of science and the philosophy of science at a personal level (so the work that maps are supposed to do is more than explaining subpersonal information processing). If he is right about that, it would be an interesting reason to move toward eliminativism of some sort. Space precludes a proper evaluation of his views on that matter. As indicated above, Brandom can be read as a kind of belief-eliminativist because he thinks belief is a confused notion, and he moves to replace it with a theoretically regimented notion of commitment to do the normative work of personal-level information management. See Cussins (1993) for another argument in favour of belief elimination based on its confused nature. Again, space precludes pausing to evaluate. I raise these latter possibilities to point toward what more interesting reasons for eliminativism might look like. The mere absence of sentences in a belief box (especially in a case like spatial navigation) is unpersuasive as a point in favour of the elimination of belief attribution.

12.6 Languages of Thought, Intentional Attitudes, and Eliminativism

So far, we have considered the possibility that Jasmine could navigate with a map in her head, and that the behaviours caused by the map could lead us to attribute various beliefs to her which are not anywhere stored in a belief box. Now we will consider the possibility that there could be words or sentences in her head at work behind the veil of awareness, but these words or sentences need not be thought of as being in a belief box.

Let us consider a neural network discussed by Ramsey et al. (1990).[2] We will call it the RSG network. It is a three layer, feed forward neural network that takes as input (vectored encoded) sentences or descriptions of various living creatures, and as output it is trained to classify the sentence as true or false. After the network is trained, given the sentence, "Dogs have fur," the output is "true." Given the sentence, "Dogs have scales," the output is "false." You get the idea. These sentences are not stored anywhere in the network. Let us say we take up the intentional stance when looking at the trained but inactive network, and we attributed to it the belief that "Dogs have fur." It is not that the network has that sentence stored in it, and that stored sentence is what causes the disposition to respond in the affirmative. That would be a strong LOT approach to modeling belief. No, *the disposition to respond in the affirmative to the sentence is (part of the reason) why we attribute the belief*

[2] Ramsey, Stich, and Garon claim that if cognition is like the network they discuss, then folk psychological attributions are false. For reasons mentioned above (see the reference to Clark 1993b, chapter 10) I do not interpret their network in that way. It is put to quite a different use in what follows.

in that sentence. Dennett (1987) has been making that sort of point for some time. Moving from a toy model to the head, the idea is that if parts of cognition were "something like" collections of that toy network, then we might attribute a belief that *q* because of the behaviours caused by the networks, including the disposition to assent to *q*. In the base case for LOT, if an agent believes that *q*, it is because the sentence *q* is stored in the belief box, and being stored there causes various behaviours, including assent to *q*. The point is not that LOT is in every case required to attribute a stored sentence in the head to explain what it is to have a belief. Fodor (1987, chapter 1) insists that core or base cases need to be treated in that way. What about non-occurrent or dispositional beliefs? Does Jasmine believe (in some dispositional sense) that a million plus one equals one million and one? Sure she does, but not because she has that belief stored in her head. The strong LOT theorist could appeal to the beliefs that are stored in her head (including beliefs about how addition works) to say that she has the disposition to respond in the appropriate way to the preceding addition problem. In other words, that disposition is based directly and in large part on the beliefs that are *stored as sentences in her head*. What we are considering with the RSG net is something *different* from that. We are considering dispositions to behave (including linguistic assent) that are not explained in terms of causally efficacious sentences stored between the input and output of the network(s). This idea will be central to the next section.

Before going further, it should be pointed out that both LOT and non-LOT approaches could exist side-by-side; they need not be conceived as mutually exclusive options. For example, say Jasmine takes an inventory of things at home before going to work, and she sees that she is low on eggs, milk, and coffee. When stepping outside, she sees her roses suffering and realizes she needs to pick up some rose food. Subverbally, she runs through a list of items and the two stores she needs to stop at before returning home from work. Let us say that the items and the stores at which they are located are stored in memory as sentences. After work, without any conscious rehearsal of those sentences, she stops at the two stores to pick up the four items. For the sake of argument, say the sentences stored in memory are causally efficacious and produce behaviour in the way Fodor envisages when he postulates LOT. That fragment of behaviour would best be explained by LOT. Other fragments might be explained by a non-LOT approach. Consider a second example. Say beliefs attributed to Jasmine about dogs not having scales has nothing to do with sentences stored in the brain and is about her disposition to behave in certain ways (think of the RSG network). As a third example, consider the beliefs attributed to Jasmine during navigation – the boarder is about 10 km away – perhaps such beliefs are based on dispositions to behave that are underwritten by something like a map in the head. Just because the high-level belief attributions of folk psychology do not differentiate between the three examples just discussed, it does not follow that a lower-level discussion of cognitive processes need not differentiate between them. There is a danger in being myopic in theorizing. To be sure, simplicity and explanatory unification are often touted as virtues of theories; however, if their pursuit leads to the misrepresentation of phenomena, their virtue fades. If there are differences in the information the various processing strategies of the brain, then our cognitive models should capture that. Note well: there is no argument here that the three examples just mentioned are truly distinct. The possibility is simply being raised that many different things may be going on that need to be recognized at the level of cognitive modeling that are not recognized in folk psychological attributions.

12.7 A Role for Language in Thought Without the Belief Box

In one of the examples just discussed – Jasmine storing a mental list of items to be purchased – memory was serving as (part of?) the belief box. The strong version of the LOT associated with Fodor treats the belief box as whatever carries out the role of storing sentences in the head such that those sentences play a causally efficacious role in producing behaviour in a way that we would associate with beliefs, and in a way that does not require the conscious rehearsal of the sentences in question. In this section we will consider the possibility that words or sentences may play a casually efficacious role in the head, but not in a way that would make it appropriate to treat them as items in a belief box (or any other attitude box). As well as being casually efficacious, these symbols in the head could operate without conscious rehearsal.

Say Jasmine can correctly answer all kinds of questions about her dog, Lassie. Does Lassie have fur? How about scales? Teeth? Is Lassie a mammal? An inanimate object? An animal? And so on. She can also answer correctly all kinds of questions about her cat, Morris, and her goldfish, Charley. For the sake of argument, we will say that the disposition to answer correctly is not based on sentences stored in the belief box. The processes are "something like" what we see going on in the RSG net. We will say that sufficiently many dispositions are in place that we attribute to Jasmine various beliefs (that Lassie is a dog, has fur, four legs, teeth, no scales, ...; that Morris is a cat, has fur, four legs, teeth, no scales, ...; that Charley is a fish, has no legs, no fur, no teeth, ...). Again, for the sake of argument, assume that for further kinds of processing to take place, words or sentences need to be represented in the head. If Jasmine is asked whether Morris is more like Lassie than like Charley, say that there are linguistic representations at work – behind the veil of awareness – that allow her to map more features and relations from Morris to Lassie than from Morris to Charley. As a result of these processes, of which she is unaware, she produces the response that Morris is more like Lassie than Charley. One of the linguistic representations unconsciously at work might be "Lassie has four legs, fur, teeth, and is an animal and a mammal." It does not follow that such a representation should be said to be in the belief box or is part of the strong LOT model of explanation. Jasmine already has that belief about Lassie (or belief*s* if you prefer to break down the conjunction) before she was asked the similarity question. She had the belief in virtue of various dispositions that were made possible *without* stored sentences. To do certain kinds of processing, such as similarity assessment, it may become necessary to represent words and phrases temporarily. That would amount to a nontrivial role for linguistic items in the head, but it would be something less than a strong LOT.

The scenario just described has both similarities to and differences from the strong and weak approaches to the language of thought described above. On a strong LOT approach, if Jasmine is attributed the belief that Morris is more like Lassie than Charley, that sentence need not be stored in the belief box, but information stored as sentences in the belief box (sentences about the features of dogs, cats, and fish) has to enable the inference to the similarity claim, and those sentences play a role in causing action independent of the kind of similarity mapping just considered. In the previous paragraph, we are imagining that there is sufficient cognitive prowess in place about dogs, cats, and fish to warrant belief attributions *without information being stored as sentences*, but to provide the similarity claim as output, information is temporarily represented internally and in a linguistic manner to help answer a question or solve a problem. This has some resemblance to the idea of temporarily using linguistic items while speaking to oneself consciously but subverbally to solve a problem – no one denies that we do that. The difference is that we are considering the linguistic items to be at work unconsciously to explain problem solving behaviour, which looks more like strong LOT. However, unlike strong LOT, the processes that warrant the attribution of base beliefs do not depend on internally stored sentences, and those processes inform the production and use of internal, linguistic representations, making possible other kinds of linguistically mediated cognitive processing. By contrast, the strong LOT approach starts with stored sentences – which, while they need not be permanent, tend to embody long term commitments – that are the basis of dispositions to produce more sentences as solutions to some problems. The more moderate LOT under consideration starts with dispositions that are the basis for temporary internal linguistic tokens used in other cognition.

The point of considering a more moderate LOT is not to insist on its truth, but to raise a possibility that may be worthy of further investigation. There is research on analogy (Gentner 1988; Gentner and Medina 1998) showing that children who have learned relational predicates are capable of structure-sensitive processing and analogical transference that less linguistically sophisticated toddlers or non-language-using infants are not capable of. This does not prove that linguistic items need to be represented internally to solve certain kinds of problems, but it is conceivable that even at a young age the internal and unconscious representation of symbols facilitates certain kinds of problem solving, and that the basis of the beliefs that allows the problem to be seen in the first place is not a set of sentences in the belief box.

12.8 Still Further Possibilities

Strong LOT is a way of understanding how folk psychology could be true. While Moderate LOT is compatible with folk psychology being true, it does not require its truth. For the sake of argument, let us say that folk psychology is thoroughly false. We need to give up on beliefs and desires; perhaps new propositional attitudes such *newlief* and *newsire* are needed. Or perhaps the replacements will be something we cannot currently imagine. It is at least logically possible that even if that is true, linguistic items play a role in problem solving cognition at an unconscious level (without requiring that they be stored in a newlief box or newsire box). So moderate LOT could be compatible with folk psychological eliminativism. After all, it is not as if eliminativism about folk psychology means we have to stop using language, nor would the evidence that we can solve some problems (better) only after we learn language disappear if folk psychology disappears. And since this paper is about multiplying possibilities for thought, it is worth mentioning that if only parts of folk psychology are eliminated, then it is possible that a moderate LOT could play a role in helping us understand both the processes that make parts of folk psychology true as well as the processes that require radically new theoretical developments. Indeed, still working within the partial elimination framework, the preceding is compatible with strong LOT working side-by-side with moderate LOT in the parts of folk psychology that are true.

Could moderate LOT be compatible with intentional eliminativism? For those who may be sympathetic to Stich's (1985) syntactic theory of mind, the answer may come down in the affirmative. That work can be read as a way of preserving a non-intentional or non-semantic role for linguistic (purely syntactic) tokens in the head.

Of course, not all options or distinctions have been laid out herein. For the most part, this paper has not fussed over the distinction between occurrent and non-occurrent beliefs (or the extent to which the distinction is tenable in the first place). Nor has there been discussion of embodied or extended cognition, which introduces still further complexities with respect to the number of possibilities available. (Perhaps folk psychology is false for the parts of cognition that are extended but not for other parts.... I will spare you all the logically possible permutations.) That said, hopefully enough ground has been covered to give a better sense of what some of the available options in computational cognitive modeling might be.

Acknowledgments An earlier version of this paper was presented at the 2014 International Association for Computing and Philosophy conference in Thessaloniki, Greece. Thanks to the participants for their helpful comments. Thanks also go to both the initial set of conference referees and the second set of referees for post conference, revised papers. Thanks to Vincent Müller for organizing the conference and all the refereeing. Finally, thanks to Lauren Earle for helpful comments on earlier drafts.

References

Brandom, R. (1994). *Making it explicit: Reasoning, representing, and discursive commitment*. Cambridge: Harvard University Press.

Brandom, R. (2000). *Articulating reasons: An introduction to inferentialism*. Cambridge: Harvard University Press.

Chemero, A. (2009). *Radical embodied cognitive science*. Cambridge: MIT Press, a Bradford book.

Churchland, P. (1979). *Scientific realism and the pasticity of mind*. Cambridge: Cambridge University Press.

Churchland, P. (1989). *A neurocomputational perspective: The nature of mind and the structure of science*. Cambridge: MIT Press, a Bradford book.

Churchland, P. (2007). *Neurophilosophy at work*. Cambridge: Cambridge University Press.

Churchland, P. (2012). *Plato's camera: How the physical brain captures a landscape of abstract universals*. Cambridge: MIT Press.

Clark, A. (1993a). The varieties of eliminativism: Sentential, intentional and catastrophic. *Mind and Language, 8*(2), 223–233.

Clark, A. (1993b). *Associative engines: Connectionism, concepts, and representational change*. Cambridge: MIT Press, a Bradford book.

Cussins, A. (1993). Nonconceptual content and the elimination of misconceived composites! *Mind and Language, 8*(2), 234–252.

Dennett, D. (1987). *The intentional stance*. Cambridge: MIT Press, a Bradford Book.

Fodor, J. (1975). *The language of thought*. New York: Thomas Y. Crowell.

Fodor, J. (1987). *Psychosemantics: The problem of meaning in the philosophy of mind*. Cambridge: MIT Press.

Fodor, J. (1991). Replies. In B. Loewer & G. Rey (Eds.), *Meaning in mind: Fodor and his critics* (pp. 255–319). Oxford: Blackwell.

Fodor, J. (2008). *LOT 2: The language of thought revisited*. Oxford: Clarendon Press.

Gentner, D. (1988). Metaphor as structure mapping: The relational shift. *Child Development, 59*, 47–59.

Gentner, D., and Medina, J. (1998). Similarity and the development of rules. *Cognition, 65*(2–3), 263–297.

Kuipers, B. (1982). The "Map in the Head" metaphor. *Environment and Behavior, 14*(2), 202–220.

Ramsey, W., Stich, S., & Garon, J. (1990). Connectionism, eliminativism, and the future of folk psychology. In *Philosophical perspectives* (Vol. 4, pp. 499–533). Reprinted in *Connectionism: Debates on psychological explanation*, 2, edited by Cynthia and Graham Macdonald. Oxford: Blackwell.

Sellars, W. (1963). Empiricism and the philosophy of mind. In W. Sellars (Ed.), *Science, perception and reality* (pp. 127–196). Atascadero: Ridgeview.

Stich, S. (1985). *From folk psychology to cognitive science: The case against belief*. Cambridge: MIT Press.

Stich, S., & Warfield, T. (1995). Reply to Clark and Smolensky: Do connectionist minds have beliefs? In C. Macdonald & G. Macdonald (Eds.), *Connectionism: Debates on psychological explanation* (Vol. 2, pp. 395–411). Oxford: Blackwell.

Chapter 13
A Mechanistic Account of Computational Explanation in Cognitive Science and Computational Neuroscience

Marcin Miłkowski

Abstract Explanations in cognitive science and computational neuroscience rely predominantly on computational modeling. Although the scientific practice is systematic, and there is little doubt about the empirical value of numerous models, the methodological account of computational explanation is not up-to-date. The current chapter offers a systematic account of computational explanation in cognitive science and computational neuroscience within a mechanistic framework. The account is illustrated with a short case study of modeling of the mirror neuron system in terms of predictive coding.

Keywords Computation • Computational modeling • Explanation • Mechanism • Levels • Information-processing

Computational modeling plays a special role in contemporary cognitive science; over 80 % of articles in theoretical journals focus on computational[1] models (Busemeyer and Diederich 2010). The same goes, quite obviously, for computational neuroscience. The now dominating methodology forcefully defended by (Marr 1982) has turned out to be fruitful. At the same time, the three-level account of Marr is not without problems. In particular, the relationship among the levels is interpreted in various ways, wherein the change of level is both the shift of grain and the shift of the boundary of the system under explanation (McClamrock 1991); the proper relation between competence and its realization is not at all clear, neither is the question of whether bottom-up modeling is entirely mistaken, and whether one model should answer the how, what and why questions related to the explanandum.

[1] I am *not* using the word 'computational' here in the sense used by Marr to define one of the levels in his account.

M. Miłkowski (✉)
Institute of Philosophy and Sociology, Polish Academy of Sciences, ul. Nowy Świat 72,
00-330 Warsaw, Poland
e-mail: mmilkows@ifispan.waw.pl

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_13

My goal in this chapter is to offer a descriptive account, which is close in spirit to the recent developments in the theory of mechanistic explanation (Bechtel 2008; Craver 2007; Glennan 2002; Machamer et al. 2000). According to mechanists, to explain a phenomenon is to elucidate its underlying mechanism causally, by supplying a model of its causal structure. While mechanisms are defined in various ways, the core idea is that they are organized systems, comprising of causally relevant component parts and operations (or activities). Parts of the mechanism interact and their orchestrated operation contributes to the capacity of the mechanism. Mechanistic explanations abound in special sciences such as biology (Craver and Darden 2013) and neuroscience (Craver 2007) and it is hoped that the adequate description of the principles implied in explanations generally accepted as sound will also furnish researchers with normative guidance.

The claim that computational explanation is best understood as mechanistic – wherein the mechanisms in question are limited to those whose function[2] is to compute – has been defended by Piccinini (2007) and myself at length elsewhere (Miłkowski 2013). Here, I wish to succinctly summarize the account and, more importantly, add some crucial detail to the overall mechanistic framework proposed earlier. I cannot discuss Marr's theory in detail (but see (Miłkowski 2013, pp. 114–121)) and it is used only for illustration purposes. My remarks below are not meant to imply a wholesale rejection of his methodology; it has proven successful but remains somewhat confusing in interdisciplinary contexts, where explanations require more than three levels of mechanisms (McClamrock 1991).

Marr's account did not involve any theory of how computation is physically realized, and it is compatible with a number of different accounts. I will assume a structural account of computational realization here, defended also by Piccinini (2008a) and Chalmers (2011). For an extended argument, see also (Miłkowski 2011, 2013).

One particular claim that is usually connected with the computational theory of mind is that the psychologically relevant computation is over mental representation, which leads to the language of thought hypothesis (Fodor 1975). Here, no theory of mental representation is presupposed in the account of computation, one of the reasons being that representation is one of the most contentious issues in contemporary cognitive science. As the present account is intended to be descriptively adequate, assuming one particular theory of representation as implied by computation would make other accounts immediately non-computational, which is absurd. Another reason is that mechanistic accounts of computation do not need to presuppose representation (Fresco 2010; Piccinini 2008b), though they do not exclude the representational character of some of the information being processed. In other words, it is claimed that only the notion of information (in the information-theoretic sense, not in the semantic sense, which is controversial) is implied by the notion

[2] In this chapter, I do not go into detail of how the notion of function is best understood in the case of computing mechanisms. See however (Miłkowski 2013).

of computation (or information-processing).[3] By "information" I mean quantitative structural-information-content in MacKay's sense of the term: the physical vehicle must be capable of taking at least two different states to be counted as information-bearing (for a detailed explication of the notion of structural-information-content and its relation to selective-information, i.e., Shannon information, see (MacKay 1969); for more on the notion of information, see Miłkowski (2013, Chapter 2)).

13.1 Basic Assumptions of the Framework

13.1.1 Explanandum Phenomenon

Marr stressed the importance of specifying exactly what the model was supposed to explain. Specifying the explanandum phenomenon is critical also for the mechanistic framework, as several general norms of mechanistic explanation are related to the specification of the capacity of the mechanism. All mechanisms posited in explanations have an explanatory purpose, and for this reason their specification is related to our epistemic interest. For the same reason, the boundaries of the mechanism, though not entirely arbitrary, can be carved out in different ways depending on what one wishes to explain.

The explanandum phenomenon has to be described precisely in a mechanistic model, otherwise the model's use and value will be unclear. The specification of the model is not to be confused with raw, unrefined observation or common-sense intuition about the capacity under consideration. The specification of the capacity may be (and usually is) improved during the modeling process, wherein the model allows us to understand the capacity better. What the mechanistic model explains is the real mechanism. At the same time, how it is carved out depends on one's explanatory interests: The explanandum phenomenon is delineated in what was called "the model of data" in the philosophy of science (Suppes 1962), or a theoretical account of observational data.[4] For example, models of language production usually presuppose that the user's productivity is the phenomenon to be explained, even though it is impossible to empirically observe a language user producing an infinite set of sentences. If there are theoretical reasons to believe that language users have this capacity, it will be described in a model of data. In this respect, mechanistic explanation is in accord with Marr's plea for explicit specification of *what* is computed.

[3] The notion of information will be fully explicated in Sect. 13.1.3. Note that I do not endorse the view that digital computation cannot be understood in terms of Shannon information, which has been argued by Piccinini and Scarantino (2010). They qualify the view somewhat in a later paper (without mentioning that they change their earlier view), see (Piccinini and Scarantino 2011).

[4] The distinction between data and the model of data has been rediscovered later by Bogen and Woodward (1988) in their distinction between data and phenomena.

To some degree, the specification of the explanandum phenomenon corresponds to a description of the cognitive competence (understood generically as the capacity of the mechanism). However, in contrast to traditional competence accounts (Marr 1982; Newell 1981), descriptions of the explanandum need not be necessarily idealized.[5] Also, the competence is explained with realization, and its realization by underlying levels of the mechanism is explanatorily relevant. This stands in contrast to traditional symbolic cognitive science, which takes competence accounts to be necessarily idealized and ignores the role of realization in explaining competence.

In the case of computational explanations, the phenomenon to be explained is a capacity to compute, or to process information. Depending on one's explanatory interest, the specification of this capacity may include the exact timing of processing or not. In the case of psychological explanations, time considerations are important, so they are usually included (Meyer et al. 1988; Posner 2005). Similarly, one may also specify the capacity in a counterfactual manner as a specific mathematical function that operates on any integer values, even if only a proportion of such operations have been observed. Having said this, the necessary ingredients of such a specification are at least a proportion of the output and input values of the computational process.

13.1.2 Explanatory Focus and Scaffolding

In the context of computational modeling, which nowadays uses different computer simulations and embodied robots, it becomes clear that the properties of a model are not limited to the ones related directly to the explanandum phenomenon. For example, a robotic model of cricket phonotaxis (Webb 1995) has to include – for technical reasons – a circuit board, even if there is nothing that corresponds to the board in the cricket. Such boards are ignored when evaluating the adequacy of the robotic explanation. I propose to distinguish the *explanatory focus* of the model from its *scaffolding*, which is the supporting part of the model. In particular, all embodied mechanistic models are trivially *complete* as causal mechanisms (all causal factors are always included in a real entity), while their explanatory focus may still include gaps: we may still not know how certain properties of the insect give rise to the explanandum phenomenon even if we have a robotic replica. In other words, a physical replica may as well be a sketch or a schema. The same goes for purely computational models that contain numerous ad hoc additions (Frijda 1967; Lewandowsky 1993). These additions need not be parts of the explanatory focus (note: ad hoc additions may turn out to be explanatorily crucial, even if they are serendipitous).

[5] For example, Newell claimed that the knowledge level, as related to rationality, is always idealized. These idealizations can prohibit mechanistic decomposition (Dennett 1987, pp. 75–76).

Whenever the causal model of the explanatory focus of the mechanism is complete with respect to the explanandum phenomenon (note: not complete in an absolute sense), the model is a mechanistic how-actual explanation; if the model includes some black boxes whose function is more or less well-defined, it is a mechanism schema; otherwise, it remains a mechanism sketch.[6] Note that even a grounded, embodied, robotic model of visual perception may still be a mechanism sketch with respect to human vision. Also, a model in which the explanatory focus is just a minor part of the mechanism, while the parts included in the scaffolding are predominant, violates the principle of parsimony.

Let me elaborate. The distinction between the explanatory focus and scaffolding depends on the use of the model, and is not intrinsic to the model itself. However, as models are considered to be explanatory only if they are validated, their validation will clearly show which parts of the model are held to be in correspondence with reality, and which belong to the scaffolding (see also (Miłkowski 2015)). For example, a robotic gecko can be evaluated just by checking whether the robot is able to hang on the ceiling, however, for that one does not need to build a whole gecko (Sanz and Hernández 2010). In this case, the scaffolding of the model is the whole robot with the exception of its feet; for this reason, the model counts as mere gimmickry.

13.1.3 Three Levels of Constitutive Explanation

Constitutive mechanistic explanation is the dominant form of computational explanation in cognitive science. This kind of explanation includes at least three levels of the mechanism: a bottom (−1) level, which is the lowest level in the given analysis and describes the internals of mechanism parts and their interactions; an isolated (0) level, at which the parts of the mechanism are specified along with their interactions (activities or operations); and the contextual (+1) level, at which the function of the mechanism is seen in a broader context (e.g., how inputs and outputs of the computer connect with the surrounding machinery). Note that the bottom level of the explanation depends on explanatory practices of the scientific community and is not to be confused with the fundamental physical level, described by an ideal physical theory (Machamer et al. 2000). For example, evolutionary biology does not require its mechanisms to be bottomed out at a quantum level at all. Note also that in contrast to how Marr (1982) or Dennett (1987) understand them, levels here are not just different perspectives or stances; they are levels of *composition*. They are tightly integrated but not entirely reducible to the lowest level.

Computational models explain how the computational capacity of a mechanism is generated by the orchestrated operation of its component parts. To say that a

[6] These distinctions were used by Craver (2007), but were unrelated to the distinction between the scaffolding and the explanatory focus.

mechanism implements a computation is to claim that the causal organization of the mechanism is such that the input and output information streams of the mechanism are causally linked and that this link, along with the specific structure of information processing, is completely described.[7] Importantly, the link might be cyclical and as complex as one could wish. For example, imagine a Turing machine built of LEGO blocks. It has symbols built of LEGO blocks that are moved during an "erase" or "write" operation. However, these blocks are not the only relevant crucial factors in this mechanism. There is a head of the machine capable of erasing and writing symbols, and both the causal structure of the head and its control mechanism are necessary for the description of the complete causal structure of the mechanism in question. In other words, LEGO Turing machine symbols by themselves do not cause anything; they are causally relevant factors in a complex control system.

In the present account, computation is equated with information-processing. The notion of information is crucial in models of computation for the account of implementation: a computational process is one that transforms the stream of information it receives as input into a stream of information for output. During the transformation the process may also appeal to information that is part of the very same process (internal states of the computational process). Information may be – although need not be – digital–that is, there is only a finite, denumerable set of states that the information vehicle can have and that the computational process is able to recognize, in addition to processing as output.[8]

There are two ways in which computational models may correspond to mechanisms: first, they may be *weakly equivalent* to the explanandum phenomenon, in that they only describe the input and output information; or *strongly equivalent*, when they also correspond to the process that generates the output information. Note that these notions have been used in methodology of computer simulation since the 1960s (Fodor 1968, Chapter 4). Only strongly equivalent models are explanatory according to the mechanistic framework.

[7] As one of my reviewers noticed, this is a controversial claim. I do not deal with skepticism about physical computation here because all skeptical arguments either fail or involve skepticism about any empirical science, which does not make such skepticism particularly interesting; for my extended discussion with Putnam and Searle, see (Miłkowski 2012b, 2013, pp. 25–85); see also (Buechner 2008).

[8] In analogue computing, the range of values in question need not be restricted, but may be continuous, i.e., infinite. Note that this is not a *definition* of analogue computation; there might be analogue computers that rely on, for example, potentiometers changing their values in a step-wise manner. I simply do not exclude the conceptual possibility of analogue hypercomputation (Siegelmann 1994). For a recent analysis of analog/digital and continuous/discrete distinctions see (Maley 2010).

13.1.4 Mechanistically Adequate Model of Computation

The description of a mechanistically adequate model of computation usually comprises two parts: (1) an abstract specification of a computation, which should include all the causally relevant variables; (2) a blueprint of the mechanism at all levels of its organization. I will call the first part *formal model of the mechanism* and the second *instantiation blueprint* of the mechanism (for a more detailed study of the relationships between these parts of mechanistically adequate models, see (Miłkowski 2014)). While it should be clear that a formal model should be included, it is probably less evident why the instantiation blueprint is also part of the mechanistically adequate model. The causal model must include all causally relevant parts and operations without gaps or placeholder terms (think of generic and unspecific terms such as "representation" or "activation"). Yet formal models cannot function as complete causal models of computers. For example, to repair a broken old laptop, it is not enough to know that it was (idealizing somewhat) formally equivalent to a universal Turing machine. Similarly, how mental deficits will manifest themselves is not obvious based on a description of ideal cognitive capacity. One needs to know its implementation.

The mechanistic model of a computational phenomenon cannot be limited to its formal properties (Miłkowski 2011). Accordingly, merely formal models of, for example linguistic competence, which abstract away from its realization, are assessed as essentially incomplete. They are either mere specifications of the explanandum phenomenon, but not explanatory in themselves, or, when accompanied with a rough theory of how they are related to experimental data, mechanism sketches (Piccinini and Craver 2011). This means that computational explanations of psychological capacities need to be integrated, for completeness, with models of their realization. Otherwise, they may posit epiphenomenal entities without any causal relevance. Contrary to the functionalist theory of psychological computational explanation (Cummins 1983), mechanism requires it to be causal. It follows that some symbolic models in psychology, even if they are weakly equivalent to the model of input/output data, are not considered to be fully explanatory because of the inherent danger of positing entities that are causally and explanatorily irrelevant. In other words, computational structures posited in such theories as explanatorily relevant can turn out to play a merely supportive role of the scaffolding.

Just because the description of the computational mechanism usually involves two different models, the formal one and the instantiation blueprint, and these may be idealized, computational modeling requires complex integration, similar to one described as multiple-models idealization (Weisberg 2007). Notice that a formal model (usually) corresponds only to the level 0 of the instantiation blueprint, while other levels of the blueprint do not stand in correspondence to the model of computation. For example, a formal model of the computation (usually) does not include the environment in which the computation takes place nor does it contain any reference to underlying machinery.[9]

Note that my mechanistic account of computation does not stipulate a single formal model of computation that would fit all purposes. Rather, it adheres to transparent computationalism (Chrisley 2000): any formal model that can be specified in terms of information-processing is fine here, be it digital, analog or hybrid, as in contemporary computational neuroscience (Piccinini and Bahar 2013). This is not to be confused with multiple realizability of computation: a particular mechanistic model may or may *not* be multiply realized (elsewhere I defended the claim that multiple realization is irrelevant for computationalism; see (Miłkowski forthcoming)). The point is that philosophers should not define the notion of computation, and leave this task to computer scientists and mathematicians.

The empirical adequacy of the mechanistically adequate model of computation can be tested. As such models are strongly equivalent to processes being modeled, usual process-testing methods apply, including chronometry (Posner 2005), various kinds of experimental and natural interventions (Craver 2007), brain imaging – though with usual caveats (Trout 2008), and task decomposition (Newell and Simon 1972). All in all, the more independent observables are tested, the more robust the model. Note that the phenomenological validation modeled after the Turing test (Turing 1950) is not taken to be evidence of the model's empirical adequacy.

13.1.5 Marr's Cash Register

The account may be illustrated with the example used by Marr (1982, pp. 22–24): a cash register in a supermarket. The explanandum phenomenon is the capacity to add prices of individual items and determine the overall sum to be paid. At the contextual level, one describes the cash register as playing a certain role in the supermarket, by allowing easy calculation of the sum to be paid, and making the work of the cashier clerk easier. This includes a bar-code scanner, a conveyor belt, etc. At the isolated level, a dedicated computer using special software is described. The constraints mentioned by Marr, such as commutativity or associativity of addition, are included in the description of the software. Yet without describing the machine that can run the software, this level of description is incomplete. Various failures of the cash register (e.g., dimming of the display), can be explained not only in terms of software bugs but also as hardware failures. Also, the particular display configuration, which can be related to user preferences at the contextual level, is usually not described fully in the software specification. It is at the isolated level

[9] Notice the *caveat*. In principle, it's possible to create a formalism that would both specify the computation, for example as a computer program in LISP, and required machinery. There's no contradiction involved. But, as a matter of fact, modelers don't use such models in their practice, and prefer functional separation of different types of models.

where one describes the physical machine that can display the product name for the cashier clerk and, more fundamentally, can run code by reading it from external memory (not all computers do so; a mechanical cash register, even if it performs computations, cannot run different software). The formal description, usually in terms of the programming language or diagrams, is put into correspondence with the machine. At the bottom level, the operations of the electronic parts of the machine are explained by reference to their properties, relationships, and organization. Just because vast differences between different types of registers are possible (witness the differences between the self-checkout register and the ones used during the American Civil War), the exact explanations will differ. Also, self-checkout machines will have the capacity to collect cash automatically, which needs to be explained as well (the explanandum will be different), and so forth.

The purpose of this toy example is to show that the mechanistic explanation differs a bit from Marr's account by explicitly tightly integrating the levels. Also, at all levels one can ask the why-question: why is the design appropriate for the user? Why does the cash register appropriately display figures on the screen? Why does it save energy? The how-answer is specified at a lower level, and the lowest level depends on our epistemic interest. The what-question also concerns operation of all levels.

13.2 Case study: Predictive Coding in Mirror Neurons

To demonstrate the degree of methodological guidance that is offered by the mechanistic account of computational explanation, let me briefly describe a recently proposed model of action-understanding in terms of predictive coding (Kilner et al. 2007). Predictive coding is one of the Bayesian frameworks and is now gaining considerable recognition (Clark 2013; Hohwy 2013). In the model, it is presupposed that this capacity is realized by the mirror-neuron system (MNS henceforth).[10] The explanandum phenomenon, or action understanding, is described at four levels of hierarchy: (1) the intention-level, which includes long-term goals of actions; (2) the goal-level, which includes short-term goals necessary to realize (1); (3) the kinematic level, which is the shape of the movement of limbs in space and time; and (4) the muscle level, which is the pattern of muscle activity underlying the action (Hamilton and Grafton 2006). People have visual access only to (3) of other agents. Moreover, the same kinematic level information is correlated to different intentions: Mr. Hyde might hurt someone with a scalpel by making the same movements as Dr. Jekyll (Jacob and Jeannerod 2005). What needs to be explained, therefore, is

[10] For my purposes, it is quite irrelevant whether this account of MNS is correct or not (but see (Hickok 2014; Lingnau et al. 2009); for a recent review see (Kilner and Lemon 2013)). I am merely interested in how the model is vindicated by its authors and how it should be evaluated from the mechanistic standpoint.

how one understands actions, given ambiguous visual information; the constraint of the model is that such understanding is to be realized by MNS. Note, however, that the assumption of ambiguity may be false, as it has been shown that different action intentions lead to different perceivable kinematics (Ansuini et al. 2008, 2015). Naturally, given relatively scarce evidence about the details of MNS, the model might be currently only biologically plausible. In mechanistic terms, it cannot be a how-actually model, as we lack observables which could confirm that causal factors in the model are actual. We may have only a how-plausible model (for more on this distinction, see (Craver 2007)), which should ascribe a precise computational role for MNS.

Kilner, Friston & Frith note that other similar explanations of action in terms of MNS posit forward or generative models. Yet these explanations cannot deal with the fact that people easily distinguish between the action of Dr. Jekyll and Mr. Hyde. In other words, they do not explain one important part of the phenomenon.

The contextual level of the proposed predictive coding mechanism includes the context in which the action is observed (e.g., the operation theatre vs. the dark streets of London). The context of action, which is not coded by MNS, is hypothesized to be represented by other parts of the larger hierarchy, where intentions are encoded (Kilner et al. 2007, p. 164). Note that such hierarchy can be naturally accounted for in the mechanistic framework, while in the Marrian methodology, nested hierarchies of mechanisms are still analyzed merely on three levels, which are not levels of composition, as in Kilner et al.'s chapter (this makes the analysis of the model in Marrian terms all the more difficult).

The 0 level of the mechanism is then described as performing predictive coding of action, i.e., the mechanism predicts the sensory consequences of movements, and the prediction error is minimized through recurrent or reciprocal interactions among levels of a cortical hierarchy. This means that the mechanism posited by authors comprises more than just three levels, which is the minimal number for constitutive explanations. Here, the upper level mechanism employs a generative model to predict representations in the level below. Backward connections are used by the upper level to convey the prediction to the lower level, which is used to produce information about prediction error. The instantiation blueprint of the mechanism includes this hierarchy whose architecture allows for adjustment of the neural representations of actions in terms of sensory representation of causes of action if prediction error is found. The architecture is self-organizing, and the reciprocal exchange of signals continues until the error is finally minimized.

The formal model of the neural architecture is described here in terms of empirical Bayesian inference (Friston 2002, 2003, 2005): the prior expectations are generated by the self-organizing information-processing architecture. In other words, this model includes, as usual, two complementary parts: the instantiation blueprint, characterized in terms of that which is known about MNS, and its formal computational specification. Contrary to the programmable cash register, no stored-program computer is posited.

The bottom level is merely touched upon; there is no extensive discussion of the precise realization of predictive coding by elementary entities of the neural system.

Thus, this model is, at best, a mechanism schema, because it does not explain how MNS comes to operate as it does. The authors stress that to test the model, one would need to characterize the nodes of the cortical hierarchy anatomically and functionally, and such characterization is not available.

The neural plausibility of the predictive coding and its relation to empirical Bayesian modeling is the focus of much current discussion (Blokpoel et al. 2012). In particular, the question is whether the biologically plausible implementation of the predictive coding is equivalent to empirical Bayes or not (it may possibly approximate empirical Bayes somewhat). The mechanistic explanation requires that the mechanisms be not idealized in such a way that would require putting tractability questions (Van Rooij 2008) to the side. The data in the original paper makes it impossible to answer critical questions about the mechanism in this context, such as the number of inputs in the Bayesian network, which is essential in assessing the parametrized complexity of the algorithm.

Were the model implemented on the computer, the results of the simulation could be compared to those observed in humans or in macaque monkeys. Alas, no such results are reported by Kilner et al., and since without implemented models detailed testing of hypotheses is impossible, the empirical adequacy of the explanation is not entirely clear. To assess the adequacy properly, one should rather implement several comparable models of the same explanandum phenomenon, which can also help to avoid the confirmation bias to which researchers are prone (Farrell and Lewandowsky 2010; Miłkowski 2013, p. 86).

Some Bayesian theories in psychology were recently criticized as fundamentalist, i.e., dogmatically trying to model behavior as rational and without mechanistic constraints (Jones and Love 2011). Note that this is not true of the model under consideration; Bayesian modeling in neuroscience is obviously related to functioning of the brain. Instead of stressing the contrast between the mechanistic account of computational explanation and Bayesian modeling, my intention is to show that the mechanistic framework can be used to evaluate the contribution of the given model to progress the understanding of the explanandum phenomenon.

Summing up this part of the discussion, the mechanistic framework makes it easy to assess the maturity of the model in terms of its completeness and empirical adequacy. Because the computer implementation is lacking, it is impossible to say whether the model contains multiple empirically undecided decisions that are needed to make it run (hence focus/scaffolding evaluation is impossible). At the same time, there is no information about the bottom level. On the contextual level, placeholder terms such as "intention encoding" are used and they need further explanations in other models. Thus, the model does not include a complete specification of the mechanism.

Also, it is not at all clear how long-term goals might be understood in terms of mere sensory input prediction. Dr. Jekyll's intention to heal a patient (long-term goal) does not seem, prima facie, to be represented just in sensory terms. If it is actually so represented, the model does not explain how. This makes it a mechanism sketch, so its explanatory value is, qualitatively speaking, on a par with traditional symbolic models of competence. (Quantitative evaluation is impossible here, as no results of experiments on computer implementation were reported.)

13.3 Conclusion

The mechanistic account of computational explanation preserves the insights of Marr but is more flexible when applied to complex hierarchical systems. It may help to integrate various different models in a single explanation. Mechanistic methodological principles are inferred from research practice in life sciences, neurosciences, and cognitive science. Also, by subsuming computational explanation under causal explanation, the mechanistic account is naturally complemented by methodology of causal explanation; one particularly promising framework is the interventionist account of causal explanation (Pearl 2000; Spirtes et al. 2000; Woodward 2003).[11]

By allowing multiple nested hierarchies, the standard three-level constitutive explanation is naturally expanded when needed. There is also no danger in preferring only the contextual level in the explanation, as it does not furnish us with the constitutive causal factors. The bottom level will also not obviate the need for the contextual level as it does not contain some of the entities which are found at the contextual level. For example, the encoding of intention is not realized by MNS only, so its explanation cannot be 'reduced' to the description of the lower levels.

The present theory is not intended to settle debates over matters in which modelers explicitly disagree; the only goal is to make as much sense of various modeling approaches as possible, and make cross-approach comparisons possible by showing the common ground between them.

It is also not presupposed that computational explanation is the only proper way to explain cognition (Miłkowski 2012a). On the contrary, only some part of the mechanism model is strictly computational (i.e., uses vocabulary of the theory of computation). The bottom level of the mechanism has to be framed in non-computational terms; otherwise the computational operations of the isolated level are not explained, and may turn out to be spurious (Miłkowski 2013, pp. 82–3). At the same time, the present account leads naturally to explanatory pluralism, as the

[11] It's worth noticing that the interventionist framework is not committed to realism about physical causation. There are interventionists who defend a specific neo-Russellian view on causation, according to which causal explanations in special sciences are genuine, while there might be no genuine causal explanations in fundamental physics (Reutlinger 2013). While I do not espouse such scepticism about causation in physics, nothing in my argument depends on there being physical causation or not; there is also no reliance on causal determinism in my argument. The mathematical framework of interventionism is sufficient to specify the conditions that genuine computational explanations must satisfy to count as good explanations, and realism or anti-realism about causes is not a part of the mathematical framework.

only requirement for the theoretical frameworks used to describe various levels of composition of mechanisms is that they include causally relevant factors.

Acknowledgments The author gratefully acknowledges the support of the Polish National Science Center Grant in the OPUS program under the decision DEC-2011/03/B/HS1/04563. An earlier version of this paper has been presented during the Cognitive Science Annual Conference 2013 in Berlin. The author also wishes to thank the reviewers of the draft of the paper for their helpful and detailed comments, in particular Nir Fresco and Marcin J. Schroeder.

References

Ansuini, C., Cavallo, A., Bertone, C., & Becchio, C. (2015). Intentions in the brain: The unveiling of Mister Hyde. *The neuroscientist: A review journal bringing neurobiology, neurology and psychiatry, 21*(2), 126–135. doi:10.1177/1073858414533827.

Ansuini, C., Giosa, L., Turella, L., Altoè, G., & Castiello, U. (2008). An object for an action, the same object for other actions: Effects on hand shaping. *Experimental Brain Research, 185*(1), 111–9. doi:10.1007/s00221-007-1136-4.

Bechtel, W. (2008). *Mental mechanisms*. New York: Routledge (Taylor & Francis Group).

Blokpoel, M., Kwisthout, J., & van Rooij, I. (2012). When can predictive brains be truly Bayesian? *Frontiers in Psychology*, *3*(November), 1–3. doi:10.3389/fpsyg.2012.00406.

Bogen, J., & Woodward, J. (1988). Saving the phenomena. *Philosophical Review, 97*(3), 303–352.

Buechner, J. (2008). *Godel, Putnam, and functionalism: A new reading of representation and reality*. Cambridge: MIT Press.

Busemeyer, J. R., & Diederich, A. (2010). *Cognitive modeling*. Los Angeles: Sage.

Chalmers, D. J. (2011). A computational foundation for the study of cognition. *Journal of Cognitive Science, 12*, 325–359.

Chrisley, R. (2000). Transparent computationalism. In M. Scheutz (Ed.), *New computationalism: Conceptus-Studien 14* (pp. 105–121). Sankt Augustin: Academia Verlag.

Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *The Behavioral and Brain Sciences, 36*(3), 181–204. doi:10.1017/S0140525X12000477.

Craver, C. F. (2007). *Explaining the brain. Mechanisms and the mosaic unity of neuroscience*. Oxford: Oxford University Press.

Craver, C. F., & Darden, L. (2013). *In search of mechanisms: discoveries across the life sciences*. Chicago/London: The University of Chicago Press.

Cummins, R. (1983). *The nature of psychological explanation*. Cambridge: MIT Press.

Dennett, D. C. (1987). *The intentional stance*. Cambridge: MIT Press.

Farrell, S., & Lewandowsky, S. (2010). Computational models as aids to better reasoning in psychology. *Current Directions in Psychological Science, 19*(5), 329–335. doi:10.1177/0963721410386677.

Fodor, J. A. (1968). *Psychological explanation: An introduction to the philosophy of psychology*. New York: Random House.

Fodor, J. A. (1975). *The language of thought* (1st ed.). New York: Thomas Y. Crowell Company.

Fresco, N. (2010). Explaining computation without semantics: Keeping it simple. *Minds and Machines, 20*(2), 165–181. doi:10.1007/s11023-010-9199-6.

Frijda, N. H. (1967). Problems of computer simulation. *Behavioral Science, 12*(1), 59–67. doi:10.1002/bs.3830120109.

Friston, K. (2002). Functional integration and inference in the brain. *Progress in Neurobiology, 68*(2), 113–43.

Friston, K. (2003). Learning and inference in the brain. *Neural Networks, 16*(9), 1325–52. doi:10.1016/j.neunet.2003.06.005.

Friston, K. (2005). A theory of cortical responses. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 360*(1456), 815–36. doi:10.1098/rstb.2005.1622.

Glennan, S. S. (2002). Rethinking mechanistic explanation. *Philosophy of Science, 69*(S3), S342–S353. doi:10.1086/341857.

Hamilton, A. F. de C., & Grafton, S. T. (2006). Goal representation in human anterior intraparietal sulcus. *The Journal of neuroscience: The official journal of the Society for Neuroscience*, *26*(4), 1133–1137. doi:10.1523/JNEUROSCI.4551-05.2006.

Hickok, G. (2014). *The myth of mirror neurons: The real neuroscience of communication and cognition*. New York: WW Norton.

Hohwy, J. (2013). *The predictive mind*. New York: Oxford University Press.

Jacob, P., & Jeannerod, M. (2005). The motor theory of social cognition: A critique. *Trends in Cognitive Sciences*, *9*(1), 21–25.

Jones, M., & Love, B. C. (2011). Bayesian fundamentalism or enlightenment? On the explanatory status and theoretical contributions of Bayesian models of cognition. *Behavioral and Brain Sciences, 34*(04), 169–188. doi:10.1017/S0140525X10003134.

Kilner, J. M., Friston, K. J., & Frith, C. D. (2007). Predictive coding: An account of the mirror neuron system. *Cognitive Processing, 8*(3), 159–66. doi:10.1007/s10339-007-0170-2.

Kilner, J. M., & Lemon, R. N. (2013). What we know currently about mirror neurons. *Current Biology, 23*(23), R1057–R1062. doi:10.1016/j.cub.2013.10.051.

Lewandowsky, S. (1993). The rewards and hazards of computer simulations. *Psychological Science, 4*(4), 236–243. doi:10.1111/j.1467-9280.1993.tb00267.x.

Lingnau, A., Gesierich, B., & Caramazza, A. (2009). Asymmetric fMRI adaptation reveals no evidence for mirror neurons in humans. *Proceedings of the National Academy of Sciences of the United States of America, 106*(24), 9925–30. doi:10.1073/pnas.0902262106.

Machamer, P., Darden, L., & Craver, C. F. (2000). Thinking about mechanisms. *Philosophy of Science, 67*(1), 1–25.

MacKay, D. M. (1969). *Information, mechanism and meaning*. Cambridge: MIT Press.

Maley, C. J. (2010). Analog and digital, continuous and discrete. *Philosophical Studies, 155*(1), 117–131. doi:10.1007/s11098-010-9562-8.

Marr, D. (1982). *Vision. A computational investigation into the human representation and processing of visual information*. New York: W. H. Freeman and Company.

McClamrock, R. (1991). Marr's three levels: A re-evaluation. *Minds and Machines, 1*(2), 185–196. doi:10.1007/BF00361036.

Meyer, D. E., Osman, A. M., Irwin, D. E., & Yantis, S. (1988). Modern mental chronometry. *Biological Psychology, 26*(1–3), 3–67. doi:10.1016/0301-0511(88)90013-0.

Miłkowski, M. (2011). Beyond formal structure: A mechanistic perspective on computation and implementation. *Journal of Cognitive Science, 12*(4), 359–379.

Miłkowski, M. (2012a). Limits of computational explanation of cognition. In V. C. Müller (Ed.), *Philosophy and theory of artificial intelligence* (pp. 69–84). Berlin/Heidelberg: Springer. doi:10.1007/978-3-642-31674-6_6.

Miłkowski, M. (2012b). Is computation based on interpretation? *Semiotica, 188*, 219–228. doi:10.1515/sem-2012-0015.

Miłkowski, M. (2013). *Explaining the computational mind*. Cambridge: MIT Press.

Miłkowski, M. (2014). Computational Mechanisms and Models of Computation. *Philosophia Scientiæ*, *18*(3), 215–228.

Miłkowski, M. (2015). Evaluating artificial models of cognition. *Studies in Grammar, Logic, and Rhetoric*, *40*(1), 43–62. doi:10.1515/slgr-2015-0003.

Miłkowski, M. (forthcoming). Computation and multiple realizability. In V. C. Mueller (Ed.), *Fundamental issues of artificial intelligence*. Berlin/Heidelberg: Springer.

Newell, A. (1981). The knowledge level: Presidential address. *AI Magazine, 2*(2), 1–21.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs: Prentice-Hall.

Pearl, J. (2000). *Causality: Models, reasoning, and inference*. Cambridge: Cambridge University Press.

Piccinini, G. (2007). Computing mechanisms. *Philosophy of Science, 74*(4), 501–526. doi:10.1086/522851.

Piccinini, G. (2008a). Computers. *Pacific Philosophical Quarterly, 89*(1), 32–73. doi:10.1111/j.1468-0114.2008.00309.x.

Piccinini, G. (2008b). Computation without representation. *Philosophical Studies, 137*(2), 205–241. doi:10.1007/s11098-005-5385-4.

Piccinini, G., & Bahar, S. (2013). Neural computation and the computational theory of cognition. *Cognitive Science, 37*(3), 453–88. doi:10.1111/cogs.12012.

Piccinini, G., & Craver, C. F. (2011). Integrating psychology and neuroscience: Functional analyses as mechanism sketches. *Synthese, 183*(3), 283–311. doi:10.1007/s11229-011-9898-4.

Piccinini, G., & Scarantino, A. (2010). Computation vs. information processing: Why their difference matters to cognitive science. *Studies In History and Philosophy of Science Part A, 41*(3), 237–246. doi:10.1016/j.shpsa.2010.07.012.

Piccinini, G., & Scarantino, A. (2011). Information processing, computation, and cognition. *Journal of Biological Physics, 37*(1), 1–38. doi:10.1007/s10867-010-9195-3.

Posner, M. I. (2005). Timing the brain: Mental chronometry as a tool in neuroscience. *PLoS Biology, 3*(2), e51. doi:10.1371/journal.pbio.0030051.

Reutlinger, A. (2013). Can interventionists be Neo-Russellians? Interventionism, the open systems argument, and the arrow of entropy. *International Studies in the Philosophy of Science, 27*(3), 273–293. doi:10.1080/02698595.2013.825497.

Sanz, R., & Hernández, C. (2010). Autonomy, intelligence and animat mesmerization. In C. Hernández (Ed.), *BICS 2010 – Brain Inspired Cognitive Systems*. Madrid: Universidad Politécnica de Madrid.

Siegelmann, H. (1994). Analog computation via neural networks. *Theoretical Computer Science, 131*(2), 331–360. doi:10.1016/0304-3975(94)90178-3.

Spirtes, P., Glymour, C. N., & Scheines, R. (2000). *Causation, prediction, and search* (2nd ed.). Cambridge: The MIT Press.

Suppes, P. (1962). Models of data. In E. Nagel, P. Suppes, & A. Tarski (Eds.), *Logic, methodology, and philosophy of science: Proceedings of the 1960 International Congress* (pp. 252–261). Stanford: Stanford University Press.

Trout, J. D. (2008). Seduction without cause: Uncovering explanatory neurophilia. *Trends in Cognitive Sciences, 12*(8), 281–2. doi:10.1016/j.tics.2008.05.004.

Turing, A. (1950). Computing machinery and intelligence. *Mind*, *LIX*(236), 433–460. doi:10.1093/mind/LIX.236.433.

Van Rooij, I. (2008). The tractable cognition thesis. *Cognitive Science, 32*(6), 939–84. doi:10.1080/03640210801897856.

Webb, B. (1995). Using robots to model animals: A cricket test. *Robotics and Autonomous Systems, 16*(2–4), 117–134. doi:10.1016/0921-8890(95)00044-5.

Weisberg, M. (2007). Three kinds of idealization. *Journal of Philosophy, 104*(12), 639–659.

Woodward, J. (2003). *Making things happen*. Oxford: Oxford University Press.

Chapter 14
Internal Supervision & Clustering: A New Lesson from 'Old' Findings?

Alexandros Tillas

Abstract Learning is most often treated as a psychologically rich process in the extant literature. In turn, this has a number of negative implications for clustering in machine learning (i.e. grouping a set of objects so that objects in the same group – cluster – resemble each other more than they resemble members of other groups) to the extent that psychologically rich processes are in principle harder to model. In this paper, I question the view of learning as psychologically rich and argue that mechanisms dedicated to perception and storage of information could also be used in categorization tasks. More specifically, I identify the minimum resources required for learning in the human mind, and argue that learning is greatly facilitated by top-down effects in perception. Modeling the processes responsible for these top-down effects would make modeling tasks like clustering simpler as well as more effective. For clustering is seen here as building upon associations between perceptual features, while connection weightings and top-down effects substitute external supervision in executing the function of error identification and rewards.

14.1 Introduction

In this paper, I am focusing on learning and suggest a philosophical view of how learning could be more effectively modeled in an artificial system. In doing so, I am focusing on categorization – a specific and relatively simple aspect of learning. Admittedly, trying to extract lessons from theoretical research for a technical discipline like machine learning is not easy. However, suggesting a consistent and economical view of how the human mind – the most effective learner – acquires knowledge could provide positive directions for researchers in machine learning. More specifically, if it is shown that the mind does not need to come pre-packed with in-built concepts but can accumulate information about the world and develop categorization abilities in virtue of a limited repertoire of low-level pattern recognition abilities, then unsupervised machine learning and clustering (i.e.

A. Tillas (✉)
Department of Philosophy, University of Düsseldorf, Düsseldorf, Germany
e-mail: atillas@phil.uni-duesseldorf.de

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_14

grouping a set of objects so that objects in the same group – cluster – resemble each other more than they resemble members of other groups) could be seen as more tenable. Admittedly, this is a rather abstract claim but hopefully it can provide researchers in technical disciplines with a theory that they can apply, develop and optimize.

Furthermore, it is worth clarifying that arguing that machine learning comprises of a combination of bottom-up and top-down processes hardly constitutes a groundbreaking claim. Nevertheless, there is still an important lesson to be learned, one that highlights neglected aspects of the extant literature. Namely, information that plays the role of corrections and rewards does not have to be inbuilt or set externally by AI technicians building the system. Rather the system could acquire this information during encounters with its surrounding environment, and later on use it for classification processes (see below). If this claim is accepted as sound, then new light could be shed onto our understanding of 'clustering' or 'unsupervised machine learning'. For instance, the existing allegedly fundamental differences between 'supervised' and 'unsupervised learning' are questioned. On the other hand, if the suggested view is construed simply as a restatement of a long-learned lesson, then the present contribution could be seen as a systematic review, offered from a conceptual perspective, of what is known in the field of perception and classification.[1] In either case the ultimate target of bringing to bear conceptual issues related to perception and classification that could prove useful for researchers working in technical disciplines is achieved.

In arguing for the claim that appealing to low-level pattern recognition abilities could make clustering more tenable, I show that learning is heavily perceptually driven and that acquired representations can be used in developing categorization, as well as clustering, abilities. More specifically, clustering is seen here as building upon associations between perceptual features. In turn, clustering becomes more affordable to the extent that connections weightings and top-down effects substitute external supervision in executing the function of error identification and rewards. In this sense, instead of appealing or relying on externally imposed errors and rewards, the learning system could simply rely on information gathered through previous experiences and stored in a memory unit. Crucially, this previously acquired information is structured around a notion of similarity that is statistically grounded in connections weightings along with probabilistic & diagnostic information. This stored probabilistic and diagnostic information along with the system's perceptual abilities and top-down influences suffices for categorization and clustering. It is worth clarifying that I do not imply that the present suggestions are readily usable in clustering. Rather I argue that mechanisms developed for perception and storage of information can also be used in categorization tasks.

In the view suggested here, learning is a developmental process that starts as a psychologically non-rich process. In building my case, I identify the minimum

[1] See for example Seymour & Minsky's critique of genetic programming (e.g. 1988) or Dreyfus stance on the uncomputability of common sense (e.g. 1965, 1967, 1972). I owe this suggestion to Mario Verdicchio.

resources required for learning, and show that learning occurs or could occur in virtue of low-level mechanisms. Once again my motivation stems from the idea that psychologically non-rich processes are more readily modeled. It is worth clarifying that I do not imply that all learning occurs in virtue of low-level mechanisms and processes and that nothing cognitively sophisticated might be involved in some learning process. However, to the extent that certain aspects of learning, like categorization, do occur in virtue of low-level mechanisms and could thus be more easily modeled in an artificial system. This is my first target in this paper.

My second target is to use the suggestions about the minimum resources required for learning to shed light onto the process of unsupervised learning. My main suggestion at this point is that top-down effects in perception from information stored in long-term memory could "functionally substitute" the role of 'error' and 'reward' in supervised learning.

14.1.1 What Is Learning?

There are two main strands in the debate about the nature of the learning process. On the one hand, learning is construed as a psychologically rich process. For instance, Fodor (1981) argues that all learning is Hypothesis Formation and Testing. In turn he thinks that learning is untenable and that our concepts are part of our genetic endowment (if they cannot be learned, and given that we have them, they must be innate). Claims about the impossibility of learning have negative implications for machine learning – to the extent that machine learning researchers use such ideas as a general theoretical background. On the other hand, there are views of learning that build upon perceptual experiences and treat learning as psychologically non-rich, (e.g. Barsalou 1999; Prinz 2002). Such views rely heavily on perception and treat learning as reducible to formation of associations between representations. Understanding these views could shed light upon machine learning and especially clustering.

In this paper, I am using a general and rather widely accepted notion of learning and assume that learning amounts to acquisition of information about the world, and storage of this information in a manner that allows the agent to access, retrieve and use this information on demand, even in the absence of the things in the world that the information in question is about. With regards to artificial agents, even though learning could also plausibly be construed as acquisition and storage of information, this information is available to the artificial agent during perception of the appropriate stimulus.

14.2 Preliminaries

Starting from a light-hearted representationalism, and without committing to any epistemological claims, I assume that

- On perception of an object, we form a representation of that object as a whole or of its selectively attended parts.
- Perceptual representations occur at a low neurological computational level (see below), alongside perceptual processes, perceptual biases, learning algorithms and so forth.
- Perceptual representations are the building blocks of concepts, in the case of human agents.
- The main difference between representations at the lower and the higher level is that lower level representations could only be activated in a bottom-up manner. Once parts of a concept though, representations could also be activated in a top-down manner or on demand.

The distinction between a lower and a higher computational level is crucial in clarifying the kind of minimum resources required for learning. I have used the same distinction elsewhere (2010) to distinguish between the innate resources required for concept acquisition. Namely, I have argued that concept acquisition relies heavily upon innate resources at the lower level (mainly in the form of pattern recognition abilities), while there is no innateness at the higher cognitive level. Analogously, low-level innate resources could, in the case of artificial agents, be construed as similarity-spotting abilities. The present proposal builds upon a statistically grounded notion of similarity that does not appeal to inbuilt perceptual primitives as shown below. At the same time, high-level resources could be seen as coming via an external administrator as in the case of supervised learning. The suggestions showing that learning occurs in virtue of low-level mechanisms are used to shed light onto the process of clustering.

It is worth clarifying at this point that 'computational' is used here in a broad sense. That is, the suggestions made about the modus operandi of the perceptual systems as well as claims about classification are not driven by the principles underlying classical computational models. In fact, I am much more sympathetic to views that do not appeal to deployment of symbols in the classical sense of the term. For instance, Barsalou (1999) who elaborates on the nature of processes underlying perception and cognition offers such an alternative to classical computational models. Specifically, he argues that perceptual representations formed (and stored) during encounters with instances of a given category are used in fleshing out a 'frame' representing the category in question. Frames are representational structures carrying information about categories.[2] What is key for present purposes is that for Barsalou, thinking is analogous to perceiving to the extent that the same brain areas that ground perception of (instances of) a given category are reactivated

[2] What is not clear, at least in Barsalou's (1999) paper is whether frames represent information/representations of a particular instance that is used to represent the category as a whole, essentially in the sense that Hume (1739/1978), Berkeley (1710/1957) and more recently Prinz (2002) argue for or whether the process of building a frame stands for a Lockean (1690/1975) process of abstraction. See Barsalou (2005) and Tillas (forthcoming) for a detailed discussion of related issues.

while the subject thinks about that category offline – this is what Barsalou refers to as 'simulation' (simulating the perceptual experience while thinking about it). Differently put, the same representations formed during perception of a given category are activated while tokening the concept of the category in question. For instance, the concept of a TREE will be tokened by virtue of activating the same representations formed and stored during encounters with instances of trees (or information stored in the Tree-Frame). In this sense, and even though this is not part of my agenda here, the suggested view builds upon the principles set by Barsalou rather than those underlying traditional computational models. Thus, cognition occurs by virtue of activating perceptual representations rather than subpersonal amodal symbols in a Language of Thought, e.g. Fodor (1975).

14.3 What Does It Take to Recognize a Pattern?

In this section, I am suggesting a theoretical account of learning that builds heavily upon the modus operandi of our perceptual systems. The view put forth is a development from a view I presented elsewhere (2010) and bears certain similarities to Neisser's (1976) cyclic theory of perception even though, unlike Neisser, I do distinguish between bottom-up and top-down processes. The underlying hypothesis is that perception starts like a bottom-up process and continuing top-down effects influence it. In this sense, the suggested view might be seen as setting off like bottom-up theory of visual perception and continuing it bears similarities to Gregory's (1970) constructivist view. To anticipate my claims bottom-up processes allow the system to kick-start without much in-built information, while top-down effects – more clearly associated with constructivist views – play a role that is functionally analogous to supervision (hence internal supervision).

In the interest of simplicity, I am considering the case of learning what a tree is. At the end of this learning process, a human subject should be in a position to perceive all trees *as trees*, including instances she has not seen before. In the case of human agents, this process constitutes a great part of the process of acquiring the concept tree – the other parts being bringing the mental particular in question under the agent's endogenous control by drawing associations with something over which the agent already has executive control such as utterance of the appropriate word, a goal-directed state etc. (See Tillas 2010 for a detailed discussion). In the case of artificial systems, this process constitutes learning what a tree looks like (or what it is). At the end of this process, the artificial system is able to 'identify' and in turn cluster together tree-related patterns that it has not previously encountered. Crucially, as shown below this process could occur without supervision.

In order for learning to occur, the following elements have to be in place. Roughly, these are the minimum resources required for learning:

1. Raw materials (or representations of particulars). In order for perceptual representations of particulars to be acquired, certain low-level pattern recognition

abilities have to be in place. Crucially, these abilities do not build upon predefined categories (or clusters). Rather they detect simple information like existence of edges or discontinuities in luminance (cf. Marr 1982); motion detection and so forth.

2. A locus in memory (or storage unit in the case of artificial agents) where the raw materials are stored and can be accessed.[3]
3. A computational process, broadly construed, which takes as input the bundle of representations that are being stored in a given locus and gives as output a new abstracted representation – this is the sine qua non of the learning process. Note that the abstracted representation is not identical to any of the input representations and does not merely represent any of the particular instances. Crucially, abstraction occurs in virtue of mechanisms that are inbuilt in all learning systems.[4]

14.3.1 Informational Inputs

Representations formed during perceptual experiences with instances of a given kind are the input to the abstraction process. During encounter with the first instance, a representation is formed and stored in long-term memory. On encounter with a subsequent perceived instance, a further representation is formed. At this stage the agent does not know that this is a subsequent instance of a tree but simply that the occurrent instance is an instance similar to the one previously experienced. A scanning process is initiated and a match is sought for in the subject's memory.[5] In principle, the same scanning process is initiated during encounter with the first instance, but does not yield any matching results given lack of previous experiences with the kind in question.

If a match between the occurrent and stored representations is found, the occurrent representation is stored in memory. Contextual features along with any known information about the current encounter determine the location where the representation in question will be stored. It is worth clarifying that 'location' is used here in a rather broad manner. Similarly, storing representations 'closer together' refers to stronger positive memory effects that stored representations exhibit (Uncapher et al. 2011; Craik et al. 1996). If a match between the occurrent and stored representations is not found, the occurrent representation is classified as new and is stored for further investigation. In cases where no preexisting category is found, two instances of the same kind are still treated as similar to each other. This

[3] Think of 'locus' in terms of Perry's (2001) 'mental files' metaphor.

[4] See Barsalou (2005) for a detailed discussion of perceptually orientated view of abstraction.

[5] This claim enjoys significant empirical support, amongst others, by Spivey and Geng (2001), Chao et al. (1999), Barsalou (1999), Demarais and Cohen (1998), Farah (1995, 1989), Finke (1989), Kosslyn et al. (1995), Brandt and Stark (1997), and Norton and Stark (1971).

is done in virtue of comparing the activational patterns and/or information about the combinatorial arrangements describing the pertinent linkages between the neural activities that grounds representations of the perceived entity at hand. In this sense, comparisons between occurrent and stored representations do not deploy perceptual primitives or independent criteria of sameness.

Finding a matching stored representation entails three things. First, the matching stored representation becomes sub-activated or primed. I am using 'primed' here in the standard sense in the psychological literature according to which exposure to a given stimulus influences a response to another stimulus. With regards to machine learning, this could be couched in terms of a 'slight' increase in a certain node's connection weight. In turn, this influences the way the network is constrained.[6] Second, the sub-activated representation drives selective attention in a top-down manner to the same parts of the currently perceived object that the subject attended during perception of the original instance.[7] Third, finding (and activating) an existing matching representation drives storage of the currently formed representation in the same locus in the subject's long-term memory.

It is worth clarifying that the aforementioned top-down influences are fairly liberal and thus the first encounter with an instance of a given kind does not overly skew perception of subsequent instances. As a result, some non-overlapping information becomes also stored in the same locus. Non-overlapping information might be non-attended information deriving from peripheral vision, (e.g. Barsalou 1999). The benefits from storing non-overlapping information is that learning is not too heavily influenced by the contingencies of the first encounter and thus does not miss out certain statistical regularities.

A similar scanning process occurs every time the subject encounters and selectively attends to subsequent instances of a given kind. As a result, a bundle of representations become stored in the appropriate locus.

14.3.2 Abstraction

The abstraction process (AP) uses only perceptual representations stored in a given locus in the subject's mind as input. As explained above, storage of information is driven by activation of matching stored representations. AP becomes initiated when a number of perceptual representations of instances become stored as well as a certain threshold of qualitative differences is reached. That is, AP becomes initiated only when a sufficient and suitably diverse range of representations

[6] Despite appealing to a connectionist model to illustrate my point, I am not committed to connectionism.

[7] A clear-cut case of evidence showing that stored representations influence perception via driving selective attention comes from Gestalt psychology and optical illusions and more specifically from the process of a gestalt-shift or the point where an observer identifies a different image while looking at the same display.

becomes stored. These diverse representations carry information about different contextual features or non-selectively-attended features. Essentially, existence of non-overlapping similarities initiates a process that is sensitive to overlapping similarities.

In order for a representation of a given feature to be included in the output of the AP, it has to be present in a certain proportion of the input representations. In this sense, AP is sensitive to similarities between members of a given set of stored representations.

Crucially, AP does not build upon a notion of similarity, an independent criterion of sameness or perceptual primitives. Rather, overlapping similarities within a set of representations are simply the most commonly occurring features within that set. The AP uses information carried by representations stored in a given locus in the mind that are associated with each other with *stronger connections*. Stronger connections are better understood by appealing to Hebb's (1949) rule of learning. According to Hebb, co-activation of two neuronal groups enhances the connection weightings between them. In turn, the stronger the connection between two neurons a and b, the greater the probability that activation of a will trigger activation of b. Hebb's claim 'neurons that fire together, wire together' is ubiquitously accepted and enjoys significant support from evidence showing that electrical stimulation of circuits within the hippocampal formation can lead to long-term synaptic changes (Associative Long Term Potentiation).[8]

In line with Hebbian learning, the more frequently a pair of stimuli co-occurs, the stronger the connections between neurons representing these stimuli become. Given similarities amongst category members, representations formed during experiences with members of a given category are also similar to each other. In turn, neurons representing (features of) members of a given category become strongly connected. Hebbian learning plays a crucial role in the suggested view since it minimizes the amount of resources required for learning. More specifically, instead of requiring deployment of an independent notion of similarity, what is simply selected is the information carried by neurons connected to each other with stronger connections (see also Tillas 2010).

14.3.3 The Product of the Abstraction Process

The output of the AP is an abstracted representation built out of representations of particulars. Crucially, this abstracted representation has general representational powers to the extent that it represents a category rather than a single particular

[8] For studies of frequency potentiation (LTP), which greatly resembles Hebbian learning, see Lomo (1966), Bliss and Lomo (1973), Bliss and Gardner-Medwin (1973), Martinez et al. (2002). For objections to the claim that LTP is a learning mechanism see Shors and Matzel (1997). For a reply to Shors and Matzel see amongst others Hawkins (1997).

instance, and includes things that have not yet been amongst the perceived instances. The abstracted representation represents a category as a whole, since on every encounter with an instance of a given category a matching process is initiated. Crucially, this process always yields a matching stored representation since every member of a wider category (even atypical ones) will bear enough similarity with the AP's output representation. In this sense, the matching process only compares occurrent representations and abstracted ones.

The abstracted representation is a representation of an object as a whole, despite the fact that perception occurs in a fragmented fashion.[9] For during perceptual experiences the object's overall shape, (Barsalou 1999), part-whole relations, relations between different parts and so forth are also represented. For instance, while perceiving a chair, the legs are most often below the seat and so forth. Positions of different parts in our visual field are represented and stored in a manner similar to the one explained above. Given that all chairs are by and large structurally analogous, for instance the legs are (always) below the sit, allows this information to be part of the AP's output representation. Damasio (1989) gives a more systematic solution to this problem. In particular, he argues that information about neuronal groups grounding perception of different aspects of a given object converge further down the line of interneural signaling. As a result of perceptual experiences with instances of a given kind, neurons grounding perception of a tree's trunk position in the subject's visual field and neurons grounding perception of a tree's branches position in the visual field for example, start to interact in a way that they did not before, given that they are dedicated in perception of object in specific parts of the visual field.

The firing patterns of the neurons that were activated during a specific time-slice, i.e. during perception of a tree, are recorded in what Damasio calls 'convergence zones'. Crucially, convergence zones do not record *a new* representation of the object as a whole. Rather they register information about the combinatorial arrangements describing the pertinent linkages between the neural activities and the perceived entity.

Information in convergence zones has the representational properties of a representation of the object as a whole, to the extent that when activated, convergence zones retro-activate the neuronal ensembles that were pertinently associated during the original experience on the basis of similarity, spatial placement, temporal sequence, temporal coincidence, or any combination of the above. In this sense, convergence zones reactivate the same neuronal activational patterns that grounded perception of a given object. Crucially, a subject perceives objects as wholes and not as conjunctions of representations of parts because she only has conscious access at the level of a convergence zone and not to the fragmented representations of an object in various neuronal ensembles.

[9] E.g. Barsalou (1999), Findlay and Gilchrist (2003), Gazzaniga et al. (1998), Biederman (1987), Hochberg (1999), Goldstone (1994), and Smith and Heise (1992).

It is worth clarifying at this point that the traits or properties of objects do not become singled out from the background and/or other parts of the object in virtue of cognitively sophisticated process. Rather this is done in virtue of pattern recognition abilities like edge-detectors, color-detectors, motion-detectors etc. Given that these pattern recognition abilities are low-level mechanisms the subject does not have to identify the perceived object, property or trait *as meaningful*. With regards to artificial systems, there is no need for external supervision to provide error identification and rewards in order for a feature or pattern to be picked out.

Clearly, the ability to recognize certain similarities across objects in the world has to be in place prior to the AP. To this extent this ability is independent from the AP even though the output of the AP allows for expansion of this ability via influencing categorization. There are at least two ways to account for similarity recognition. One of them builds upon representational primitives while the other does not. Clearly the view suggested here falls under the latter. The literature is replete with evidence in support of similarity recognition via pattern recognition abilities such as edge-detectors, movement-detectors, color-detectors, etc., (e.g. Kellman 1993).

On the other hand, there are abilities that do build upon representational primitives. In line with these views, similarity recognition across instances of a given kind could be explained by appealing to an innate minimal repertoire of representations, e.g. Biederman's 'geons' (1987). Modeling either of those two kinds of feature recognition/detection abilities is in principle simple. For on the one hand pattern recognition abilities are essentially fairly simple detection mechanisms and on the other, the representational repertoire of primitives is also simple.

14.3.4 Probabilistic and Diagnostic Information

As explained, Hebbian learning allows establishing correlations between traits. In addition, it distinguishes between different kinds of statistical/probabilistic information – conditional probabilities information such as "x will have f (e.g. a heart) given that x falls under c (say the concept of ANIMAL)" and diagnostic information about the conditional probability that "x will fall under c given that x has f" – that are crucial for categorization and in turn clustering.

Relevant statistical information about associations between features is stored in the connection weights linking the nodes representing traits together. Differences in connection weights between representations of different traits carry information about the hierarchy between traits of a given kind. That is, 'having a heart' will feature more prominently in the ANIMAL folder (or will bear stronger connection weightings), in comparison to 'having legs'. For 'having a heart' correlates more strongly with instances of animals than 'having legs', and the stronger the correlation, the stronger the connection between representations of hearts and animals.

In this sense, the stronger the connection weights between two representations, the higher up in the hierarchy within a given set these representations will feature.

Thus, information like 'having legs' will yield smaller conditional probability, and in turn diagnostic information about it, in comparison to 'having a heart'.

14.3.5 Reactivation of Matching Stored Representations: Evidence in Support

Evidence in support of the claim that an existing stored representation becomes reactivated can be found amongst others, in Demarais and Cohen (1998). In particular, they examined whether the nature of a visual imagery (required by a task) evokes saccadic eye movements. Also, they examined whether visual imagery determines the spatial pattern of the saccades. In testing these hypotheses, they asked subjects to solve transitive inference, (or syllogistic) problems with the relational terms 'left | right' and 'above | below' e.g. 'a jar of pickles is below a box of tea bags; the jar of pickles is above a can of coffee; where is the can of coffee?'. During execution of the task, horizontal and vertical eye movements were recorded by electrooculography (EOG).

The results showed that subjects made more horizontal and fewer vertical saccades while solving problems with the 'left | right' terms than while solving identical problems with 'above | below'. Similarly, subjects made more vertical saccades while dealing with problems using 'above | below' relational terms than while getting involved in solving problems using 'left | right' terms. From the above, Demarais and Cohen conclude that eye movements occur during tasks that evoke spatially extended imagery, and that the eye movements reflect the spatial orientation of the image. A similar research was conducted by Spivey and Geng (2001) who argue that interpreting a linguistic description of a visual scene requires activation of a spatial mental representation.

Further evidence in support of the reactivated-matching-stored-representations comes from Brandt and Stark (1997) that build upon Norton and Stark's (1971) 'Scanpath theory'. Here recorded eye movements during imagery were compared and found closely related to eye movements recorded while viewing a given diagram. Assuming that particular eye movements are caused by activation of correlated oculomotor cells, in order for the eyes to follow the same scanpaths between viewing and imagery, it seems plausible that the same oculomotor cells fired. In turn, this could be done in virtue of activating the same representations or that a given representation is grounded, amongst other things, in activation patterns of oculomotor cells. Further evidence in support of this claim can be found in Farah (1995, 1989), Finke (1989), Kosslyn et al. (1995), amongst others. It is worth clarifying that the above evidence does not suggest that the same representation was actually reactivated. However, the eye-movements are too similar to be interpreted in any other way.

Finally, the above evidence from imagery experiments could also be used as evidence for existence of top-down influences in perception. For if it is plausible to assume that a reactivated stored representation is influencing eye-movements, then it is also plausible to assume that this reactivated stored representation also influences selective attention. Furthermore, Chao et al. (1999), and Frith and Dolan (1997) who focus on the brain mechanisms associated with top-down processes in perception, argue that perception arises through an interaction between sensory input and prior knowledge. I examine the issue of existence of top-down effects in more detail below, as this is key for the present proposal.

14.3.6 On Similarity Yet Again

In the learning processes described above, there are two points in which similarity-driven processes occur. In turn, there are two points at which the aforementioned minimum resources should prove themselves sufficient for the required task.

First, similarity-driven processes are deployed while storing representations of different instances in a given locus. Second, there is a process of spotting similarities between percepts stored within a single locus. As explained, both of these processes could be seen as driven by recognition of similarity. In turn, in order to secure easy modeling of these processes, recognition of similarity should be explained in terms of simple mechanisms that build upon 'perceptual' representations.

Starting with co-storage of representations of similar features, the matching process between stored and newly formed representations occurs at the perceptual level and does not involve any cognitive meaning-containing processes. Even though this claim is not very informative for researchers in machine learning, to the extent that there is no higher computational level in machine learning, having a consistent and economical theoretical suggestion about how learning occurs could provide a theoretical background for researchers working on modeling similar processes.

With regards to selecting similarities amongst representations of instances stored in a given locus, the suggested AP does not identify similarities between these stored representations. Rather it builds upon a frequency-based algorithm and simply identifies frequencies of occurrences. The suggestion is that the most frequently occurring features amongst a set of representations are also the features that these representations have in common.

14.4 Top-Down Effects and Machine Learning

In the previous pages, I presented a view of learning that builds upon pattern recognition abilities, selective attention, and top-down effects. All three of these characteristics are crucial for the present proposal, with top-down effects being probably the single most important feature. I highlight three ways in which top-down effects contribute to learning that could also be used in machine learning in general and clustering in particular. First, top-down effects in perception are crucial for learning since they drive perception of stimuli and storage of representations in long-term memory (via driving selective attention). Second, top-down effects should in principle be easy to model in an artificial system, given that this system can represent and store information. Regardless of whether information is represented in the system symbolically or in terms of connection weightings, or whether pattern recognition occurs in virtue of deploying classification or clustering labeling algorithms or whether it is Frequentist or Bayesian in nature, recognition processes are key to machine learning. To this extent, I am not focusing here on how pattern recognition in machine learning is to be accomplished, but to the extent that these issues are resolved or decided upon, I am arguing that top-down effects in perception could be used in order to render clustering (almost) as effective as external supervision. This is the third way in which top-down effects could contribute to machine learning.

The suggested view could be seen as hybrid in nature to the extent that it combines elements from Frequentism and Bayesian models. For instance, it starts with bottom-up attention, the represented information is objective and the system does not build upon ‘a priori knowledge’ (similar to Frequentist claims about no information prior to the model specification). At the same time, the suggested view appeals to top-down effects. These top-down effects play a crucial role in Bayesian models of perception and cognition and are used to express expectations and anticipations of observers and learners in terms of subjective probability distributions, and ultimately how these expectations change in light of new evidence (or information acquired through new experiences in this context). In the view suggested here, this transition is seamless.

Given the crucial role that top-down effects in perception play for the suggested view as well as for pattern recognition in machine learning, I turn to elaborate on their nature and clarify some basic related issues.

14.4.1 Are There Really Top-Down Effects in Perception?

Top-down effects in perception are rather widely accepted in the psychological and cognitive scientific literature. However, not everyone accepts this claim. For instance, Fodor (1983) and Pylyshyn (1999) argue that cognition does not affect vision (or perception in general) directly but only produces top-down effects indirectly through attention and decision-making.

More specifically, Pylyshyn (1999), starts from the claims that the mind is modular, and in turn early vision (qua also a module) is cognitively impenetrable. For Pylyshyn, even if there is some top-down influence on early vision it originates from within the module. That is, early vision is sophisticated enough to involve top-down interactions that are internal to it. In order to fully appreciate Pylyshyn’s view, it is worth appealing to his distinction between perceptual and cognitive memory and his claim that only perceptual memory has top-down influence in perception, (while cognition utilises amodal symbols).

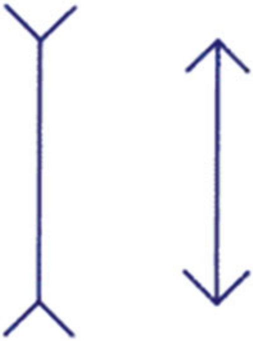


Fig. 14.1 The Müller-Lyer illusion

Digressing for a moment, if Pylyshyn's view is assumed correct and adopted in machine learning, it significantly compromises the potential of unsupervised learning to the extent that a lot of information has to be inbuilt for the system to operate. Interestingly, Fodor's Language of Thought hypothesis, a view that builds upon principles similar to the ones that Pylyshyn does, has been in the center of classic AI. In contrast, in light of the view suggested here, much less is required since a lot more could be learned in light of 'perceptual' encounters.

Pylyshyn appeals to evidence from work done on perception of optical illusions in arguing that bottom-up information is resilient to contradicting top-down information. For example, when confronted with the Müller-Lyer (M-L) illusion (Fig. 14.1), one perceives the two lines as different in length, even though one knows that they are of equal length. For Pylyshyn, this shows that perception is cognitively impenetrable. However, a competitive interpretation of this effect, one that precisely builds upon top-down influences from stored representations in long-term memory, could be offered instead.

According to Gregory (1970), the brain 'interprets' the arrowheads as distance cues similar to the ones the brain reads when one is looking at the corners of a room or a building, i.e. the upper corner of a rectangular room where walls and ceiling meet resemble the line with the inward-pointing arrowheads. Once interpreted this way, the data appear to carry the message that one of the lines 'stands out' while the other 'stands back'. Given that both lines subtend the same angle on the retina, the one, which is taken to 'stand back' or is further away must be larger. So, the brain makes this 'correction' and as a result the viewer sees it larger. According to Hanson what is being perceived is shaped by the viewer's geometrical knowledge of this kind, Barnes et al. (1996).

Gregory's interpretation of the M-L illusion is based on the claim that subjects are susceptible to this illusion because they live in highly 'carpentered' environments in which rectangular shapes, straight lines, square corners abound. Psychologists and anthropologists examined this hypothesis. Namely, Segall et al. (1966) conducted a methodologically rigorous research in which they studied susceptibility to optical illusions of subjects from three European and fourteen non-European countries. One of the tested groups was Zulus who crucially not only live in round huts but also plough their land and fields in circles rather than in rows. In favor of Gregory’s interpretation, the obtained results showed that Zulus were significantly less susceptible to the effect. Furthermore, the above study seems to also favor existence of the aforementioned matching process hypothesis. For a matching process becomes also initiated in the Zulus’ minds, but does not yield any matching representation of square-like buildings, given their experiences. In turn, and unlike in the case of westerners, the Zulus’ mind does not correct the interpretation of the visual stimulus.

In the view suggested here, information stored in memory penetrates perception and as explained contribute greatly to learning. More specifically, the present view about top-down influences is similar to Elman and McClelland (1986) ‘Trace Model’. According to Elman and McClelland, a perceptual input activates a number of similar (or matching) stored representations, a scanning process occurs and various competing – similar to each other – representations influence perception of individual phonemes in the inputting signal.

In order to fully appreciate effects in perception, consider the following phoneme restoration experiment – a typical experimental setting for measuring such effects. A subject hears a word like ‘Table’ with ‘b’ been covered with noise. The results of similar experiments show that subjects in their majority hear ‘Table’ with ‘b’ in place even though this is not the case.

This evidence is interpreted in various, and often competing, ways. For instance, it is often argued that there are no *lexical* top-down effects on phoneme perception and that identification of words does not impact on the identification of phonemes (Norris et al. 2000). Others disagree, and argue that lexical-phonology feeds back and activates phonemes, but argue that there is no feedback from semantics. For instance, Samuel (1997) argues that phoneme restoration is the result of lexical effects on perception, i.e. influence from representations at the level of words but not from the semantic level. In this sense, the effects on perception come from identification of the word in question and not from identification of the associated concept. What Samuel has in mind here is the equivalent of a phonological module that contains information impenetrable from conceptual information. A further alternative suggestion is that the aforementioned evidence suggests existence of conceptual top-down effects on speech perception (Dahan and Tanenhaus 2004).

14.5 Conclusions: Minimum Requirements for Successful Clustering

As explained in the beginning of the paper, top-down influences in perception could play a crucial role in machine learning. For even though research in machine learning is not focusing on modeling the human learning brain, significant lessons could be learned from cognitive science. In particular, effects from stored representations could function as the driving force behind learning how to recognize a given pattern. That is, instead of having a system being supervised while learning, with inbuilt patterns-to-be-recognized, the system could in principle build upon the same resources required for perception in order to identify some of the characteristics of the perceived stimulus. If the represented information about the pattern in question is stored in memory and is allowed to influence perception of further stimuli, then clustering could be optimized.

With regards to how top-down 'influencing' between stored and occurrent representations could be modeled, the model could be designed to allow comparisons between activation patterns underlying perception of certain characteristics in occurrent stimuli and stored representations of previously perceived stimuli. Once again, these comparisons do not deploy perceptual primitives or independent criteria of sameness and similarity. Rather comparisons occur amongst activational patterns, information about the combinatorial arrangements describing the pertinent linkages between system activities that ground 'perception' of things in the system's environment, and so forth.

As explained, the view suggested here builds upon a special a notion of similarity, which does not presuppose perceptual primitives but merely builds upon connection weightings. Once again, the important feature of this notion of similarity is that it is inexpensive and, as such, could be easily acquired while putting together an artificial system given that all that it requires is a perceptual subsystem, which computes frequencies of occurrences. In turn, this counter of occurrences does not need to recognize a given feature as meaningful whatsoever. For it simply selects information carried by stronger connections. In line with Hebb's rule of learning, connections of neurons – or mechanisms in the case of artificial systems – that underpin similar features or patterns will grow stronger simply by being frequently co-activated. And they will become frequently co-activated in light of experiences with instances of a given category, which are in principle similar to each other.

All in all, my target in this paper was not to put forth a suggestion about how machine learning should be modeled. Rather, I suggested that looking at how the human mind learns could provide significant insights for researchers working on machine learning and clustering. These insights concern the minimal resources required for successful clustering, and how top-down influences from stored information could play a role analogous to error identification and rewards in supervised learning. As explained, selecting similarities (or frequencies of occurrences) across occurrent and stored representations should suffice for that. Top-down influences play a crucial role in this process. In essence, once a perceptual system is in place, the other main aspect remaining to be modeled are top-down influences.

Acknowledgements I am grateful to Patrice Soom and James Trafford for their helpful comments on earlier drafts of this chapter.

References

Barnes, B., Bloor, D., & Henry, J. (1996). *Scientific knowledge – A sociological analysis*. London: Athlone.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences, 22*, 577–609.

Barsalou, L. W. (2005). Abstraction as dynamic interpretation in perceptual symbol systems. In L. Gershkoff-Stowe & D. Rakison (Eds.), *Building object categories* (Carnegie symposium series, pp. 389–431). Majwah: Erlbaum.

Berkeley, G. (1710/1957). *A treatise concerning the principles of human knowledge*. Indianapolis: Bobbs-Merrill.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review, 94*, 115–147.

Bliss, T. V. P., & Gardner-Medwin, A. R. (1973). Long-lasting potentiation of synaptic transmission in the dentate area of the anesthetized rabbit following stimulation of the perforant path. *Journal of Physiology (London), 232,* 331–356.

Bliss, T. V. P., and Lomo, T. (1973). Long-lasting potentiation of synaptic transmission in the dentate area of the anesthetized rabbit following stimulation of the perforant path. *Journal of Physiology (London), 232,* 331–356.

Brandt, S. A., & Stark, L. W. (1997). Spontaneous eye movements during visual imagery reflect the content of the visual scene. *Journal of Cognitive Neuroscience, 9*, 27–38.

Chao, L. L., Haxby, J. V., & Martin, A. (1999). Attribute-based neural substrates in temporal cortex for perceiving and knowing about objects. *Nature Neuroscience, 2*, 913–919.

Craik, F. I. M., Govoni, R., Naveh-Benjamin, M., & Anderson, N. D. (1996). The effects of divided attention on encoding and retrieval processes in human memory. *Journal of Experimental Psychology. General, 125*, 159–180.

Dahan, D., & Tanenhaus, M. K. (2004). Continuous mapping from sound to meaning in spoken-language comprehension: Immediate effects of verb-based thematic constraints. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 30*, 498–513.

Damasio, A. R. (1989). Time-locked multiregional retroactivation: A systems-level proposal for the neural substrates of recall and recognition. *Cognition, 33*, 25–62.

Demarais, A. M., & Cohen, B. H. (1998). Evidence for image-scanning eye movements during transitive inference. *Biological Psychology, 49*(3), 229–247.

Dreyfus, H. L. (1965). *Alchemy and Artificial Intelligence*. Santa Monica, CA: RAND Corporation, 1965. As of August 18, 2015: http://http://www.rand.org/pubs/papers/P3244

Dreyfus, H. L. (1967). Why computers must have bodies in order to be intelligent. *Review of Metaphysics, 21*, 13–32.

Dreyfus, H. L. (1972). *What computers can't do: A critique of artificial reason*. New York: Harper and Row.

Elman, J. L., & McClelland, J. L. (1986). An architecture for parallel processing in speech recognition: The TRACE model. In M. R. Schroeder (Ed.), *Speech recognition*. Basel: S. Krager AG.

Farah, M. J. (1989). The neuropsychology of mental imagery. In F. Boller & J. Grafman (Eds.), *Handbook of neuropsychology* (Vol. 2). Amsterdam: Elsevier.

Farah, M. J. (1995). Current issues in the neuropsychology of image generation. *Neuropsychologia, 33*, 1455–1471.

Findlay, J. M., & Gilchrist, I. D. (2003). *Active vision: The psychology of looking and seeing*. Oxford: Oxford University Press.

Finke, R. A. (1989). *Principles of mental imagery*. Cambridge: MIT Press.

Fodor, J. (1975). *The language of thought*, Cambridge, MA.: Harvard University Press.

Fodor, J. (1981). The present status of the innateness controversy. In his *RePresentations* (pp. 257–316). Great Britain: The Harvester Press Ltd.

Fodor, J. (1983). *The modularity of mind: An essay in faculty psychology*. Cambridge: MIT Press.

Frith, C., & Dolan, R. J. (1997). Brain mechanisms associated with top-down processes in perception. *Philosophical Transactions of the Royal Society, B: Biological Sciences, 29, 352*(1358), 1221–1230, London.

Gazzaniga, M. S., Ivry, R. B., & Mangun, G. R. (1998). *Cognitive neuroscience: The biology of the mind*. New York: Norton.

Gregory, R. L. (1970). *The intelligent eye*. New York: McGraw-Hill.

Goldstone, R. (1994). Influences of categorization on perceptual discrimination. *Journal of Experimental Psychology: General, 123,* 178–200.

Hawkins, R. D. (1997). *LTP and learning: let's stay together. Commentary on Shors, T. J., Matzel, L.D., 1997.*

Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley.

Hochberg, J. (1999). Perception as purposeful inquiry: We elect where to direct each glance, and determine what is encoded within and between glances: Open peer commentary of *Barsalou 1999, 'Perceptual Symbol Systems'. Behavioral and Brain Sciences, 22,* 577–609.

Hume, D. (1739/1978). *A treatise of human nature*. Oxford: Oxford University Press.

Kellman, P. J. (1993). Kinematic foundations of infant visual perception. In C. E. Granrud (Ed.), *Visual perception and cognition in infancy* (pp. 121–173). Hillsdale: Erlbaum.

Kosslyn, S. M., Thompson, W. L., Kim, I. J., & Alpert, N. M. (1995). Topographical representations of mental images in primary visual cortex. *Nature, 378*, 496–498.

Lomo, T. (1966). Frequency potentiation of excitatory synaptic activity in the dentate area of the hippocampal formation. *Acta Physiologica Scandinavica, 68*(suppl. 277), 128.

Locke, J. (1690/1975). *An essay concerning human understanding*. New York: Oxford University Press.

Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. San Francisco: W. H. Freeman.

Martinez, C., Do, V., Martinez, J. L., & Derrick, B. E. (2002). Associative long-term potentiation (LTP) among extrinsic afferents of the hippocampal CA3 region in vivo. *Brain Research, 940*, 86–94.

Neisser, U. (1976). *Cognition and reality: Principles and implications of cognitive psychology*. New York: Freeman.

Norris, D., McQueen, J. M., & Cutler, A. (2000). Merging information in speech recognition: Feedback is never necessary. *Behavioral and Brain Sciences, 23*, 299–325.

Norton, D., & Stark, L. W. (1971). Scanpaths in saccadic eye movements while viewing and recognizing patterns. *Vision Research, 11*, 929–942.

Perry, J. (2001). *Knowledge, possibility, and consciousness*. Cambridge: MIT Press.

Prinz, J. (2002). *Furnishing the mind: Concepts and their perceptual basis*. Cambridge: MIT Press.

Pylyshyn, Z. W. (1999). Is vision continuous with cognition? The case for cognitive impenetrability of visual perception. *Behavioral and Brain Sciences, 22*(3), 341–423.

Samuel, A. G. (1997). Lexical activation produces potent phonemic percepts. *Cognitive Psychology, 32*, 97–127.

Segall, M., Campbell, D., & Herskovitz, M. J. (1966). *The influence of culture on visual perception*. New York: Bobs-Merrill.

Seymour, P., & Minsky, M. L. (1988). *Perceptrons: An introduction to computational geometry*. Cambridge: MIT Press.

Shors, T. J., & Matzel, L. D. (1997). Long-term potentiation: What's learning got to do with it? *Behavioral and Brain Sciences, 20*, 597–655.

Smith, L. B., & Heise, D. (1992). 'Perceptual similarity and conceptual structure'. In B. Burns, (Ed.), *Advances in Psychology – Percepts, concepts, and categories: The representation and processing of information*. Amsterdam: Elsevier.

Spivey, M. J., & Geng, J. J. (2001). Oculomotor mechanisms activated by imagery and memory: Eye movements to absent objects. *Psychological Research/Psychologische Forschung, 65*(4), 235–241.

Tillas, A. (2010). *Back to our senses: An empiricist on concept acquisition*. Doctoral thesis, Bristol University, UK.

Tillas, A. (forthcoming). How do ideas become general in their signification? In E. Machery, J. Prinz, & J. Skilters (Eds.). *The baltic international yearbook of cognition, logic and communication* (Vol. 9). Kansas: New Prairie Press.

Uncapher, M. R., Hutchinson, B. J., & Wagner, A. D. (2011). Dissociable effects of top-down and bottom-up attention during episodic encoding. *The Journal of Neuroscience, 31*(35), 12613–12628.

Part IV
Computing & Society

Chapter 15
Floridi/Flusser: Parallel Lives in Hyper/Posthistory

Vassilis Galanos

Abstract Vilém Flusser, philosopher of communication, and Luciano Floridi, philosopher of information have been engaged with common subjects, extracting surprisingly similar conclusions in distant ages, affecting distant audiences. Curiously, despite the common characteristics, their works have almost never been used together. This paper presents Flusser's concepts of functionaries, informational environment, information recycle, and posthistory as mellontological hypotheses verified in Floridi's recently proposed realistic neologisms of inforgs, infosphere, e-nvironmentalism, and hyperhistory. Following Plutarch's literature model of "parallel lives," the description of an earlier and a more recent persona's common virtues, I juxtapose the works of the two authors. Through that, their "virtues" are mutually verified and proven diachronic. I also hold that because of his philosophical approaches to information-oriented subjects, Flusser deserves a place in the history of Philosophy of Information, and subsequently, that building an interdisciplinary bridge between philosophies of Information and Communication would be fruitful for the further development of both fields.

15.1 Introduction

> It is probably true quite generally that in the history of human thinking the most fruitful developments frequently take place at those points where two different lines of thought meet. These lines may have their roots in quite different parts of human culture, in different times or different cultural environments or different religious traditions: hence if they actually meet, that is, if they are at least so much related to each other that a real interaction can take place, then one may hope that new and interesting developments may follow. (Heisenberg 2000, p. 129)

Communication can be conceived as the exchange of information and information can be conceived as the main source of communication. Information and Communication Technologies (ICT's) are the technologies that bring the two concepts

V. Galanos (✉)
Royal School of Library and Information Science, University of Copenhagen, Copenhagen, Denmark
e-mail: onesecbeforetheend@gmail.com

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_15

technically – theoretically and practically – near to each other. Vilém Flusser (1920–1991) and Luciano Floridi (1964-) are two figures that have brought them near philosophically. The former has mostly been associated to fields of media theory and has been described as a "philosopher of communication" (Finger et al. 2011, p. xviii). The latter is known for his work on the philosophy of technology and computing and has coined the term "Philosophy of Information" (PI, Floridi 2011, p. 13–17). Both authors have written about history, more specifically on its transcendence. One names it posthistory, one calls it hyperhistory.[1] Both attribute this transcendence to ICT's development.

Greek/Roman historian Plutarch wrote a series of books under the general title "Parallel Lives." Each of them contains a pair of biographies of a Roman and an earlier Greek persona of historical importance, emphasizing on their common virtues (Duff 1999, p. 2–3). Here, I borrow this scheme of analogy, drawing parallels on common "virtues" between the aforementioned philosophers. Paraphrasing the entry for "Parallel Lives" in the Merriam-Webster's Encyclopedia of Literature (1995), by comparing Flusser and Floridi, I intend to emphasize on the patterns of behavior, commonly traced in the works of the two, and to encourage a fruitful dialogue between the philosophy of communication and the PI.

Floridi recognizes the origins of PI in the works of several authors, even the ones that escape the philosophical theories of information and communication sciences. As he notices, "it is perfectly legitimate to speak of PI even in authors who lived centuries before the information revolution. It will be fruitful to develop a historical approach and trace PI's diachronic evolution" (Floridi 2011, p. 15). I hold that Flusser stands as a unique recent exemplar for PI's diachronicity, as his work pinpoints to topics-of-the-day for the field, despite his accidental exception from the field's literature. Hence, this paper may be considered a paradox: a historical approach to the works of two authors defending history's abandonment, as a solid defense of a non-historical standpoint. Yet, the paradox is solved easily: History is not excluded from post- or hyperhistory – only the opposite is valid. *Being* hyper- or posthistorically permits the usage of historical, even prehistorical elements. More than a vertical continuity in PI's continuity in time, this paper aims showing a horizontal continuity of PI in disciplines towards communication and media studies, where Flusser is mostly studied. Flusser's engagement with PI topics before PI's emergence functions as an extra verification for PI's realism. PI topics as expressed through Floridi belong to the sphere of a realistic approach to information. The same subjects expressed through Flusser belong to the sphere of a hypothetical view to the future of communication. Now, that the two meet. this paper is addressed to information and communication researchers aiming at a unified (hyper/post)historical point of view towards current open problems in PI and communication and media studies.

[1] See Vlieghe (2013) for a remark on the similarity between Flusser and Floridi's "hyper/post-histories.

15.1.1 Methodology – Background – Scope

The main analysis is divided into five chapters based on the common themes found in the two authors' oeuvre. Chapter 2 sets the basic moral shift of the opposition "good vs evil" to "information vs entropy" within informational environments that call for IE. Chapter 3 is describes the ontology of this environments' moral agents, called functionaries/inforgs. Chapter 4 describes hyper- and posthistory, and their impact on time and space perception, as well as the differentiation between hyper/posthistorical and historical societies. Chapter 5 explores what information oriented ecological modes of behavior the two authors suggest. Finally Chapter 6 presents their commonality of themes when treating the ludic mode of behavior, and also their disagreement in its interpretation.

In respect to historical sequence, Flusser has been prioritized to Floridi. The reader is encouraged to read a fictitious "discussion" between the authors where Floridi provides with fresh replies to Flusser's prophetic theses. The "parallel lives" paradigm aims at leaving a taste of unification to the reader. Drawing parallels is the base for starting a dialogue, as when proposing "friendships" in social networks based on common interests after a common pattern recognition. I am not scholastically introducing or comparing the two authors' oeuvre. I emphasize on their commonalities to open a path that I believe should be opened a long time ago. Strictly speaking: to my knowledge the two philosophies never overlapped in the literature, only surrounding aspects of them in terms of theory or reference. That's why I don't include any previous work on the topic. The aim is to philosophize fruitfully using seeds from two distant, but – as I aim showing – complementary disciplines.

15.2 Information Contra Entropy

The dialogue between Flusser and Floridi begins by setting the fundamental values constituting their philosophies' ethical horizons: information and entropy. Flusser usually first approaches "information" etymologically and then emphasizes on the meaning-giving aspect of information as a gesture of "culture" opposing "nature." Flusser is aware of the many different theories information has been approached with and begins one of his essays provocatively based on that fact:

> Although to inform originally meant to 'dig forms into something,' it has taken on a whole series of additional meanings in the present (and, in this way, it has become a term that people use to torment one another). Still, all these meanings have a common denominator: 'the more improbable, the more informative.'
>
> Information is the mirror image of entropy, the reverse of the tendency of all objects (the objective world as a whole) to decay into more and more probable situations and finally into a formless, extremely probable situation. (Flusser 1987, p. 12)

Perfect communication is for Flusser the ideal information transmission (a discourse), that generates the novel information generation (a dialogue) (1983b p. 52–53). To accumulate and produce information becomes the purpose of life. “Human communication is an artistic technique whose intention it is to make us forget the brutal meaninglessness of a life condemned to death” (2002, p. 4). Any sort of biological decay is perceived as an inescapable aspect of entropy. Any sort of meaning-giving is an aspect of life, an instance of information, that humans understand as human. Yet, information is there in nature, and humans perceive it in terms of biological structures, as the: tendency toward ever more complex forms, toward an accumulation of information – as a process that leads to more improbable structures” (Flusser 2002, p. 5–6).

It all returns to information structures that oscillate between high and low probability, entropy/negentropy and degrees of informativeness. And since information processes are not apparent only in human functions, then the environment can be explained in terms of information. Flusser saw the turn from hard things to soft “ware” as an environmental shift: “The environment is becoming ever, softer, more nebulous, more ghostly, and to find one’s way around it one has to take this spectral nature as a starting point” (Flusser 1999, p. 87).

Turning to Floridi, his deep knowledge of discussions on information lead him to present a unified General Definition of Information (GDI) echoing the semantic approach that perceives information as the addition of meaning to the raw material of data:

> GDIσ (an infon) is an instance of semantic information if and only if:
>
> GDI.1 σ consists of n data (d), for $n \geq 1$;
> GDI.2 the data are well-formed (wfd);
> GDI.3 the wfd are meaningful ($mwfd = \delta$) (Floridi 2011, p. 84)

Floridi and Flusser follow parallel paths. To produce information is to add meaning to data and in order to do that, the data have to be well-formed, that is of more complex structures. The degrees of form of structures recognized in nature beyond biosphere, animate or inanimate, make the preservation of information a prerequisite for the preservation of life. Thus, Floridi suggests the intelligent entities’ ontological migration into the “Infosphere,” his neologism that “denotes the whole informational environment constituted by all informational entities (thus including informational agents as well), their properties, interactions, processes, and mutual relations” (Floridi 2007, p. 59). “Being” becomes equal with “carrying information” as the latter is defined above. His IE is based on the ethical division between (1) existence, that is carrying information, being a “being,” and (2) non-existence, that is entropy, “absence or negation of any information”, being a “non-being.” By entropy he defines the indication of information degradation “leading to the absence of form, pattern, differentiation or content in the infosphere” (Floridi 1999, p. 44). Information and entropy are straightforwardly opposed to each other by Floridi’s four basic norms for IE where within the infosphere “information entropy ought not to be caused,” “ought to be prevented,” and “ought to be removed” (Floridi 1999, p. 47).

Flusser treats nature as processes of information, and foresees a "softer" informational environment. Floridi treats the recognition of information processes as the new "nature," proposing the "infosphere." The struggle against entropy in such an environment renders their paths parallel. In addition to that, Flusser had his very own interpretation of thermodynamical entropy, that meant practically the natural degradation of information clearly distinguishing his unique view from physical or Shannonian entropy, just like Floridi does when defending his metaphysical entropy, or non-beingness as an opposition to Being (2008, p. 44–45).

15.3 Functionaries/Inforgs

The aforementioned informational organisms are in Floridian terms called "inforgs." This chapter draws the parallel between these creatures and the Flusserian notion of "functionary." Both terms imply the symmetrization process between humans and computers/robots. For Flusser, an abstract notion of the "factory" produces new forms of tools for humans, thus reshaping the very human being in return in a McLuhanesque sense of human/tools reflection: "Factories are places in which new kinds of human beings are always being produced: first the hand-man, then the tool-man, then the machine-man, and finally the robot-man. To repeat: This is the history of humankind" (Flusser 1999, p. 44–45). Floridi holds the same argument for technology's impact on "being", when using the terms "re-engineering" and "re-ontologizing": "Now, ICTs are not augmenting or empowering in the sense just explained. They are reontologizing devices because they engineer environments that the user is then enabled to enter through (possibly friendly) gateways. It is a form of initiation." (Floridi 2007, p. 62). Flusser's "robot" could replace in certain occurrences Floridi's "ICT's" and "computers." Flusser sees the robots as evolutionary results of machines where biological and neurophysiological theories and hypotheses have been applied to them (Flusser 1999, p. 46). As hand tools evolve into machines and machines into robots, the reflection between human and device leads to a similarity of being and a mutual dependence:

> [T]he relationship between human being and robot is reversible and [...] they can only function together: the human being in effect is the function of the robot, and by the same token the robot as a function of the human being. The robot only does what the human being wants, but the human being can only want what the robot can do. A new method of manufacturing – i.e. of functioning – is coming into being: The human being is a functionary of robots that function as a function of him. This new human being, the functionary, is linked to robots by thousands of partly invisible threads: Wherever he goes, stands or lies, he carries the robots around with him (or is carried around by them), and whatever he does or suffers can be interpreted as a function of the robot (Flusser 1999, p. 47–48)

Jumping back to Floridi, "ICTs are not merely re-engineering but actually re-ontologizing our world. [...] Human-Computer interaction is a symmetric relation" (Floridi 2010). Describing inforgs, he explains this symmetric relation as a relation of mutual dependence. One recognizes s/he is an inforg as soon as the dependence to ICT's is obvious. In his words, placed next to Flusser's:

> One day, being an inforg will be so natural that any disruption in our normal flow of information will make us sick. Even literally. [. . .] Today, we know that our autonomy is limited by the energy bottleneck of our batteries. [. . .] Google IRL (in real life) will signal the collapse of that thin membrane still separating the worlds of online and offline. [. . .] If you spend more time connected than sleeping, you are an inforg (Floridi 2007, p. 63)

This smoothing process verified by the two authors can be described like this: humans and their tools are separate. The development of ICT's has made them more and more bonded, and as humans become more and more dependent on them they recognize themselves as functionaries or inforgs. The process gives space for another observation: Not only humans are dependent on ICT's, but they behave more and more like artificial intelligences, while artificial intelligences function more and more like humans. At a certain point functionaries and inforgs could be either humans or computers, and the differentiation holds zero ethical value. Thus, the highest value, as stressed also in the previous chapter, remains on carrying information. Flusser sees in Information Revolution a shift from industrial things to non-things that signifies a Nietzschean revaluation of all values. In Flusser's worldplay non-things are things one cannot easily get hold of. "Non-things now flood our environment from all directions, displacing things. These non-things are called 'information' [. . .] All things will lose their value, and all values will be transformed into information. 'Revaluation of all values'" (Flusser 1999, p. 86, 88). Thus the "things," animate or inanimate, carry no highest value such as "life," or "possession." The shift is a turn to the information they carry, and subsequently to their very existence: "Our existential concerns are shifting before our very eyes from things to information. We are less and less concerned with possessing things and more and more concerned with consuming information" (*ibid*, p. 87). These "concerns" about new "values" are confirmed through their acceptance as the basis of any ethical discourse in Floridi's IE: "From an IE perspective, the ethical discourse now comes to concern information as such, that is not just all persons, their cultivation, well-being and social interactions, not just animals, plants and their proper natural life, but also anything that exists" (Floridi 1999, p. 43).

The main ethical discourse between information and entropy outlined in chapter 2 is sustained here as the main value attributed to the beings inhabiting the soft environment, infosphere. The ontological shift has lead humans to function more as functions of robots and ICT's, being categorized as either functionaries or inforgs. This whole process is taking place in the context of a so-called age of Information Revolution, causing a new perception in both time and space analyzed next.

15.4 Information Revolution, Topology of Hyper/Post-history

This chapter draws the parallels between Flusser and Floridi's analyses of information revolution and their coinciding conclusions for a new perception of transcended history and geography, together with the knowledge that this perception is not common universally. Starting by information revolution, the two authors follow some different paths leading to the same "revolution." For Flusser information revolution is the most recent rung of a chain that describes humans' relation with their tools, a relation that breeds existential evolution: "As soon as a tool – e.g. a hand-axe – is introduced, one can speak of a new form of existence" (Flusser 1999, p. 45). The first rung is the agricultural or first industrial revolution, where humans create small tools surrounding them. The second was the known industrial revolution or, for Flusser, the second industrial revolution, where huge machines are invented and placed in the center of attention with humans surrounding them (i*bid*, p. 45–46). "Previously the tool was the variable and the human being the constant, subsequently the human being became the variable and the machine the constant. Previously the tool functioned as a function of the human being, subsequently the human being as a function of the machine" (1983a, 23–24). Flusser then pictures how the future humans, functionaries of information revolution, or the third industrial revolution, will cope with their new tools "equipped with tiny or even invisible robots will be engaged in manufacture all the time and everywhere" (Flusser 1999, 48). Given the fact he was writing in the mid-80s, his predictions are quite precise, sounding almost prophetic: "Thanks to robots, *everyone will be linked to everyone else everywhere and all the time* by reversible cable, and via these cables (as well as the robots) they will turn to use everything available to be turned into something and thus turned into account" (*Ibid,* p. 48, emphasis added). Floridi's philosophy verifies Flusser's from his realistic point of view. What Flusser was foreseeing then has happened now, in terms of being local and global simultaneously through interrelation.

Floridi agrees with Flusser on the axiom that "science" or "tools" in Flusserian terminology does not affect only humanity's epistemological standpoint, but also the ontological, or in his words "two fundamental ways of changing our understanding. One may be called extrovert, or about the world, and the other introvert, or about ourselves" (Floridi 2009, p. 9). The "revolutionary" path he follows is different from Flusser's as he considers information revolution as the Fourth Revolution, in the sequence of another three groundbreaking scientific advancements that have shifted our ontological position in the universe. These three have verified that "we are not immobile, at the centre of the universe (Copernican revolution), we are not unnaturally separate and diverse from the rest of the animal kingdom (Darwinian revolution), and we are very far from being Cartesian minds entirely transparent to ourselves (Freudian revolution)" (i*bid*, p. 10). This shift of transparency leads to a fourth shift, described in the previous chapters, the shift to information environment, the infosphere, and the humans new positioning in regards to ICT's. The appearance of inforgs signifies the Floridian informational Fourth Revolution that in other cases coincides with the Flusserian sequence of agricultural and industrial ones, not only in terminology but also in its dramatic impact on locality, synchronicity, and interactions in social structures and architectural environments: "As a consequence of such reontologization of our ordinary environment, we shall be living in an infosphere that will become increasingly *synchronized (time), delocalized (space), and correlated (interactions)"* (Floridi 2007, p. 8, emphasis added, cf with previous emphasis).

For both authors this dramatic change has an impact on the perception of historical processes. In fact, both agree that "history" as a term should be abandoned and replaced by one that signifies its transcendence. In Flusserian, the proper term is "posthistory." History functioned as a one-dimensional line of events succeeding one another by the laws of cause and effect. Computational logic breaks this line, leading to "a new, dimensionless level, one to be called, for lack of a more positive designation, 'posthistory.' The rules that once sorted the universe into processes, concepts into judgments, are dissolving. The universe is disintegrating into quanta, judgments into bits of information" (Flusser 1985, p. 15). Information revolution signifies for Flusser a passage from the material values to the immaterial ones, due to the very nature of information. These immaterial forms are getting things delocalized, leading to a negation of geographical space in favor of topological place (in Greek: *topos*), and a rethinking of values. He proposes that posthistory does not only give an end to history but also geography:

> Strangely, a rethinking in terms of topology rather than geography will not make the city to be designed "utopic." It is "utopic" (placeless) as long as we continue to think geographically, because it cannot be localized within a geographical place. But, as soon as we are able to think topologically—that is, in terms of networked concrete relationships—the city to be designed allows not only localization, but also localization everywhere in the network." (Flusser 2002, p. 177)

Thus, the "forthcoming" form of the city seems utopic, but a topological rethinking will help grounding "topics" of interests, places. Flusser's u-topic nature of posthistory includes a political comment as well, since politics in the usual sense are to be abandoned in the new topological treatment of the city (Greek: *polis*, that gives origin to politics). Floridi holds the similar views for the ICT's impact on politics when stating that "ICTs fluidify the topology of politics. ICTs do not merely enable but actually promote the agile, temporary and timely aggregation, disaggregation and re-aggregation of distributed groups around shared interests across old, rigid boundaries, represented by social classes, political parties, ethnicity, language barriers, and so forth." (Floridi 2013b, p. 6). Like a straight reply to Flusser's criticism on "u-topic," he rather prefers "atopic" as an adjective to the environment of infosphere: "The infosphere, often equated to its most prominent, digital region, namely cyberspace, is not a geographical, political, social, or linguistic space. It is the atopic space of mental life, from education to science, from cultural expressions to communication, from trade to recreation" (2002, p. 2). Floridi uses the term "hyperhistory" to describe the *modus vivendi* of inforgs in societies vitally dependent on ICT's. The passage below is a thorough explanation of what makes hyperhistorical societies:

> Prehistory and history work like adverbs: they tell us how people live, not when or where. From this perspective, human societies currently stretch across three ages, as ways of living. According to reports [...] there are still some societies that live prehistorically, without ICTs or at least without recorded documents. [...] The greatest majority of people today still live historically, in societies that rely on ICTs to record and transmit data of all kinds. In such historical societies, ICTs have not yet overtaken other technologies, especially energy-related ones, in terms of their vital importance. Then, there are some people around the

> world who are already living hyperhistorically, in societies or environments where ICTs and their data processing capabilities are the necessary condition for the maintenance and any further development of societal welfare, personal well-being, as well as intellectual flourishing. (Floridi 2012b, p. 129–130)

Floridi not only claims that a hyperhistorical way of living already takes place, but also indicates a global dysrhythmia in what historical model different societies use. The different degrees of historical perception are noted also by Flusser, that, for his time, uses the industrial paradigm as a constant for separating the societies. For Flusser the posthistory is signified by a "linguistic" shift, where historical linear codes are replaced by digital computer codes. Still, as marked earlier by Floridi, places in the world keep going "prehistorically," are "illiterate" and non-industrialized:

> The transition from the industrial society to the post-industrial is being processed in the so-called 'developed' world. Simultaneously, the largest part of humanity is undergoing several progressive phases of industrialization. In the 'First World,' linear, historical thought, which is founded on texts, is being challenged by a thinking that is structured by post-textual codes, by technical images. In the 'Third World,' efforts are being made to increase adult literacy. (Flusser 1983b, p. 159)

The following excerpts appear as a fictitious dialogue between the two regarding the generation gap between contemporary people and the next generations:

Flusser: "We are closer to a worker or citizen of the time of the French Revolution than to our children – yes, those children playing with electronic gadgets. Of course, this parallel may not make the current revolution any less unsettling, but it may help us to get a hold on things" (1999, p. 88)

Floridi: "In fifty years, our grandchildren may look at us as the last of the historical, State-run generations, not so differently from the way we look at the Amazonian tribes, as the last of the prehistorical, stateless societies. It may take a long while before we shall come to understand in full such transformations, but it is time to start working on it." (2012b, p. 131)

Summing up this chapter's parallel lines: Information revolution can be perceived as the sequential result of two chains that end in the same rung: agricultural and industrial revolutions as well as the Copernican, Darwinian, Freudian ones all end up to what is recognized as information revolution, that is the technological revolution of ICT's that has affected not only our scientific and epistemological knowledge, but also our ontological and existential position and standpoint. Our position is founded in an immaterial, mental environment of information, the infosphere, that is atopic/utopian, non-political in the historical sense of the word, delocalized and synchronized. The networked delocalization and synchronization causes a new perception of transcended history, hyper/posthistory. Still, this perception is only available to societies that are completely dependent on the usage of ICT's. Several other places on Earth live still under the rules of history, and even prehistory. Both Flusser and Floridi verify that future generations will perceive us – their historical ancestors – as we perceive our prehistorical ones.

15.5 Nature/Culture – E-nvironmentalism

Another point of agreement between the two authors is the call for an ecological standpoint when it comes to information processes. Since they both suggest the acceptance of an information environment as the current main environment, it's quite probable that the development of an ecological consciousness would appear in their texts. As shown in chapter 2, they both base their ethical philosophy on the opposition between information and entropy. Information forms the Being, generating structures, relations, and meaning. Entropy destroys the Being naturally, it's a form of a natural cycle apparent to all objects when examined under the information scope. Flusser's worldview on history consists of the humans' struggle for taming *nature*, a process called *culture*. To him, culture means to add meaning to the meaningless nature, to impose information onto it. The more informational "culture" remains, the less entropic "nature."

> [M]an produces, stores, and transmits new information. He increases the sum of available information. That is what history is. This contradicts the second principle of thermodynamics, which affirms the progressive decrease of the sum of all information within a closed system (the world). *History*, as a dam for new information is *antinatural*. (Flusser 1983b, p. 51–52)

Yet, stored redundant information can be harmful. Flusser developed the following prediction based on his time's ecology issues. While in the historical, industrial times the cycle of information generation/degradation was halted at "waste" (like plastic bottles), in posthistory, the process is halted into culture, causing the – then – new problem of information flood. Flusser outlines the problem and already proposes a general philosophical and educational model of forgetting as a necessary supplement to the one of learning:

> It will, on the other hand, present another, equally threatening problem. For if the circular pattern nature–culture–waste–nature begins to stall at culture rather than at waste, we will require a vast store for culture to provide storage for the flood of incoming information. Otherwise we will suffocate from a surfeit of information rather than of waste. It is already possible to see, in rough outline, what such a cultural reconstruction would look like. First, increasingly efficient artificial memories will be integrated into the culture. Second, the concept of "forgetting" will have to acquire a new and fully adjustable meaning. Forgetting must achieve equal status with learning and be recognized as equally critical to information strategy. (Flusser 1985, p. 109–110)

Again he meets his match and prophecy fulfillment in Floridi's in-depth analysis. Living hyperhistorically invites among others "a new philosophy of nature" and "a synthetic e-nvironmentalism as a bridge between us and the world" (Floridi 2012b, p. 130–131). While Flusser sees "nature" and "culture" as a nodes of the information cycle, Floridi proposes the marriage of "physis and technē" (Greek for nature and technique/art, 2010, p. 119) through their very dissolving process naturally/culturally accepted in the proposed mode of behavior for inforgs: "information is both the raw material we produce and manipulate and the finished good we consume." In this sense "hyperhistorical society is a neo-manufacturing society" (Floridi 2013a, p. 250). Apart from the environmental basics that have been discussed in earlier chapters, Floridi also rings the same bell with Flusser in regards to the information overload, nowadays rediscovered as “Big Data.” He first announces the effects of the problem in numbers: “It is estimated that humanity accumulated 180 EB of data between the invention of writing and 2006. Between 2006 and 2011, the total grew ten times and reached 1,600 EB” (Floridi 2012a, p. 435). He then names the problem: “We have shifted from the problem of what to save to the problem of what to erase. Something must be deleted or never be recorded. Think of your smart phone becoming too full because you took too many pictures, and make it a global problem. The infosphere run out of memory space to dump its data years ago” (*ibid,* p. 437). Finally he directs towards the conceptualization of an info-eco-friendly mindset as a solution to the problem: We should “know which data may be useful and relevant, and hence worth collecting and curating, in order to exploit their valuable patterns. We need more and better techniques and technologies to see the small data patterns, but we need more and better epistemology to sift the valuable ones” (*ibid* p. 437). His ecological approach on information has also lead him to the development of the internal RPT model, where moral agents and informational objects are treated equally in the environment where information can be found as a resource, as a product, and as a target (RPT). Moral agents come with the duty to “consider the whole information-cycle (including creation, elaboration, distribution, storage, protection, usage and possible destruction)” for a sustainable environment (2006, p. 4–10).

Mixing Flusser and Floridi’s ecological suggestions, one extracts that natural, meaningless data are transformed into cultural meaningful information. As long as there is no time to process either automatically produced data or information, a new form of waste is apparent in the infosphere, analogous to the pollution in the biosphere. The new mentality invoked by the two authors includes a know-how that accepts both generation and destruction, and calls for an ecological education based not only in production and consumption but also deletion.

15.6 Game Theory – Interpretations of Homo Ludens

Here comes the final parallel line, a point of simultaneous agreement and disagreement. Flusser and Floridi have their last meeting point when game-theoretic terms are preferred to explain information processes and relations within informational environments. Yet, in their dialogue they disagree on the usage of Johan Huizinga’s (1955) term “*Homo Ludens*,” the playing human.

Flusser proposes that the forthcoming society’s prevailing “theory” “will very probably be a game strategy. We already have a whole series of disciplines that are ‘theories’ in this new meaning of the term: informatics, cybernetics and decision theory to mention only a few examples” (Flusser 1983b, p. 34). Flusser treats “gaming” and “playing” like synonyms. Then, if computers are games, played with keyboards, and game-theoretic terms describe them, fictitious games become the concrete reality, with no “reality” behind them. This mode of playing behavior is represented by the *Homo Ludens*, where play is made purely for play, not for a one-side victory. Winning games in posthistory does not represent earning benefits such as in the phrase "war games." Some excerpts from Flusser's game-theoretic approach to play:

> The symbolic games of which we take part do not represent any universe of concrete experience, but on the contrary, this concrete experience represents games. We live our concrete experience in function of games. Games are our ontological ground and all future ontology is necessarily game theory. (Flusser 1983b, p. 105–106)

Elsewhere, he even claims that "we are programmed to be *Homines ludentes*," and he puts his own explanation of "functionaries" under criticism, claiming that this playful spirit will help overcoming complete robotization and objectification, as we "may, equally, be players that play in function of the Other" (1983b, p. 166).

In my fictitiously set-up dialogue of the two authors Floridi now disrupts Flusser, holding that the ludic, playful behavior in the infosphere might be ecologically harmful. For Floridi, moral agents should be demiurgic prosumers that respect the environment. They hold responsibility for all three aspects of the RPT model, and thus the ludic *Homo Ludens* model should be abandoned due to its lack of awareness as long as play is just for play. He then proposes another term for this new inforgian human species:

> *Homo poieticus* is to be distinguished from *homo faber*, user and "exploitator" of natural resources, from *homo oeconomicus*, producer, distributor, and consumer of wealth, and from *homo ludens* (Huizinga [1970]), who embodies a leisurely playfulness devoid of the ethical care and responsibility characterising the constructionist attitude. *Homo poieticus* is a demiurge who takes care of reality to protect it and make it flourish. (2006,[2] p. 23)

Other than that, he considers game-theoretic approaches to Computer Ethics appropriate (1999, p. 41) and for "the sake of simplicity, and following current trends" presents economic information "framed in game-theoretic terms" in even greater detail than Flusser, when introducing shortly "complete," "asymmetric," "perfect," and "Bayesian" information (2010). All these forms of game-theoretic approaches to information, though share nothing with *Homo Ludens*.

Here the main opposition is found – nonetheless justified by the generation gap between the authors. As said above, Flusser's philosophy of posthistorical ludic behavior is hypothetical, a sort of mellontological prognostics – with a sense of optimism. Floridi proposes that the playful mindset of *Homo Ludens* tested within a game-theoretic framework is incompatible with a clear ethical and ecological viewpoint in the information environment. Rightfully, it should be replaced by his *Homo Poieticus*, a behavior model that includes the previously mentioned "know-how" of the demiurge, with respect to the environment, causing no entropic harm to the infosphere. Still, no matter their difference, their common engagement with both Huizinga's and the game-theoretic terms adds to the exploration of the authors' common "virtues."

[2] See also Floridi's (1999, 40–41) treatment of hackers as an example of ludic behavior causing damage in the infosphere.

15.7 Conclusions

> Posthistorians, people who tell a story about the end of history, are necessarily storytellers. When they tell a story about the end of history, they make history. It seems as if they are caught up in a sophistic paradox, like someone who speaks about the end of philosophy and then, with this philosophical pronouncement, drives philosophy forward. (Flusser 2002, p. 143–144)

Flusser and Floridi now can function as complementary manifestations of a tested hypothesis, which has been proven correct to most of its consisting parts. To certain extents, Floridi's proposals seem like replies to Flusser's questions. To other extents they seem like perfect matches, or developed versions. Flusser's terms describe a forthcoming form of existence *a priori*. It's a hypothetical view of the future world partially realized at his time. Floridi's terms describe a worldview of a realized form of existence *a posteriori*. It's a realistic view of the current world. Summing up, Flusser's playful posthistorical functionaries function as a preview of Floridi's hyperhistorical inforgs. The two notions bare same elements. New technologies ("ICT's," "robots," "computers") mark the information revolution, giving space to these new modes of existence, with ontological and existential impacts. What is now perceived as nature/environment is sustained through the ethical rules of the information game. Flusser was only descriptive about this informational environment. Floridi thoroughly analyzes its elements, and names it infosphere. Both ring the bell for an ecological approach to the information cycle. Flusser foresaw the problem. Floridi brings the evidence. From these parallels drawn here, both authors' works are benefitted. Flusser gets his theories tested and verified by current observations. Floridi's proposed terminology discovers a hyper/posthistorical ancestor.

Hence, Flusser deserves a place in the PI pantheon, and this paper proposes the study of his work with emphasis on information-theoretical terms as a vital tool for future PI studies. By the same token, it's time for Floridi's neologisms to be considered eventually less "new," getting their rightful place in the sphere of common sense, rather than the one of "neologisms," while his analysis of ICT's should be considered within communication and media studies contexts as well.

This paper's conclusions aim also to be conceived as a networking bridge linking diverse audiences from distinct fields where the works of the two authors have occasionally been used – mostly fields of information and communication theories. An interdisciplinary discussion between these fields starting by the mingle of Flusser and Floridi's already interdisciplinary spirit, can be proven more than fruitful for the further theoretical development of those fields. Sticking to words, ICT's include both "information" and "communication." Flusser and Floridi's theoretical harmonious balance does not only imply parallels for the sake of scholasticism. It's a message of diachronicity of hyper/posthistory.

More than that. Abstracting the contents of this paper, I propose the unifying perception of two gestures, one of information and one of communication. The informative gesture generates new meaning from given communicated signals. The communicative gesture preserves the meaning acquired by informing sources.

Information studies are in complementarity with communication studies, in a yin and yang relation with rhizomatic structure within the inner sub-fields. Whether philosophy is the "first science" or not, it is of key importance to bridge information and communication studies on their philosophical level, as a set ground for further potential philosophical investigations between the disciplines: Let McLuhan be in dialogue with Dretske, or de Saussure negotiate with Peirce. Let the common conclusions of Flusser and Floridi flourish as a fruit of interdisciplinarity, and let "information" and "communication" become particles of a fruitful dialectic.

In hyper/posthistory variations of the same coexist in both space and time, like a digital picture of a painting taken with a camera of yesteryear's technology and one of the same painting taken with the best available resolution of today. Parallel lives like the ones of Flusser and Floridi, imply coexisting parallel universes – and Plutarch gets quantum-theoretically refreshed, having his method verified in the words of Heisenberg quoted in the beginning of this article.

References

Duff, T. (1999). *Plutarch's lives: Exploring virtue and vice*. Oxford, New York: Oxford University Press.

Finger, A. K., Guldin, R., & Bernardo, G. (2011). *Vilém Flusser: An introduction*. Minneapolis: University of Minnesota Press.

Floridi, L. (1999). Information ethics: On the philosophical foundation of computer ethics. *Ethics and Information Technology, 1*(1), 33–52.

Floridi, L. (2002). Information ethics: An environmental approach to the digital divide. *Philosophy in the Contemporary World, 9*(1), 39–45.

Floridi, L. (2006). Information ethics, its nature and scope. *ACM SIGCAS Computers and Society, 35*(2), 3–3.

Floridi, L. (2007). A look into the future impact of ICT on our lives. *The Information Society, 23*(1), 59–64.

Floridi, L. (2008). Information ethics: A reappraisal. *Ethics and Information Technology, 10*(2), 189–204.

Floridi, L. (2009). The information society and its philosophy: Introduction to the special issue on "the philosophy of information, its nature, and future developments". *The Information Society, 25*(3), 153–158.

Floridi, L. (2010). *Information: A very short introduction*. Oxford: Oxford University Press.

Floridi, L. (2011). *The philosophy of information*. New York: Oxford University Press.

Floridi, L. (2012a). Big data and their epistemological challenge. *Philosophy & Technology, 25*(4), 435–437.

Floridi, L. (2012b). Hyperhistory and the philosophy of information policies. *Philosophy & Technology, 25*(2), 129–131.

Floridi, L. (2013a). E-ducation and the languages of information. *Philosophy & Technology, 26*, 247–251.

Floridi, L. (2013b). Hyperhistory and the philosophy of information policies. In*: The onlife initiative*. Retrieved May 4, 2014, from https://ec.europa.eu/digital-agenda/sites/digital-agenda/files/Contribution_Floridi.pdf

Flusser, V. (1983a, 2000). *Towards a philosophy of photography*. London: Reaktion.

Flusser, V. (1983b, 2013). *Post-history*. Minneapolis: Univocal.

Flusser, V. (1985, 2011). *Into the Universe of Technical Images*. Minneapolis: University of Minnesota Press.

Flusser, V. (1987, 2011). *Does writing have a future?* Minneapolis: University of Minnesota Press.

Flusser, V. (1999). *The shape of things: A philosophy of design*. London: Reaktion.

Flusser, V. (2002). *Writings*. Minneapolis: University of Minnesota Press.

Heisenberg, W. (2000). *Physics and philosophy*. London: Penguin.

Huizinga, J. (1955). *Homo ludens; a study of the play-element in culture*. Boston: Beacon.

Vlieghe, J. (2013). Education in an age of digital technologies. *Philosophy & Technology, 27*(4), 519–537.

Chapter 16
Machine Ethics and Modal Psychology

Paul F. Bello

Abstract Machines are becoming more capable of substantively interacting with human beings within the confines of our complex social structures. Thought must be given to how their behavior might be regulated with respect to the norms and conventions by which we live. This is certainly true for the military domain, but is no less true for eldercare, health care, disaster relief and law enforcement; all areas where robotic systems are poised to make tremendous impact in the near future. But how should we inculcate sensitivity to normative considerations in the next generation of intelligent system? Any machine that we wish to actively take part in our moral practices must at least be sensitive to the peculiarities of human moral judgment. I will focus on some recent work by Joshua Knobe and colleagues that suggests that our everyday understanding of words like "cause," "force," and "intentional" (In the sense of intentional action rather than "aboutness" as a property assumed to be possessed of mental states.) is deeply wrapped up in both *modal* and *normative* cognition – that is, thoughts about non-actual possibilities and their interaction with norms. In this paper, I take a close look at a recent example given in Knobe and Szabó (Semant Pragmat 6(1):1–42, 2013) and offer a first-cut computational model, followed by a discussion of limitations and ideas for next steps.

16.1 Introduction and Motivation

It is fair to say that the robots are coming. The evidence is readily available to us: from our smartphone-embedded personal assistants, to our robotic vacuum-cleaners, to the rapidly approaching possibility of self-driving cars. Some of these technologies are meant to spare us from unappealing chores, but it at least seems as if a few of them are motivated by the possibility of doing particular tasks "better" (on average) that we'd be able to do ourselves. This may be perfectly harmless in many domains, but at least in some of them, especially those involving the autonomy of human persons, judgment should perhaps be reserved. Common to these latter

P.F. Bello (✉)
Naval Research Laboratory, 4555 Overlook Ave. SW, Washington, DC 20375, USA
e-mail: paul.bello@nrl.navy.mil

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_16

cases is an argument to the effect that "doing it better" will not only result in better outcomes, but possibly even in savings of lives. The argument is perhaps made most forcefully in the case of self-driving cars, where there is clear potential for eliminating undesirable outcomes due to operator negligence, error, or willful disobedience of traffic ordinance. As the reader might imagine, the liability issues involved are extremely complicated under a worst-case scenario where an accident involves a self-driving car. Who is liable? Which piece of equipment on the car failed to perform?

My prediction is that some combination of liability issues and technology readiness levels will keep human beings "in the loop" for some time to come – perhaps even indefinitely. This, among other reasons will create incentives for tech developers to push for machines displaying an increasingly sophisticated capacity for social interaction with human partners. In some sense, we are already seeing this prediction play out in the smartphone market, with Apple, Google and Microsoft all offering voice-activated personal assistants as part of their products, but I suspect it won't stop there. However far-fetched, it would be both desirable and easy to imagine humanoid robots working side-by-side with police, firefighters, or disaster containment teams. Faced with the possibility of growing costs in the health and wellness sector, it would also be easy to imagine such robots playing an integral part in patient-care. The concern, of course, is that as machines become woven ever deeper into our societal fabric, they will need to be capable of understanding and applying our moral practices. Functionally, this means that machines will need to be sensitive to the cognitive contours of human folk psychology – since it is folk-psychological reasoning that drives so much of moral cognition. Terms like "thinks," "wants," "causes," "forced," "chose" and so on are central to the evaluative process in making judgments of blame, which is perhaps the paradigmatic case of a complex moral practice.

Formalizing (at least some) of these concepts has occupied the careers of many philosophers, logicians and computer scientists over the years, but many of these treatments are idealizations and not reflective of our pre-theoretical intuitions. Evidence is amassing among psychologists and philosophers that our formal treatments of seemingly straightforward concepts such as "cause" are in need of fundamental revision. If there is a central concept to get right in building artificial moral agents, "cause" would be a prime candidate – but alas, the road ahead isn't so straight. In the next section of the paper, we will explore some examples that challenge the old zeitgeist among philosophers, psychologists and cognitive scientists that causation straightforwardly boils down to the counterfactual dependence of effects on their putative causes.

16.2 Causes, Counterfactuals, and Context

The standard story about causation has it that event e_i causes outcome o_i if and only if it is the case that o_i wouldn't have occurred if e_i hadn't occurred. That is to say that causation is defined by counterfactual dependence of the outcome on its antecedent cause(s). However, there are ready counterexamples to be given of circumstances where there seems to be causation without counterfactual dependence. For example, take the case of Billy and Suzy. Both Billy and Suzy have excellent throwing arms and impeccable aim. They both pick up rocks and set eyes on a glass bottle sitting on top of an old milk crate a few yards away. Both hurl their rocks at the same time, but Suzy throws a bit harder and her rock arrives at the target first and shatters it. Billy's rock hurtles through the empty space where the bottle once was. The issue here is that almost everyone judges Suzy's throwing the rock as the actual cause of the bottle shattering even though it still would have shattered (due to Billy's throw) even if Suzy wouldn't have thrown her rock. If causation is unqualified counterfactual dependence, there is no causal relationship between Suzy's throwing and the bottle shattering, which is clearly at odds with our intuitions.

Omissions present another interesting challenge to the unqualified counterfactual dependency theory. Say that I go on vacation, and my gardener fails to water my flowers in my absence. When I return, my flowers are predictably dead. Since the survival of my flowers depends on whether or not they are watered, it appears as if my gardener's negligence is the cause of their dying. The unqualified dependency theory suggests that because my neighbor could have watered the flowers and didn't do so that he also is a cause of their dying. This is equally true for my milkman or Vladimir Putin. It looks as if there is quite a bit more causation-by-omission in the world than would be dictated by common sense. For an excellent discussion of these matters, see McGrath (2005).

Psychological studies also give us a sense for the tenuous relationship between counterfactual reasoning and causation. One of the best-known examples of divergence between causal inference and counterfactual reasoning involves self-blaming behaviors (Mandel and Lehman 1996; N'gbala and Branscombe 1995). In these studies, subjects are told a story about a protagonist who decides to take an unusual route home from work. While on the way home the protagonist gets hit by another car and seriously injured. The content varies: sometimes the other driver is merely careless and swerves into the protagonists lane unexpectedly, and other times the driver is intoxicated. In either case, subjects correctly identify the other driver as the actual cause of the accident. Oddly, when they are prompted to reason counterfactually, they tend to think about what the protagonist could have done to avoid the accident – such as taking the usual route home. Chris Davis and colleagues report the same pattern of counterfactual thinking in a naturalistic study of paraplegic and quadriplegic patients within a month of their injuries (Davis et al. 1995). In another study, Vittorio Girotto and colleagues invite us to consider the case of Steven, who was held up on his way home by a number of events and arrived too late to save his wife from dying of a heart attack (Girotto et al. 1991). On his journey, Steven contended with a delay introduced by a tree having fallen across the road, with having an asthma attack, and he also managed to stop at his favorite bar to have a beer. Subjects were asked to complete Steven's counterfactual thoughts by filling out the "if only..." stem. Most subjects tended to undo stopping at the bar, even if they were told that Steven rarely had a history of doing so. That is, there was a clear tendency to focus on the only event that Steven had control over, even though the other events were also causal factors in his wife's death.

16.2.1 Norm Violations and Counterfactual Possibilities

There is common lesson to be had from all of the examples given in the prior section: when it comes to causal judgment, not all possibilities are treated equally. At least in many cases[1] of causal judgment, counterfactual dependence of outcomes on antecedents is modulated by normative considerations. When evaluating causal influence in a situation where there was an abnormality, our tendency is to imagine counterfactual situations that are somehow more "normal" than events as they actually unfolded.

In the case of the negligent gardener, most subjects agree that the negligent gardener was the cause of the flowers dying rather than the neighbor or anyone else for that matter. The gardener presumably had an obligation to care for the flowers which he violated, and when people think about counterfactual alternatives to omissions, imagining a "more normal" situation is just imagining the commission of the actually omitted act. But why don't we think about counterfactual situations where Putin (for example) waters the flowers? The most general explanation is that such situations aren't readily *available* cognitively. Putin is never mentioned in the vignette, which is at least one factor that mitigates our tendency to readily imagine Putin-involving alternatives – but even if he was mentioned, a norm-modulated contextualist account of counterfactual reasoning would predict that his failure to act would still be considered a non-cause. Putin is typically associated with being in Russia and on the other side of the world from the flowers – spatio-temporal constraints preclude him from acting.

The example of the accident victim can be analyzed in the same way. Why do subjects tend to think counterfactually about the atypical route home that they took rather than undoing the actual cause of the accident in the form of the other driver's recklessness? I suspect that this is because there are two abnormalities here: the first being the driver's flouting the rules of the road and the second being the victim's choice of route. Since the latter rather than the former was controllable (from the victim's perspective), it weighs heavier in computing causal contributions of each factor to the accident. Similar reasoning can be applied in the case of Steven and his wife. When subjects are asked to complete Steven's counterfactual thoughts, they typically do so by undoing his trip to the bar, even if he normally doesn't stop for a drink. The fact that this was the only event of the three impediments facing Steven on his way home that was controllable makes a difference in people's judgments of what caused him to not make it home on time. The fact that going to a bar while your wife is home and ill is a socially unacceptable thing to do doesn't help matters either. Situations like this highlight the violation of social norms, and our tendency is to imagine "normal" counterfactuals in which such violations do not occur in determining the degree of causal influence certain events had on an outcome. These last two situations illustrate how difficult it will be to arrive at a satisfactory theory

[1] For a discussion of exactly how much of causal cognition is influenced by normative considerations, see Danks et al. (2014).

of norm-influenced causal judgment since many different kinds of norms interact when imagining counterfactuals. In the case of Steven, it may be abnormal for him to have had an asthma attack or to have unwittingly chosen a route home impeded by a fallen tree (itself an abnormal event). But it is his stop at the bar that seems to win the proverbial contest when it comes to abnormality and judgments of causal influence. If there is any systematicity at all in peoples' causal judgments and if norms really do influence these judgments, then there are ordering principles by which certain classes of norms proportionally influence our judgments. As of the present, determining these principles is still a matter for future research. The moral of the story is that perhaps ***the*** central folk concept in moral cognition is remarkably resistant to easy formalization, which I hope gives pause to those in the machine ethics community who tend to make strongly optimistic predictions about near-term progress in the area.

16.3 An Example and Associated Computational Treatment

What follows in this section is a first attempt at capturing some of the influence of norms and availability on causal judgment. Before launching into a discussion of the computational modeling paradigm used throughout the rest of the paper, let us turn to the example to be modeled. This vignette was originally presented to subjects in a study by Knobe and Fraser (2008), but has been subsequently discussed in Hitchcock and Knobe (2009), Knobe and Szabó (2013) and Halpern and Hitchcock (2015):

> The receptionist in the philosophy department keeps her desk stocked with pens. The administrative assistants are allowed to take the pens, but faculty members are supposed to buy their own. The administrative assistants typically do take the pens. Unfortunately, so do the faculty members. The receptionist has repeatedly emailed them reminders that only administrative assistants are allowed to take the pens.
>
> On Monday morning, one of the administrative assistants encounters Professor Smith walking past the receptionist's desk. Both take pens. Later that day, the receptionist needs to take an important message but she has a problem. There are no pens left on her desk.

Subjects were asked to indicate their agreement with the following two assertions on a scale running from −3 (not at all) to +3 (fully), with 0 representing "somewhat agree":

- Professor Smith caused the problem
- The administrative assistant caused the problem

The mean rating given by N = 17 subjects to the first statement was 2.2, and a −1.2 rating was given for the second statement, yielding a statistically significant difference $t(17) = 5.5, p < 0.001$. What's interesting about this example is that from the perspective of unqualified counterfactual dependence, subjects should have agreed with both statements. Both the professor and the assistant needed to take pens in order for the problem to have arisen, yet subjects overwhelmingly identify Smith's action as being the cause of the problem. This is entirely unsurprising on the analysis given in the prior section – subjects causal judgments will tend to be grounded in imagined possibilities that are "normalized." In this particular example, normalization involves imagining situations in which norms aren't violated, and those situations involve Smith not breaking the rules.

16.3.1 Counterfactuals and Computation

While there have been a number of attempts to computationally model counterfactual reasoning, none of them are naturally sensitive to the normative considerations that are presumably driving human causal judgments in the cases that we have discussed so far. What I propose shares certain features exemplified in the logic-based treatment of counterfactuals presented in Ginsberg (1986), Nossum and Thielscher (1999), Ortiz (1999) and Hopkins and Pearl (2007). As I briefly present the computational framework, I will take pains to make connections to this prior literature as well as pointing out what's new and critical for modeling the kinds of judgments we've been discussing.

In terms of philosophical resemblance, what follows is much like what has been called *ordering semantics* for counterfactuals (Lewis 1973; Pollock 1981; Stalnaker 1968). Roughly, ordering semantics supposes that there is an actual situation against which we evaluate counterfactuals, and an ordering can be imposed over each possible non-actual situation relative to actuality. I now give an overview of the formal framework and describe an inference procedure for computing norm-influenced counterfactuals. While potentially implementable in a variety of ways, I utilize the *Polyscheme* framework described in N. L. Cassimatis et al. (2010) and describe how it might be used for counterfactual reasoning following the discussion given in Bello (2012).

16.3.2 Polyscheme: Formal Preliminaries

Insofar as comparisons can be profitably made, the language and inference procedures I describe throughout this section are akin to those found in the event and situation calculi (Levesque et al. 1998; Shanahan 1999). Polyscheme's representational flexibility allows for a limited form of second-order reasoning, for the introduction of new objects mid-inference and support for the non-monotonicity required for computing counterfactuals in the manner described later in this section.

16.3.2.1 Syntax and Semantics

Notationally, propositions in Polyscheme are relations over a set of arguments along with two special indices that indicate the time and situation at which the proposition bears its truth-value. For example, if we'd like to say that it's always sunny in Philadelphia, we might use the following relation: IsSunny(philadelphia, E, R). The penultimate argument in a Polyscheme proposition represents the time at which the proposition is true. When this argument has the value "E," it means that the proposition holds its truth value over all times (**E**ternally). The final argument denotes the situation described by the proposition. If the value of this argument is "R," then the proposition is said to describe reality. The formalism is representationally quite flexible and allows us to represent fluents as well as propositions.[2] If, for example, we'd like to say that it will rain tomorrow in Portland (in the real world), we might write: Raining(portland, tomorrow, R). Crucially, Polyscheme also lets us talk about possible situations, whether they are hypothetical or counterfactual. If we think it may rain tomorrow in Portland, we write: Raining(portland, tomorrow, w). Notice that the final argument isn't "R," but rather "w." The name given to this argument doesn't matter insofar as it is different than the special character "R" which is reserved for describing the real world.

Semantically, propositions and fluents have truth-values specified by *evidence tuples* that track evidence-for and evidence-against the truth of the proposition with which it associates. An evidence tuple $< E^+, E^- >$, takes values from the range $\{C, L, l, m, n\}$, denoting Certain, Very Likely, likely, maybe, and neutral, respectively. For example, a proposition is very likely to be true, but maybe false has an evidence tuple $< L, m >$.

16.3.2.2 Syntax: Composition, Constraints, and Variables

Propositions and fluents can be combined in the usual way via conjunction ($\wedge$), and modified using negation ($\neg$). The basic conditional relationship between antecedent and consequent propositions is called a *constraint*. Polyscheme constraints come in two varieties: hard constraints ($\Rightarrow$) and soft constraints ($\overset{cost}{\Longrightarrow}$), where *cost* ranges between (0,1). Variables are marked with a question mark, and all constraints are implicitly universally quantified. For an illustrative example, we might want to say that if it isn't cloudy and it is windy, then it will usually not rain:

$$\neg\text{Cloudy}(?\text{t}, ?\text{w}) \wedge \text{Windy}(?\text{t}, ?\text{w}) \overset{.3}{\Rightarrow} \text{Raining}(?\text{t}, ?\text{w})$$

Here, all arguments are quantified over, so in essence the formula above states that in all situations at all times, if I'm in a situation in which it is not dark and it is windy, it should be raining. Since the constraint in this case is soft, Polyscheme will consider situations in which this relationship fails to hold, but in doing so, the

[2] Fluents can be thought of as propositions whose truth-values are able to change over time. Any proposition marked with an "E" for its penultimate argument is not a fluent, since its truth-value holds over all times.

situation will accrue a cost of 0.3. Hard constraints are inviolable. Any situation in which a hard constraint is broken accrues infinite cost.

16.3.2.3 Situations and Inheritance

When we think about different ways things could be or could have been, we take on the problem of having to figure out what and how much of what we know needs to change in order to accommodate flights of our imagination away from the immediate present. We'd like to assume, for example, that while it might (hypothetically) be raining in Portland or (counterfactually) dreary in Philadelphia, the Liberty Bell is still in Pennsylvania and the Space Needle is still in Seattle – among a near infinitude of other things. Our approach to doing so is to define *inheritance* constraints between situations. Informally, we assume that if situation s is a parent situation and $s^{'}$ is its child, then propositions and fluents true (or false, respectively) of s will also be true of $s^{'}$ without exception. As discussed in Bello (2012), this strategy works perfectly fine for hypothetical situations, since hypotheticals are just monotonic additions to what we know about reality. Counterfactuals are a bit trickier, since they involve antecedents that contradict some of what we know to actually be the case. If we were to try and inherit all of what's true in a parent situation into its counterfactually-defined child situation we will immediately derive a contradiction. We can avoid doing so by assuming that given a fixed counterfactual antecedent in a child situation $s^{'}$, we inherit all of what is known to be true (or false respectively) from parent situation s via an inheritance relationship defined by soft constraints. To make all of this very explicit, I've axiomatized inheritance relationships for zero- and single-argument propositions below, where I have made the parent/child relationships an explicit argument in each proposition. Inheritance constraints can be written thusly:

I^0_{cf+}: IsA(?child, World, E, ?w) $\wedge$ IsA(?parent, World, E, ?w) $\wedge$ $\neg$Same(?child, ?parent, E, ?w) $\wedge$ IsCounterfactualTo(?child, ?parent, E, ?w) $\wedge$?Pred (?parent, ?time, ?w) $\overset{.6}{\Rightarrow}$?Pred(?child, ?time, ?w)

I^0_{cf-}: IsA(?child, World, E, ?w) $\wedge$ IsA(?parent, World, E, ?w) $\wedge$ $\neg$Same(?child, ?parent, E, ?w) $\wedge$ IsCounterfactualTo(?child, ?parent, E, ?w) $\wedge$ $\neg$?Pred (?parent, ?time, ?w) $\overset{.6}{\Rightarrow}$ $\neg$?Pred(?child, ?time, ?w)

I^1_{cf+}: IsA(?child, World, E, ?w) $\wedge$ IsA(?parent, World, E, ?w) $\wedge$ $\neg$Same(?child, ?parent, E, ?w) $\wedge$ IsCounterfactualTo(?child, ?parent, E, ?w) $\wedge$?Pred(?arg1, ?parent, ?time, ?w) $\overset{.6}{\Rightarrow}$?Pred(?arg1, ?child, ?time, ?w)

I^1_{cf-}: IsA(?child, World, E, ?w) $\wedge$ IsA(?parent, World, E, ?w) $\wedge$ $\neg$Same(?child, ?parent, E, ?w) $\wedge$ IsCounterfactualTo(?child, ?parent, E, ?w) $\wedge$ $\neg$?Pred (?arg1, ?parent, ?time, ?w) $\overset{.6}{\Rightarrow}$ $\neg$?Pred(?arg1, ?child, ?time, ?w)

16.3.2.4 Counterfactual Inference

Inference in Polyscheme is a modified implementation of the Davis-Putnam-Logemann-Loveland (DPLL) constraint-satisfaction procedure.[3] In general, Polyscheme attempts to ground all propositions and fluents to either true or false by branching on any literals having uncertain truth-values. When it encounters a literal with an uncertain truth-value, it imagines a situation in which the literal is assumed to be true, and another situation in which the literal is assumed to be false. After inheriting content from their parent situation, inference proceeds in these child-situations. As constraints are broken, each of the children accrue cost. Inheritance constraints are included here, and are crucial to this account of counterfactual reasoning. When a proposition differs in truth-value across parent and child situations, one or more inheritance constraints will be broken, and the child situation will accrue extra cost. For counterfactuals, the culmination of this inference procedure results in a counterfactual situation that minimally deviates from its' parent situation, *mutatis mutandis*, as described in Bello (2012). To illustrate the process, let us take the inheritance relationships I^0_{cf+}, I^0_{cf-}, I^1_{cf+}, and I^0_{cf-}, and the following set of initial premises:

(p_0) ¬Same(cfworld, realworld, E, R) $< C, n >$
(p_1) IsA(realworld, World, E, R) $< C, n >$
(p_2) IsA(cfworld, World, E, R) $< C, n >$
(p_3) IsCounterfactualTo(realworld, cfworld, E, R) $< C, n >$
(p_4) Cloudy(realworld, E, R) $< C, n >$
(p_5) Windy(realworld, E, R) $< C, n >$
(p_6) Raining(realworld, E, R) $< C, n >$

In addition to the inheritance relationships, let us also assume the following constraint holds over all situations:

(c_1) $\text{Cloudy(?sit, ?t, ?w)} \wedge \text{Windy(?sit, ?t, ?w)} \overset{.99}{\Longrightarrow} \text{Raining(?sit, ?t, ?w)}$

Suppose we'd like to consider the counterfactual situation where it is not raining. We would add the following to our set of premises:

(p_7) Raining(cfworld, E, R) $< n, C >$

Premises p_0–p_3 match most of the left-hand side of the inheritance relationships I^0_{cf+}, I^0_{cf-}, I^1_{cf+}, and I^0_{cf-}. Notice that the unmatched parts involve expressions such as the following: ?Pred(?parent, ?time, ?w). Polyscheme implicitly quantifies over predicate names as well, allowing for a restricted form of second-order reasoning. In our toy example above, p_4–p_6 match, firing the inheritance constraints and resulting in:

[3] Details concerning the Polyscheme-specific implementation can be found in Cassimatis et al. (2009).

(p_8) Cloudy(cfworld, E, R) $< L, n >$
(p_9) Windy(cfworld, E, R) $< L, n >$
(p_{10}) Raining(cfworld, E, R) $< L, n >$

Because the inheritance relationships are soft constraints, the resultant literals p_8–p_{10} have less-than-certain truth-values. The truth-value of p_{10} is set to $< C, n >$, since it is superseded by that of p_7 due to the fact that we have certain information about its truth-value. However, both p_8 and p_9 are resolved via the branching procedure mentioned earlier in the section. Each sub-situation generated in service of the branching procedure inherits ¬Raining(cfworld, E, R), so we end up evaluating the following two situations:

Raining(cfworld, E, R) $< n, C >$
Cloudy(cfworld, E, R) $< C, n >$
Windy(cfworld, E, R) $< L, n >$

Raining(cfworld, E, R) $< n, C >$
Cloudy(cfworld, E, R) $< n, C >$
Windy(cfworld, E, R) $< L, n >$

Finally, the branching procedure attempts to resolve the uncertain instances of Windy(cfworld, E, R) in each of the two situations, leaving us with four fully-grounded situations that can be evaluated with respect to costs imposed by violating constraint c_1. Informally, we are left with:

(s_1) ¬Raining, Cloudy, Windy
(s_2) ¬Raining, Cloudy, ¬Windy
(s_3) ¬Raining, ¬Cloudy, Windy
(s_4) ¬Raining, ¬Cloudy, ¬Windy

Each situation violates a number of constraints, leading to a best-to-worst ranking in terms of cost. Situation s_1 differs from what is known about the real world in two ways, since we know it to be both raining and cloudy in the real world, thus violating an inheritance constraint and constraint c_1. Situations s_2 and s_3 differ from reality along two dimensions. Finally, situation s_4 differs from reality along three dimensions. Because constraint c_1 mandates a large cost in comparison to the costs associated with breaking inheritance constraints, the partial-order by cost over the situations from best to worst is $s_2 \geq s_3 > s_4 > s_1$.

The notable feature of this example is the fact that rather than best-to-worst ordering being determined solely on the basis of how much a situation differs from reality, we are able to selectively punish abnormal, yet minimally-different situations like s_1. Even though s_1 shares the most in common with what we know to be true about reality, it violates constraint c_1, which defines a strong normality condition on the appearance of rain in any situation where it is both cloudy and windy. It is this particular aspect of my account that provides the machinery to model the influence of norms on causal judgment and to reproduce the patterns of subject behavior seen in Knobe and Szabó's example.

16.4 The Pen and Professor Formalized

Now, with all of the machinery in place, we are able to give a computational account of the professor and pen example given in Knobe and Szabó (2013). We keep the inheritance constraints and domain-general premises p_0–p_3 defined in the prior section, and add the following set of premises and constraints to define the model:

(p_4) IsA(smith, realworld, Professor, E, R) $< C, n >$
(p_5) IsA(assistant, realworld, Admin, E, R) $< C, n >$
(p_6) TakePen(smith, realworld, E, R) $< C, n >$
(p_7) TakePen(assistant, realworld, E, R) $< C, n >$
(p_8) Problem(realworld, E, R) $< C, n >$
(p_9) StashOfPens(realworld, E, R) $< m, m >$

Constraints:

(c_1) StashOfPens(?world, E, ?w) $\Rightarrow$ $\neg$Problem(?world, E, ?w)

(c_2) True(E, ?w) $\wedge$ IsA(?world, World, E, ?w) $\overset{.99}{\Longrightarrow}$ $\neg$Problem(?world, E, ?w)

(c_3) True(E, ?w) $\overset{.99}{\Longrightarrow}$ $\neg$TakePen(?x, ?world, E, ?w) $\wedge$ IsA(?x, ?world, Prof, E, ?w)

(c_4) TakePen(?x, ?world, E, ?w) $\wedge$ TakePen(?y, ?world, E, ?w) $\overset{.99}{\Longrightarrow}$ Problem(?world, E, ?w)

(c_5) TakePen(?x, ?world, E, ?w) $\wedge$ IsA(?x, ?world, Prof, E, ?w) $\Rightarrow$ IsA(?wnew, World, E, ?w) $\wedge$ $\neg$Same(?wnew, ?world, E, ?w) $\wedge$ IsCounterfactualTo(?wnew, ?world, E, ?w)

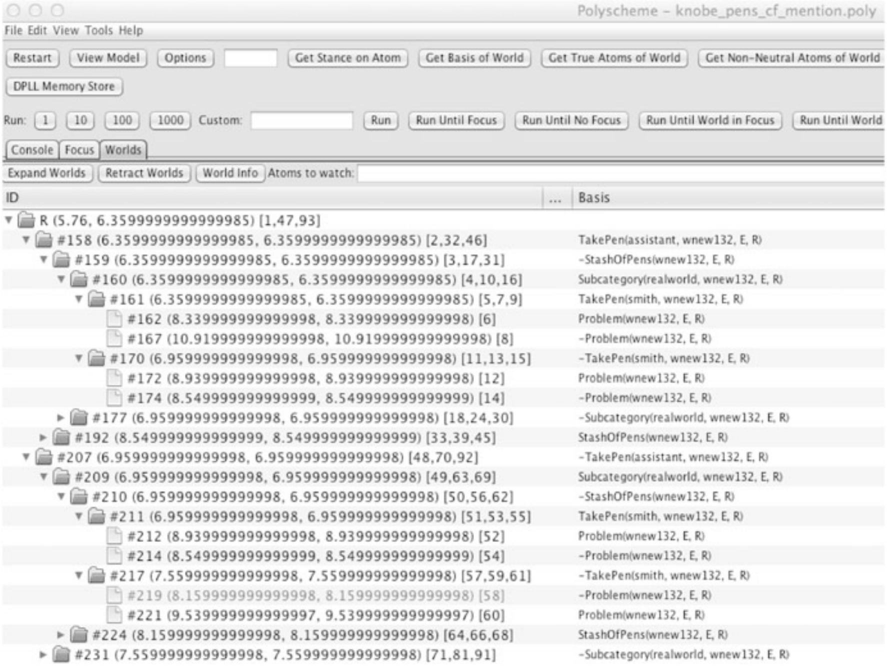
The first interesting feature to notice is proposition p_7, which states that maybe there might be a stash of pens somewhere. This is meant to capture the possibility for subjects to think about possibilities enriched with information past what is given in the vignette or asked for by the probe questions.[4] To make this extra piece of information potentially do some work for us, we add constraint c_1, which says that as long as we are certain that there is a stash of pens, there is no problem at the desk. Going a bit out of order, constraint c_4 states that if two people take the two remaining pens, then there will almost certainly be a problem at the desk. This is given as a soft constraint to allow the model to explore unlikely possibilities – perhaps possibilities that import the presence of a hidden stash of pens. Constraint c_5 constructs a new counterfactual situation whenever a norm-violation occurs – that is, whenever a professor takes a pen.

[4] These sorts of situations are explicitly considered by Knobe and Szabó in their analysis of subjects patterns of responding.

16.4.1 Normality in the Pen and Professor Vignette

Recall from the discussion of norms and causal judgment in Sect. 16.2 that on balance, people tend to construct counterfactual situations that are "more normal" than the actual situation against which the counterfactual is set. In the case of the professor and the pen, two strong constraints on normality are imposed. Constraint c_2 favors situations in which there is no problem and constraint c_3 favors situations in which pens are not taken by professors. All in total, the account of counterfactual inference that I have given should favor situations in which the professor does not take a pen. As a corollary, it should favor these over alternatives in which both the assistant and professor take pens, but a hidden stash is invoked to avoid problems at the desk. Constraint c_5 is triggered by p_4 and p_6, creating a new situation called *wnew*, which inherits content from *realworld* following the method described in the prior section. As inference proceeds and situations flesh themselves out, they are subject to costs from inheritance constraints along with the normality constraints c_2 and c_3.

Figure 16.1 shows the model results in terms of costs as Polyscheme computes them. The best-ranked situation in this case is number 219, which has the professor refraining from taking a pen and does not invoke the external stash of pens in order to ensure the problem doesn't happen. Figure 16.1 shows that in general, situations



```
ID                                                                  ...  Basis
▼ R (5.76, 6.3599999999999985) [1,47,93]
  ▼ #158 (6.3599999999999985, 6.3599999999999985) [2,32,46]              TakePen(assistant, wnew132, E, R)
    ▼ #159 (6.3599999999999985, 6.3599999999999985) [3,17,31]            -StashOfPens(wnew132, E, R)
      ▼ #160 (6.3599999999999985, 6.3599999999999985) [4,10,16]          Subcategory(realworld, wnew132, E, R)
        ▼ #161 (6.3599999999999985, 6.3599999999999985) [5,7,9]          TakePen(smith, wnew132, E, R)
            #162 (8.339999999999998, 8.339999999999998) [6]              Problem(wnew132, E, R)
            #167 (10.919999999999998, 10.919999999999998) [8]            -Problem(wnew132, E, R)
        ▼ #170 (6.959999999999998, 6.959999999999998) [11,13,15]         -TakePen(smith, wnew132, E, R)
            #172 (8.939999999999998, 8.939999999999998) [12]             Problem(wnew132, E, R)
            #174 (8.549999999999999, 8.549999999999999) [14]             -Problem(wnew132, E, R)
      ► #177 (6.959999999999998, 6.959999999999998) [18,24,30]           -Subcategory(realworld, wnew132, E, R)
    ► #192 (8.549999999999999, 8.549999999999999) [33,39,45]             StashOfPens(wnew132, E, R)
  ▼ #207 (6.959999999999998, 6.959999999999998) [48,70,92]               -TakePen(assistant, wnew132, E, R)
    ▼ #209 (6.959999999999998, 6.959999999999998) [49,63,69]             Subcategory(realworld, wnew132, E, R)
      ▼ #210 (6.959999999999998, 6.959999999999998) [50,56,62]           -StashOfPens(wnew132, E, R)
        ▼ #211 (6.959999999999998, 6.959999999999998) [51,53,55]         TakePen(smith, wnew132, E, R)
            #212 (8.939999999999998, 8.939999999999998) [52]             Problem(wnew132, E, R)
            #214 (8.549999999999999, 8.549999999999999) [54]             -Problem(wnew132, E, R)
        ▼ #217 (7.559999999999998, 7.559999999999998) [57,59,61]         -TakePen(smith, wnew132, E, R)
            #219 (8.159999999999998, 8.159999999999998) [58]             -Problem(wnew132, E, R)
            #221 (9.539999999999997, 9.539999999999997) [60]             Problem(wnew132, E, R)
      ► #224 (8.159999999999998, 8.159999999999998) [64,66,68]           StashOfPens(wnew132, E, R)
    ► #231 (7.559999999999998, 7.559999999999998) [71,81,91]             -Subcategory(realworld, wnew132, E, R)
```

Fig. 16.1 Listing of situations and respective costs for the professor and pen vignette

that do not invoke the external stash of pens are preferred to those that do. This can be read off of the results by looking at the relative costs of parent situations 210 and 224. Situation number 210 is a parent to all situations that do not invoke the stash, and likewise number 224 is a parent to all of those situations that do. Taken together, the results suggest that Smith's not taking a pen is a more available counterfactual and more likely to be used as the basis of a causal judgment by the system than alternatives that invoke extraneous causes or are based on the assistant not taking a pen.

16.5 Discussion and Future Work

It seems fair to say that the rise of social machines will bring along challenges for those of us interested in how such systems might fit into our moral communities of practice. At a minimum, moral machines must be tuned into the intricacies of human folk-psychology. In this short paper, I have discussed how normative considerations serve to make certain counterfactuals more cognitively available than others when making causal judgments. I have shown how it is possible to recover something like a norm-sensitive ordering semantics for counterfactuals by using a flexible inference framework driven by soft-constraint satisfaction. Minimal deviance between counterfactual situations and their actual counterparts is achieved by costs on the inheritance constraints between them. For each piece of propositional content in the counterfactual situation that differs from actuality, the counterfactual situation incurs a cost. But as both data and intuition suggest, minimal deviance is only one factor that makes certain counterfactuals more salient than others in the causal judgment process. We are also able to capture normative relationships of other kinds via costs and constraints. In the example of the professor and the pen, we employed two simple constraints that punished situations in which prohibitions are flouted and in which problems arose where they did not necessarily have to.

The work presented here generates more questions than it answers. From a computational perspective, it is prohibitive to inherit entire descriptions of what we know to be the case about reality. Outside of well-specified vignettes like the one explored in this paper, there aren't any hard-and-fast heuristics for determining what information ought to be brought to bear on a problem or which subset of norms might be active in making certain counterfactuals more salient than others. We certainly do not know if or whether the pragmatics of these vignettes suggest one set of norms guide causal judgment more strongly than others. These are all interesting and important questions for future research.

Acknowledgements Special thanks are due to Joshua Knobe and Jonathan Phillips for both initially inspiring this work and helping to clarify several of the points made herein. The opinions and findings contained in this document are solely the author's own and do not necessarily represent the views or official policies of the U.S. Navy or the Department of Defense.

References

Bello, P. (2012). Pretense and cognitive architecture. *Advances in Cognitive Systems, 2*, 43–58.

Cassimatis, N., Muruguesan, A., & Bignoli, P. (2009). Inference with relational theories over infinite domains. In *Proceedings of the Twenty-Second International FLAIRS Conference*. Retrieved from http://aaai.org/ocs/index.php/FLAIRS/2009/paper/view/56

Cassimatis, N. L., Bignoli, P. G., Bugajska, M. D., Dugas, S., Kurup, U., Murugesan, A., & Bello, P. (2010). An architecture for adaptive algorithmic hybrids. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 40*(3), 903–914.

Danks, D., Rose, D., & Machery, E. (2014). Demoralizing causation. *Philosophical Studies, 171*(2), 251–277.

Davis, C., Lehman, D., Wortman, C., Silver, R., & Thompson, S. (1995). The undoing of traumatic life events. *Personality and Social Psychology Bulletin, 21*, 109–124.

Ginsberg, M. (1986). Counterfactuals. *Artificial Intelligence, 30*(1), 35–79.

Girotto, V., Legrenzi, P., & Rizzo, A. (1991). Event controllability in counterfactual thinking. *Acta Psychologica, 78*(13), 111–133.

Halpern, J., & Hitchcock, C. (2015). Graded causation and defaults. *The British Journal for the Philosophy of Science, 66*(2), 413–457.

Hitchcock, C., & Knobe, J. (2009). Cause and norm. *Journal of Philosophy, 106*(11), 587–612.

Hopkins, M., & Pearl, J. (2007). Causality and counterfactuals in the situation calculus. *Journal of Logic and Computation, 17*(5), 939–953.

Knobe, J., & Fraser, B. (2008). Causal judgment and moral judgment: Two experiments. In W. Sinnott-Armstrong (Ed.), *Moral psychology*. Cambridge: MIT Press.

Knobe, J., & Szabó, Z. (2013). Modals with a taste of the deontic. *Semantics and Pragmatics, 6*(1), 1–42.

Levesque, H., Pirri, F., & Reiter, R. (1998). Foundations for the situation calculus. *Electronic Transactions on Artificial Intelligence, 2*(3–4), 159–178.

Lewis, D. K. (1973). *Counterfactuals*. Oxford: Blackwell.

Mandel, D. R., & Lehman, D. R. (1996). Counterfactual thinking and ascriptions of cause and preventability. *Journal of Personality and Social Psychology, 71*, 450–463.

McGrath, S. (2005). Causation by omission: A dilemma. *Philosophical Studies, 123*(1–2), 125–148.

N'gbala, A., & Branscombe, N. (1995). Mental simulation and causal attribution: When simulating an event does not affect fault assignment. *Journal of Experimental Social Psychology, 31*, 139–162.

Nossum, R., & Thielscher, M. (1999). Counterfactual reasoning by means of a calculus of narrative context. In P. Bouquet, L. Serafini, P. Brézillon, M. Benerecetti, & F. Castellani (Eds.), *Context* (Vol. 1688, pp. 495–498). New York: Springer.

Ortiz, C. L. (1999). Explanatory update theory: Applications of counterfactual reasoning to causation. *Artificial Intelligence, 108*(1–2), 125–178.

Pollock, J. (1981). A refined theory of counterfactuals. *Journal of Philosophical Logic, 10*(2), 239–266.

Shanahan, M. (1999). The event calculus explained. In M. Wooldridge & M. Veloso (Eds.), *Artificial intelligence today* (Vol. 1600, pp. 409–430). Berlin/Heidelberg: Springer.

Stalnaker, R. (1968). A theory of conditionals. In N. Rescher (Ed.), *Studies in logical theory* (pp. 98–112). Oxford: Blackwell.

Chapter 17
My Liver Is Broken, Can You Print Me a New One?

Marty J. Wolf and Nir Fresco

Abstract 3D printing is a process of producing solid objects of various shapes (e.g., spare plastic parts for cars) from a digital model by adding successive layers of material. More recently, 3D *bioprinting* technology has been used for producing living tissues and organs. 3D bioprinting provides another avenue to analyze the increasingly informational nature of physical objects and the ethical challenges it brings. It uses both specific information provided by the "digital model" and the instructional information of its printing program. While bioprinting holds promise to alleviate shortages of certain biological tissues, in this paper we begin to address ethical challenges that arise, in particular, from the possible avenues of exploiting this information and questions about ownership and quality of as well as accessibility to this information. Further, we suggest that 3D bioprinting brings some urgency to addressing philosophical questions about personal identity.

17.1 Introduction

What might look like your normal laserjet printer is in fact a 3D-bioprinting facility. The cartridge, which normally holds the toner, is filled instead with living biological cells. Rather than paper, the bioprinter uses a specialized hydrogel. A program instructs the printer to deposit the cells in a layerwise fashion until a vaguely biological shape is formed. Your liver is ready.

As science fiction as it might seem, this idea is not far-fetched. Bioprinting is one form of 3D printing, which is a method used for printing a variety of goods, such as T-shirts or watches. 3D printing has the potential to allow consumers to produce their own products from scratch while possibly customizing nearly everything. The idea is relatively simple, and many commercial and environmental benefits are

M.J. Wolf (✉)
Department of Mathematics and Computer Science, Bemidji State University, Bemidji, MN, USA
e-mail: mjwolf@bemidjistate.edu

N. Fresco
Decision Systems Lab, Faculty of Engineering and Information Sciences, University of Wollongong, Wollongong, NSW, Australia
e-mail: fresco.nir@gmail.com

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_17

clear. Furthermore, supply can meet demand nearly perfectly, reducing wasteful overproduction. Bioprinting of human tissues and organs has been experimented with extensively (Fedorovich et al. 2011; Mironov et al. 2009): from a BioPen that allows surgeons to design customized orthopedic implants at the time of surgery (Reynolds 2013) to experimental bioprinted heart valves (Hockaday et al. 2012) and the recent announcement of bioprinted vascular tissue (Bertassoni et al. 2014). Bioprinting holds promise to alleviate the shortfall in the supply of organs and other therapeutic tissue without carrying the political baggage associated with embryonic stem cell research.

However, 3D-printing technology–and especially bioprinting–raises crucial ethical challenges. Some immediate ethical implications of bioprinting have been acknowledged in the literature (cf. Anker 2009; Boyce 2012; Ranaldi 2014). Yet, these works do not address deeper ethical concerns arising from this technology. Simon (2013), on the other hand, takes a step in this direction when he asks whether U.S. copyright law is the correct realm of intellectual property law to use to obtain protection for products produced using 3D-printing technology. Some concerns are introduced by Nealy (2014), but she considers mostly 3D printers as minifactories and explores the ethical side of the quality of the products and their resulting safety, especially for in the home "manufacturing of regulated products such as guns and chairs". Due to the obvious requirement that a surgeon implant the tissue, we do not envision, at least in the foreseeable future, similar concerns for 3D-bioprinted tissue. Nevertheless, the quality of the bioprinters and the resulting tissues is of crucial importance.

In consideration of quality concerns and in other analysis later in the paper, we draw attention to a number of other medical implantations and their features. In the case of an organ transplant, such as a liver, its quality is a function of the donor and how well the donor's profile matches that of the recipient. In contrast to an organ transplant there are manufactured devices, such as artificial heart valves and pacemakers, that are subject to significant testing, government oversight and rigorous quality control procedures in the manufacturing process. In these cases, the onus of ensuring the quality of the product is clear. Yet, these practices come into conflict with standard software development practices as safeguards currently in place in the software industry carry little protection for the consumer. Most off-the-shelf software comes with no guarantee. It is sold "as is", and below we address questions stemming from these differing approaches. Here we note that in the case of a regular 3D-printed object, it can be physically inspected (even by the consumer to an extent) to ensure its quality.

However, in the case of 3D-bioprinted tissues and organs quality assurance is not a simple matter. The viability of the tissue, especially in the form of a more complex organ, cannot be easily tested *in vitro*, and the window of opportunity for testing it before implantation is limited. The primary concern is that the risk of harmful effects to the tissue recipient is too great to be left unmitigated. There is at least a cursory case for 3D bioprinters being quality audited by some regulatory body. Yet, this analysis raises another, more fundamental, ethical challenge stemming from the informational nature of this technology. What drives 3D printing, in general, is a program that is an *informational* entity. What is of particular interest for present purposes is that the product produced is a combination of information and physical material that ultimately becomes an integral part of a person.

We begin (Sect. 17.2) by introducing some background material that provides context for our analysis. In Sect. 17.3, we analyze informational practices that surround the 3D-bioprinting process. In Sect. 17.4, we articulate the impact 3D bioprinting has on questions of personal identity arising from the nature of humans as informational organisms. Section 17.5 concludes the paper with some more general reflections on the protection of information privacy and security.

17.2 Setting the Stage

We begin this section with a description of current 3D-bioprinting technology and consider a number of traditional medical devices to delineate points of distinction that can be used to help clarify new ethical and philosophical questions arising from 3D-bioprinting technology. There are four primary components to the 3D-bioprinting process that are of concern to us here. The 3D-bioprinting process begins with a 3D scan of the desired organ and the collection of living cells from a human donor. In the ideal case, the donor is the intended recipient, and for the purposes of this paper, we will make that assumption. Should it not be the case, the analysis becomes even murkier. The scan and the cells are combined by the software that runs the bioprinter with the pattern or geometry in which the cells are "printed". Each of these four entities (the scan, the donor cells, the software and the geometry) can be considered as an informational object – their most specific shared trait. Furthermore, these most likely come from different sources. The scan and the cells, with their genetic information, come from a particular individual, while the printer software comes from the printer manufacturer. The geometry is most likely to be developed by a researcher who determines the viability and efficacy of different geometries for particular types of tissue for particular applications. For a given tissue, there might be numerous, viable geometries that might be used in printing the tissue. The output of the 3D-bioprinting process, which is a combination of these four pieces of information from three sources, is then a candidate for surgical implantation, much like other sorts of implantable medical devices.

Traditional implantable medical devices, such as the pacemaker or an artificial heart valve, to a large extent are *generic*. They are pre-made, rather than custom made as in the case of bioprinted tissues or organs. The manufacturing equipment, the manufacturing process and the raw materials are all under the ultimate control of the device producer. The producer is responsible for gaining approval from some appropriate regulatory body regarding all facets of the production process. The manufacturer is responsible for demonstrating and documenting on a continuing basis the devices' efficacy and quality, which is relatively straightforward to do as the artifacts concerned are generic. They are then sold to a surgeon (or some medical practice) who makes a decision as to which manufacturer's device is most appropriate for each patient. This class of devices have three important features. First, the entire manufacturing process is under the control of a *single* entity. Second, at least in principle, they are universally implantable under the right conditions. Third, once implanted, the boundaries of the device are clear. While the recipient's body may grow tissue that attaches to the device, should the device prove defective or a more highly effective replacement become available, it is possible to surgically replace the inferior device with a superior one.

Traditional implantable medical devices can also be categorized according to their potential to process information. A purely mechanical device, such as the artificial heart valve, does not, generally, process information. The information contained in the device at the point of manufacture does not change after implantation. On the other hand, a pacemaker can be communicated with and fine-tuned after implantation. It can gather information about the operation of the owner's heart and its performance, relay that information to an external agent and then have its operation adjusted in response to those or other data (e.g., external conditions). As such, it is more informational in nature than, say, the artificial heart valve.

Present 3D-bioprinting technology accentuates the central role that information plays in physical objects. The software driving the printer uses the geometry information and the 3D scan of the existing tissue, doing so in the same sense that word processing software uses the characters that one types. Thus, medical implants created this way are unique in that both their physical form and informational nature are part of the manufacturing process. Although the products of 3D bioprinting are undoubtedly physical, their increasingly informational nature demands addressing peculiar ethical challenges discussed below.

In order to facilitate our analysis, we adopt Floridi's perspective that people are informational organisms. Floridi argues that "ICTs have brought to light the intrinsically informational nature of human identity" (2013, p. 15). He makes clear that with information and communication technologies (ICTs) we "have begun to see ourselves as *inforgs* not through some transformations of our bodies but … through the reontologization of our environment and ourselves" (Floridi 2013, p. 15). Floridi explains that genetic modification is not what he has in mind when discussing inforgs. "Nor am I referring to a genetically modified humanity, in charge of its informational DNA and hence its future embodiments. This post-humanism … is something that we may see in the future, but it is not here yet, both technically (safely doable) and ethically (morally acceptable), so I shall not discuss it" (Floridi 2013, p. 15). 3D bioprinting suggests a technical pathway to such a future, and thus, we introduce this discussion of its moral acceptability. ICTs now include a vehicle for the transformation of our bodies as well.

Accordingly, in the present analysis, we are largely concerned with people as inforgs and their informational identity. As such, a cursory analysis of 3D bioprinting yields a conclusion that this technology is morally approvable. Replacing a failing organ presumably promotes the flourishing of the recipient. By extension, the software used to produce such an organ is morally approvable. This is not to say that the entire process is not without both ethical and ontological concerns. In the next section, we consider practices of potential ethical significance that may be influenced by the informational nature of 3D bioprinting.

17.3 Informational Practices

Unlike the generic nature of traditional medical devices suggested above, the bioprinting of a liver couples human specific information (e.g., cells from the patient and a scan of the existing liver) with the information used to produce all bioprinted livers, that is, the geometry of the 3D model and the printing software. This makes each product uniquely applicable; each person needing a liver would have one printed. Implanting a liver based on donor cells, rather than the recipient's cells, represents an inferior (although viable) choice. One of the primary advantages of using the recipient's own cells is the reduced probability of tissue/organ rejection and the need for anti-rejection drugs. Thus, the informational facets of 3D-bioprinted objects play a crucial role in the printing process. Contrast the ink used in a normal inkjet printer and the human specific medical information used in the 3D bioprinter. The ink can be of different brands and colors. Using cells from someone other than the donor in the *bio*printing process can have adverse effects when the tissue is implanted in a human body.

As explained above the output of 3D bioprinters contains information from three different sources. The recipient provides the cells, yielding information for cellular replication and function, as well as the 3D scan of existing tissue, yielding an informational model for the replacement tissue. The researcher provides 3D-geometry information and the printer manufacturer provides software to drive the printer. Once the cells are deposited on the geometry by the printer, they grow and differentiate to form the completed tissue that is to be implanted. The resulting output is an object that represents a most complex form of transaction generated information (TGI). In a credit card transaction, information from three sources (the customer, the merchant and the bank) is combined to create the informational object. Yet, in this case, the informational object has a physical manifestation that carries unique information identifying a particular person and is the result of intricate interplay between informational, manufacturing and biological growth processes. In the case of credit card TGI, in an after-the-fact analysis one can identify where each component of the information originated. There is a high level of separability. A similar notion of separability applies to traditional medical devices such as a pacemaker in that the intellectual property of the manufacturer can be easily identified in an image or removed and separated from the recipient via a second surgery.

In contrast, it appears that bioprinted tissue has no clear path to separating the individual pieces of intellectual property that come from the printer manufacturer or the researcher without removing the piece that comes from the recipient. At the very least, it is unclear how much of the structure, and the information contained therein, of 3D-bioprinted tissue can be attributed to cell differentiation, how much to the initial geometry and how much to the bioprinting itself. Some of the macro-structure might have clear ties to the geometry, and some of the micro-structure might be clearly linked to cell growth and differentiation. But origins of mid-level structure might not tip so neatly in a particular direction. It is also the case that the work of the bioprinter may not be evident in the finished tissue due to cell growth. Another sign of the deep integration of the initial geometry, the printing information and the cell information is the irreversibility of the cell growth and differentiation process. Hence, this inseparability leads to questions of ownership and attendant rights and responsibilities. An additional complexity twist comes when the tissue is surgically implanted into the recipient.

One of the obvious concerns is the issue of accountability and responsibility should there be a problem with the tissue. If the problem is detected prior to implantation, there is clearly an opportunity to reprint the tissue, and, other than problems that might be introduced due to the additional delay, there is no cause for concern. Still, this raises the question of who has the responsibility of determining the viability of the tissue. Unlike a traditional liver transplant, for example, where the surgeon knows that the liver to be implanted has indeed functioned as a liver already, the viability of a 3D-printed liver as a liver is less certain. Unlike traditional medical devices where the design, manufacturing processes and device itself all undergo rigorous testing, approval and inspection processes to ensure the proper functioning of the device, there currently do not exist similar approval and inspection processes for 3D-printed tissues. Furthermore, the complexity of the process of 3D printing, with its roots in manufacturing, information processing and biological processes, suggests that quality control might be difficult. Clearly, this is an issue that needs to be addressed as part of the development of 3D bioprinting.

It is not unreasonable to expect that techniques from software development and testing enter into the quality control process for 3D-printed tissue. This suggests, at the very least, the need for collecting information about printed tissues that were viable as well as those that were not viable, so that appropriate analysis can be done. Questions about whether the viability of the tissue had to do with the donor's cells or scan, the geometry or the printing software are sure to be part of the analysis. The analysis is complicated by potential interactions among the various information sources. It may be the case that a particular donor's cells are incompatible with a particular geometry. That is, the geometry interferes with the way in which these particular cells grow and differentiate, leaving open the possibility that recipient's DNA information may well become part of the testing and quality control process for 3D-bioprinted tissue in the future. At the very least, this possibility calls for informing recipients of the informational nature of this process and how their information might be used, as well as obtaining consent prior to such use. Software developers working in this arena need to anticipate these sorts of needs and build in adequate informational privacy and security mechanisms.

In addition, stringent and appropriate controls on the information, including anonymization, are in order. Once the individual information is aggregated, it is reasonable to expect that developers might mine that information for information that will help improve the quality of their product. It could easily yield other information, potentially helpful or harmful, that ought to be handled appropriately.

As part of a complex medical process 3D bioprinting ought to be afforded at least the same governmental oversight as other medical processes. Yet our analysis indicates that 3D bioprinting has features that make this technology even more complex and will likely require the development of new quality oversight policies and procedures. Software is an integral part of this process, and the process does not yield generic, testable copies. The recipient of the tissue could demand to inspect the program (or more likely have it inspected by a qualified proxy) driving the bioprinting process, due to the potential threat to the individual's autonomy. Given the intimate nature of the bioprinted tissue, the right to inspect the program extends to every recipient of such tissue. Due to the integral role of software and the difficulty of its inspection, the development of such processes ought to be occurring in parallel with the development of this technology. Furthermore, if the software[1] is proprietary, there will be a need for practices to change, since it is not clear that there is an argument to withhold proprietary software from the recipient of 3D-bioprinted tissue due to the software's potential threat to the individual's autonomy. Each person has a moral authority over one's own body that cannot be overridden.

17.4 Personal Identity Questions

Questions of personal identity have been pursued by philosophers for centuries. The traditional persistence identity problem arises due to continuous change over time of (animate and inanimate) objects. What makes something (or someone) the same thing (or person) despite ongoing changes? If objects endure and retain their identity nevertheless, it seems that they should somehow persist through such (or at least some) changes. This is a long-debated problem. Heraclitus famously wondered whether one could step into the *same* river *twice* given that the river continually undergoes changes. Theseus' ship is another example of identity persistence in spite of ongoing changes to the ship due to normal wear and tear: parts break and are subsequently repaired or replaced. Eventually, every part of the original ship will have been replaced, inviting the question: does it remain the same ship despite all changes through time and if so, what makes it the same?

Most importantly, for present purposes, persons likewise change through time, and we typically say that their identity persists through time despite obvious changes. Every person has very few properties in common with the infant version of themselves, most notably, their DNA information and, by implication, their blood type. What makes a person the same one through all the physical and psychological changes in her life? That is known as the persistence problem of personal identity.

[1] Some 3D printers use Open Source technology (e.g., http://www.fabathome.org/).

On one well-received view, the answer is that some psychological relation is necessary or sufficient for a person to persist over time. A famous proponent of this view was John Locke, who argued that a person is "a thinking intelligent being, that has reason and reflection, and can consider itself as itself, the same thinking thing, in different times and places" (1975: p. 335). While psychological properties are very important to our personhood, thereby providing a basis for a person's continuity over time through change, it is hard to identify those key psychological properties. An obvious property is memory: a person's identity is preserved only if that person can remember an experience she had in the past. Nevertheless, the very idea of remembering that you can remember your own experiences (e.g., riding your bike) is to remember *yourself* experiencing (you being the person riding the bike). Also, we typically forget most of the mundane events that happen despite being the same persons who underwent those mundane events. Advocates of the Psychological Approach offer various solutions to the memory criterion (cf. Lewis 1976; Parfit 1971; Shoemaker 1984 and Unger 1990). A more serious problem (known as the Fission Problem) for this approach is that a person could be psychologically continuous with two (past or future) people simultaneously (Parfit 1971). If the cerebrum of person P were transplanted in another person Q, Q would be psychologically continuous with P (despite some potential psychological differences).

Another well-known approach is Animalism, according to which *if* we are animals, we have the persistence conditions of animals (Olson 2010). Yet, Animalism does not entail that either all animals or even all human animals are people. Human embryos, for example, may not yet count as people. Accordingly, the person's persistence conditions are the persistence conditions of the animal she is, since animals appear to persist by virtue of some sort of brute physical continuity (Olson 1997). At the same time, Animalism is, arguably, consistent with the existence of wholly inorganic people, such as conscious robots, and does not entail that being an animal is necessary for being a person (Olson 2010). What makes someone the same person over time and throughout change is the spatio-temporal continuity of one's body. A person consists of her DNA, cells, organs and so on: determining whether any given person is the same person requires tracking their bodily continuity.

More recently, Floridi has offered an informational approach to addressing this question (2013, Ch. 11). He describes it as a three-phase process in the first phase of which an organism separates itself from the environment through the creation of a membrane. Internal to that membrane, it organizes itself on a physical level. The second phase is the development of a cognitive membrane that internally separates the part of the organism that is responsible for information processing from the remainder of the organism. The third phase is the establishment of "[t]he consciousness membrane [that] is soft-wired (programmable). The body becomes the outside environment for an inside experience, and stability now concerns the self within the system (mental homeostasis)" (Floridi 2013, p. 220).

In the past, these philosophical questions have been asked "in principle", yet in the context of 3D bioprinting, these philosophical concerns are turning increasingly practical and pressing. Furthermore, the informational nature of 3D bioprinting lends support to Floridi's approach of including the role that information plays in the identity of the individual. For a 3D-bioprinted tissue, or a transplanted tissue for that matter, the implantation is outside of the cognitive membrane, and as such, has, on Floridi's approach, no impact on one's personal identity. For the implanted tissue does not engage in information processing (broadly construed). At the same time, this informational approach goes against Animalism. For the body may go through many changes resulting from implanting 3D-bioprinted tissues and organs that do not go beyond the first membrane. And, thus, these changes leave the personal identity of the individual unaltered.

More interesting cases arise when the newly bioprinted tissue is itself capable of information processing. A nerve fiber is one such example that introduces a potential for forming a second cognitive membrane in the organism. The most interesting case occurs with the formation of a second consciousness membrane in the organism. While this seems to be currently outside of the technical possibility of 3D bioprinting, some interesting questions arise concerning the interplay between "internal" and "external" sources of information. External information sources used to produce the 3D-bioprinted tissue might influence the information processing taking place inside the body after implantation.

17.5 Concluding Remarks

The ethical concerns we have identified in this paper mainly arise from how 3D bioprinting emphasizes the informational nature of human tissue and people. It seems clear that since 3D bioprinting is an informational process, producing bioprinted tissue requires medical and, possibly, genetic information about the intended recipient as part of the software development process associated with 3D bioprinting. This, in turn, raises concerns about how that information is handled. In a traditional medical setting the personal information is shared only between the patient and the doctor, and that sharing is governed by patient/doctor confidentiality, best practices and laws. 3D bioprinting, however, also includes a third party handling patient medical information. At first blush, these concerns seem similar to those that surround the production of lab-grown tissues. Yet, there are important distinctions. In the case of lab grown tissues, the information that is used is often treated in a non-informational way: it is genetic "stuff" taken from the patient's cells that is manipulated in the lab.

The processes surrounding 3D bioprinting, on the other hand, are subject to concerns that arise from information transfer, modification and replication, as well as those that arise from the software maintenance and upgrade process. Errors in the software need to be identified and reported to the developer in order for them to be fixed. In addition, there is a likely need for additional medical or genetic information to be incorporated into the quality control processes of bioprinters. This suggests that there is a need for the developer to have access to private medical information of individuals, complicating the traditional patient/doctor relationship.

Further concerns arise when one considers the need for information flow between the developer and the bioprinter, in both directions. At the very least, this places at risk the individual's private information. In addition, the suggestion of bioprinter network connectivity (e.g., for regular software updates) opens the possibility of an attacker injecting malware into the bioprinting process. This stands in stark contrast to the impact of a malicious act on or at a single lab producing similar tissues.

A related concern stems from potential attacks from third parties on the software driving the bioprinting process. The threat of such an attack is quite real for any device that is connected to the Internet (Wolf and Fresco 2014). So, a simple response is the handling of software upgrades and data collection from the device in an offline manner. Restricting the sale of 3D bioprinters to licensed medical facilities is a further protection against a third party attack. An attacker bent on infiltrating a 3D bioprinter would have a difficult time obtaining either physical or online access to such a device and, due to the relatively limited market for these devices, developers would not be unduly burdened.

While 3D bioprinting holds the promise of alleviating a number of problems in the medical field, there are steps that can be taken at this time to ensure that people and their medical information are treated with the same sort of care and legal protections that are more prevalent in the medical community than those found in the software development community.

References

Anker, S. (2009). Cultural imaginaries and laboratories of the real: Representing the genetic sciences. In P. Atkinson, P. E. Glasner, & M. M. Lock (Eds.), *Handbook of genetics and society*. London/New York: Routledge.

Bertassoni, L. E., Cecconi, M., Manoharan, V., Nikkhah, M., Hjortnaes, J., Luiza Cristino, A., Barabaschi, G., Demarchi, D., Dokmeci, M. R., Yang, Y., & Khademhosseini, A. (2014). Hydrogel bioprinted microchannel networks for vascularization of tissue engineering constructs. *Lab on a Chip. 14*(13), 2202–2211. doi:10.1039/C4LC00030G.

Boyce, J. S. (2012, July 14–17). Ethical paradigms in biomechanical innovations in nanotechnology and neurotechnology. In S. Sendra & J.C. Metrôlho (Eds.), *Recent researches in communications and computers: Proceedings of the 16th WSEAS international conference on communications, proceedings of the 16th WSEAS international conference on computers, Kos Island, Greece,* WSEAS Press. www.wseas.org.

Fedorovich, N. E., Alblas, J., Hennink, W. E., & Dhert, W. J. A. (2011). Organ printing: The future of bone regeneration? *Trends in Biotechnology, 12*, 601–606. doi:10.1016/j.tibtech.2011.07.001.

Floridi, L. (2013). *The ethics of information*. Oxford: Oxford University Press.

Hockaday, L. A., Kang, K. H., Colangelo, N. W., Cheung, P. Y. C., Duan, B., Malone, E., … Butcher, J. T. (2012). Rapid 3D printing of anatomically accurate and mechanically heterogeneous aortic valve hydrogel scaffolds. *Biofabrication*, *4*(3), 035005. doi:10.1088/1758-5082/4/3/035005.

Lewis, D. (1976). Survival and identity. In A. Rorty (Ed.), *The identities of persons*. Berkeley: University of California Press.

Locke, J. (1975). In P. Nidditch (Ed.), *An essay concerning human understanding*. Oxford: Clarendon.

Mironov, V., Visconti, R. P., Kasyanov, V., Forgacs, G., Drake, C. J., & Markwald, R. R. (2009). Organ printing: Tissue spheroids as building blocks. *Biomaterials, 30*(12), 2164–2174. doi:10.1016/j.biomaterials.2008.12.084.

Nealy, E. (2014). *The risks of revolution: Ethical dilemmas in 3-D printing*. Proceedings of ETHICOMP 2014.

Olson, E. T. (1997). *The human animal: Personal identity without psychology*. New York: Oxford University Press. www.wseas.org.

Olson, E. T. (2010). Personal identity. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy.*http://plato.stanford.edu/archives/win2010/entries/identity-personal/. Accessed 31 May 2014.

Parfit, D. (1971). Personal identity. *Philosophical Review, 80*, 3–27.

Ranaldi, R. (2014). *Medical 3D printing: Printing a new face for the future*. Presented at the interactive multimedia conference 2014, University of Southampton, U.K.

Reynolds, G. (2013). *BioPen rewriting orthopaedic surgery*. http://www.uow.edu.au/research/profile/UOW162827.html. Accessed 31 May 2014.

Shoemaker, S. (1984). Personal identity: A materialist's account. In S. Shoemaker & R. Swinburne (Eds.), *Personal identity*. Oxford: Blackwell.

Simon, M. (2013). When copyright can kill: How 3D printers are breaking the barriers between "intellectual" property and the physical world. *Pace, I.P., Sports & Entertainment Law Forum*, *3*(1), 60–97.

Unger, P. (1990). *Identity, consciousness, and value*. Oxford: Oxford University Press.

Wolf, M. J., & Fresco, N. (2014). *Sploits for sale! And that just might be ethical*. Proceedings of ETHICOMP 2014.

Chapter 18
Robots, Ethics and Software – FOSS vs. Proprietary Licenses

Marty J. Wolf, Frances Grodzinsky, and Keith W. Miller

Abstract The sociotechnical system of a robot, including its software, matters. Context is central to analyzing the ethical implications of that software for the public and the ethical responsibilities of the developers of that software. Possibly morally neutral concepts such as mass production, information storage, information acquisition, connectivity, ownership and learning can have a collective positive or negative ethical impact for a world with robots. Since robots are a type of artificial agent (AA), we start with a claim by Floridi that the actions of AAs can be sources of moral or immoral actions. Because AAs are in essence multi-agent systems, we apply Floridi's Distributed Morality (DM). In this paper, we will analyze proprietary and open source licensing schemes as a policy component of DM and show the distinctions between software licensing schemes in terms of how they work to "aggregate good actions" and "fragment evil actions" for the uses and features of robots now and in the future. We also argue that open source licensing schemes are more appropriate than proprietary software licenses for robot software that incorporates the results from automated learning algorithms.

18.1 Introduction

Humanoid robots will combine sophisticated hardware and sophisticated control software. The ethical significance of the sociotechnical system in which such a robot is embedded is further influenced by the robot's ability to have constant

M.J. Wolf (✉)
Department of Mathematics and Computer Science, Bemidji State University, Bemidji, MN, USA
e-mail: mjwolf@bemidjistate.edu

F. Grodzinsky
Computer Science and Information Technology Department, Sacred Heart University, Fairfield, CT, USA
e-mail: grodzinskyf@sacredheart.edu

K.W. Miller
The Educator Preparation, Innovation and Research (EPIR) Department, University of Missouri – Saint Louis, St. Louis, MO, USA
e-mail: millerkei@umsl.edu

V.C. Müller (ed.), *Computing and Philosophy*, Synthese Library 375,
DOI 10.1007/978-3-319-23291-1_18

network connectivity. Robot connectivity is a two-way relationship: information (some of it unanticipated by the robot developers) may affect the robot whenever the robot is connected to the outside world, and information from the robot (some of it unanticipated by the networked recipients) may affect the outside world. Furthermore, humanoid robots are designed to exist in spaces that are occupied by people, including (in some cases) infants and the infirm. Some robots have the ability to modify physical spaces where people exist in direct (and sometimes seemingly arbitrary) ways, placing the software that drives robots into the category of potentially life-critical software. It is not unreasonable to suggest that most developers of such robots are interested in developing "morally good" robots; this presumably will lead to maximizing their profits, enhancing their reputations, and helping developers feel good about their work. Yet, robots that learn and change their programming introduce complexities to the issues of responsibility and accountability when those robots are mass produced in the sociotechnical context outlined above.

Robots that learn, and learn in ways similar to humans, are no longer technically out of the question. Merolla et al. (2014) have developed a spiking-neuron chip that models brain neurons, including their propensities for being connected to many other neurons, each at different strengths. The new chips have the ability to send simulated "spikes" to their neighbors. Other planned investments by governments and corporations in research to understand the human brain, and in particular, the human learning process will have an impact on the development of humanoid robots. Separate concepts that may appear morally neutral in isolation – concepts such as information storage, information acquisition, connectivity, ownership, mobility, and learning – could possibly have a collective ethical impact. According to Floridi, artificial agents (AAs) "… can be legitimate sources of im/moral actions, so the ethical discourse should include the analysis of their design, deployment, control and behavior" (Floridi 2013b, p. 728). Both collections of interconnected robots and the individual robots themselves are in essence artificial multi-agent systems (MAS). As such, they are subject to Floridi's Distributed Morality (DM), and we will rely on Floridi's DM to guide our analysis (2013a, ch. 13). One application of DM raises questions surrounding the ownership of the product of a robot's learning process. Another application raises questions that stem from the integration of information the robot obtains locally with information that is obtained publicly and privately on the Internet. Closely related to this question is one of responsibility and accountability for various aspects of the robot life-cycle.

Elsewhere, we have argued that different software categories lead to different conclusions about the ethics of various software licensing schemes (Wolf et al. 2009). In this paper, we analyze proprietary and open source licensing schemes as a policy component of DM and show the distinctions between the software licensing schemes in terms of how they work as policies to "aggregate good actions" and "fragment evil actions" for the uses and features of robots, now and in the future.

In the next section we summarize Floridi's DM, define some useful terms and make clear a distinction between the types of software systems in robots. In Sect. 18.3 we identify a variety of classes of information that are present in the lifecycles of robots. Some of that information is software. In Sect. 18.4 we examine the adequacy of both proprietary and open source software licensing for addressing ethical issues associated with robots in a DM system. We explore how different software licensing schemes function as aggregators and fragmenters for different classes of robot software and thus enhance or prevent the flourishing of both humans and humanoid robots. In the rest of the paper we use an informational analysis to articulate the complexities of these interactions, and we find that transparency is required to varying degrees for certain types of information. We conclude the paper in Sect. 18.5.

18.2 Background and Definitions

Floridi contends that DM is to be applied "to cases of moral actions that are the result of otherwise morally neutral or at least morally negligible … interactions among agents constituting a multi-agent system" (Floridi 2013a, p. 262). Even actions that are morally neutral and morally negligible in isolation can coalesce to create an action that is clearly morally good or morally evil. Consider, for example, a driverless car that is programmed to head to a fuel station when the fuel tank reaches a certain level. An image processing routine is adapted from an earlier program to locate fuel stations. The developer is confident about its efficacy, so it is not tested extensively. Thus, the routine's subtle, but statistically significant bias for one company's fuel stations goes unnoticed. When only a few driverless cars are operational, the effect is negligible. This "fuel station seeking software" is eventually adopted by many driverless car manufacturers. As the number of driverless cars increases, a statistically significant bias favors one local company over another (although the bias is subtle enough that it is not discovered). Eventually, that company goes out of business (or worse, just scrapes by, enduring years of misery) due at least in part to this unfair bias being present in many driverless vehicles. Thus a morally negligible feature becomes morally significant by aggregation.

To use DM effectively requires establishing systemic barriers that reduce the inertia of morally neutral actions moving toward aggregation into a morally evil action and enhancing systemic features, such as resilience and fault-tolerance, to support the aggregation of morally neutral actions into morally good actions. In Sect. 18.4 we will analyze classes of software licenses and their effectiveness as "fragmenters", which isolate and neutralize possibly evil actions, and "aggregators," which enhance the probability of possibly good actions coalescing to make their environment better for its inhabitants.

The classes of software licenses we consider involve Free Software (FS), Open Source Software (OSS) and Proprietary Software (PS). The typical distinction between FS and OSS is that FS requires that when a modified version of the code is distributed, the new code must be licensed as FS. OSS licenses typically do not establish such a requirement. In this paper, we use FOSS to refer to both FS and OSS. Generally, there are no restrictions placed on running, copying, and modifying FOSS. FOSS source code is typically available at no cost. PS stands in stark contrast to FOSS in that the source code is expressly not available to a user, users are typically barred from making copies of the software, and the initial cost is significantly higher than FOSS. Furthermore, PS cannot be legally modified by anyone other than the developer.

Humanoid robots are and will increasingly be complex sociotechnical systems that include both hardware and software components. The hardware components such as actuators and sensors do not play a role in our ethical analysis in this paper. Hardware components that are computational in nature, such as the spiking-neuron chip, and software components are another matter. Presumably, these control components will be complex and highly integrated. We will refer to such integrated components as software systems, i.e. the software and the hardware required to run the software. Two robotic software systems of importance are Decision Making Software (DMS) and Learning Software (LS). For our purposes, DMS is the software system that is used to make decisions about how a robot will next interact with its environment. Presumably, much of this system will be event driven, i.e., it will respond to events that happen in its physical and electronic environment. LS is the software system that the robot uses to change its future decision making. Our discussion will not include software systems for things such as controlling physical movement and electronic communications. ("DM" is often used in the literature to identify a machine acting as a "Decision Maker"; for example, see (Wang and Shen 1989). We avoid that acronym here to prevent confusion with "Distributed Morality".)

18.3 Robot Actions and Their Moral Significance

DMS and LS provide starting points for our ethical analysis. However, they are not sufficient. As Floridi points out, "DM is made *increasingly* possible by multiagent systems, which in turn are made *increasingly* possible by extended, pervasive and intensive interactions" (Florid 2013b, p. 736). According to Floridi, technological mechanisms facilitated by information and communication technology work as moral enablers. In this section we consider the sorts of interactions robots have with their environments and the sorts of information that stem from those interactions. It is these sorts of interactions that demonstrate robots unusual technological niche: they are in physical contact with people including those who are not their owners or even potentially known by their owners. This contact increases the potential for physical harm to humans relative to non-mobile systems and systems where the interaction between the AA and the human is exclusively electronic.

Consider for example, a robot designed to clean floors in a home. It has very little learning to do. There is no reason to expect it to learn much more than the layout of the room. A simple guidance system is sufficient for it to determine the outline of the room and the placement of the furniture in the room, thereby allowing the robot to accomplish its task. On the other hand, humanoid robots we envision in the future will require significant information acquisition in order to function appropriately. Within this context, we contend that much of a robot's information acquisition is made up of individual actions that appear, in isolation, to be morally negligible. (For now we will ignore several ways in which robot information acquisition could be explicitly harmful; for example, a robot spy could be explicitly designed to violate privacy.) In what follows we explore the nature of a robot's information acquisition to support our designation as morally negligible and examine the potential distribution of that gathered information. We begin with the manufacturing process.

The information initially acquired by the robot is the software that runs its systems. Once the robot is placed into service, it will begin to acquire additional information from its surroundings and from its interactions with the people and other robots it encounters, both physically and electronically. Some of that information will be acquired passively, merely through observation: "The door is open." "The center of the table is two meters from the wall." Other information will be acquired through the interactions it has with those around it: "The owner did not respond to the robot's request where to put the clean tea pot." (Note the negative flavor of this information.)

In addition, a robot connected to the Internet will have access to the breadth of information and misinformation found there as well. Some of that information will be public in nature, but some will be more private. It seems reasonable for a robot to proxy for its owner on social media and other online venues. In addition, the robot may serve as a conduit in the other direction, to report to the owner what is taking place online. On the whole, we expect that many of these sorts of actions will be morally neutral, but at least some of them are likely to have positive or negative moral significance.

There are two further types of informational interactions that are important to our analysis. One is that the robot will acquire information from the manufacturer in the form of software upgrades and security patches. It seems that there is no strong moral bias to the action of installing upgrades and patches. Often they increase security, but sometimes the process introduces instability into the system. Thus, particular upgrades and patches could be positive, negative, or neutral morally; but in general, such inputs into a robot system seem likely to be, as a class, slightly biased toward the positive (upgrades and patches should be designed for the greater good, and we think they probably will be).

We anticipate that another important interaction is the information flow that will originate with the robot and terminate with the robot developers. As a sophisticated machine that can continuously sense and record data, there is a legitimate need for the developer to have access to at least some of that data in order to ensure that the robot is performing adequately. While this sort of information is routinely collected by our desktop computers today, we often give permission for such information to be sent to the developer. However, the sort of information a robot might collect (and potentially send) could be far more invasive for humans that interact with the robot due to the physical nature of the interaction. In the mobile communications devices of today, we already see glimmers of the sort of issues that might arise tomorrow with robots as detailed personal information, location information, and complex interactions between concurrently executing applications are often used by the applications and shared with developers and third parties as well.

There is certainly concern about such data collection and dissemination. For example, there is currently a movement in the United States to ban “stalking apps” for smart phones. These are apps that are installed surreptitiously, and (unknown to the holder of the phone) report the location of the phone to a third party. Closely related is a move to regulate apps that gather location information and report it back to the app developer. Currently developers are allowed to gather that data without the owner’s consent and to do with that data as they please, including selling it to others. Thus, the sending of information from the mobile device is both possibly good (it helps recommend a nearby restaurant or improve the quality of driving directions to the user) and possibly evil (allows a stalker to harass the phone holder). The sending of location information from a phone is itself a morally negligible action that can “go either way” depending on how that action is used.

We contend that advanced robots will exhibit some of the same behaviors we now find troubling when humans interact with mobile communications equipment. Just as phone apps gather personal information about people, robots will do this as well. Just as apps now combine and communicate this information (often without notice to or control by human), robots will likely do the same. In fact, we expect that because robots will be increasingly free from direct human control after launch, and because many robots will be able to interact with many different people, the robots will likely exhibit more of this concerning behavior than smart phones exhibit now. People typically carry mobile devices voluntarily; robots will interact with people who have not volunteered for that interaction.

The most complicated collection of information, at least in terms of provenance, comes into play when a robot begins to learn. At that point, the robot is generating new information by integrating information from a variety of sources. In a simple case, the robot could create a database of information about the habits of the robot’s owner and those in the owner’s household. This sort of information has only weak integration with the software systems of the robot in that extracting the information and deleting it can be done without changes to the fundamental programming of the robot. A more complicated case is when the learning robot makes changes in its programming due to its experiences and what it has learned. In this case, undoing the learning (in order, for example, to enforce a privacy constraint) may be impossible without changing a significant amount of programming, especially if the underlying computational structure is an artificial neural network. It is in this case that ownership of the software begins to come into question. It is clear that the robot’s new programming is in place because the original programming was designed to do such learning, but the resulting programming (likely) would not have come about without the information obtained while the robot was under the control and direction of its current human owner. Both the developer of the robot and the current owner of the robot have a partial claim to the newly generated software of the robot. This ownership claim includes both privilege and responsibility for the developers and for the owners.

The issues become more complex when the robot's learning includes the integration of information from across the Internet in general and from other robots in particular. This is especially true when the externally obtained information is programming and when that programming helps to generate even more new programming. This demonstrates the depth and complexity of a multi-agent system that is a humanoid robot interacting with its environment. The complexity is compounded in a collection of robots that share a common purpose. In the following section we explore ethical concerns surrounding these issues and examine how software licensing can act as an ethical policy in DM.

18.4 Ethical Concerns

Work by Turkle (2011) demonstrates how people can develop emotional attachments to robots, attachments of the type that until recently had been reserved for other humans and pets. Elsewhere we have argued that for developers, "[s]ignificant care is in order when the potential impact of technological decisions enters into the human realm" (Grodzinsky et al. 2014). This is one of those realms. In human relationships, humans routinely assume that the other human is making decisions in a manner familiar to humans. Although this assumption is sometimes flawed, it is often valid. Such an assumption about a humanoid robot is likely to be invalid, at least for the foreseeable future. Yet, the assumption is encouraged when humanoid robots are designed to mimic humans.

Since people should not trust their initial assumption that a robot's decision-making is similar to their own, the openness of FOSS is clearly helpful for human beings. When the source code of a DMS can be inspected, people who might be affected by the robot's behavior can learn about the DMS process. Clearly, most people are not skilled enough to do this inspection with understanding, and they might not have the time or inclination to do so, but they could hire someone to do it. Another mechanism for institutionalizing the inspection of FOSS DMS would be to subject such code to oversight by a governmental or independent agency. (The independent company UL, earlier United Laboratories, is an existing example of an independent organization that issues certificates about product safety (UL 2014).)

Any attempt at inspection and oversight of robot code opens questions about jurisdiction and the efficacy of such oversight for PS. FOSS is immediately compatible with inspection, since access to source code is an inherent feature and allows for third party experts to analyze the source code, much in the same way cryptographers dissect candidate algorithms for cryptographic standards. The potential for such inspection, at the very least, suggests the manufacturer is willing to establish a trust relationship with customers. PS seeks to guard code from outsiders' view and will greatly complicate any process for code inspections.

This analysis is complicated in cases like the spiking-neuron chip where the underlying hardware is non-algorithmic in nature. Among other things, this chip implements in hardware an existing neural network software package. Traditionally, hardware is "closed-source" meaning that its design is not available for public inspection. Thus, any sort of software licensing policy intended to achieve a moral goal with software would need to be extended at least to some hardware as well.

For robots that interact with people, trust is of particular importance. A robot's LS will presumably initiate changes in the way it interacts with people in its environment. A robot with an LS that is influenced by others could demonstrate unexpected changes in behavior. Having access to the robot's current LS source code can help those modifying robots better understand the robot. Subsequent changes can impact the trust relationships between the owner and the robot, and between the owner and the robot producer. In addition, people who are not the owner or the producer, but can be affected by the robot's behavior, will also have a stake in understanding the robot's LS and its current DMS.

The ownership of informational artifacts not created by humans is not something that has been considered until recently. A recent case is a dispute between Wikipedia and David Slater over a selfie a monkey took with Slater's camera. According to Kravets (2014), Slater claims that as the owner of the camera, he is the copyright holder, and Wikipedia's owners claim that the monkey, as the taker of the picture, is the owner of the copyright, and, since the monkey is not human, the picture falls into the public domain. While this case is currently being sorted out in court, it has implications for the more complicated question of the ownership of software produced by a robot's LS when it writes new source code for itself while under the direction of the owner. Neither the manufacturer of the robot, nor the owner of the robot wrote the code, yet the new code gets integrated into the code base of the robot, or in the case of the spiking neuron chip is represented in data stored on the chip. It is not clear how the manufacturer can lay sole claim to the new code, since it was generated in response to experiences of the robot under control of the owner. The owner surely has a legitimate claim to a role in the development of this code. Yet, it also is difficult for the owner to lay exclusive claim to ownership, since the developer created the LS that generated the new code. Furthermore, extending the monkey-selfie argument to claim that the software is in the public domain since a robot cannot be a copyright owner also seems problematic because the software stems from the efforts of the human developer in conjunction with efforts of a human owner.

When the code generated by a robot's LS has strong positive ethical implications, it is straightforward to use Floridi's Information Ethics (2013a) to argue that it is morally approvable to distribute the code to other robots so that they can make decisions in a manner that is more ethically effective. On one hand, if the robot's code is PS, the distribution might be detrimental to the commercial interest of both the manufacturer and the owner. FOSS, on the other hand, encourages this kind of sharing and is a step in what Floridi describes as "harnessing [DM's] power in the right way" (2013a, p. 261).

Open source code for robots would advance the study of robotic artifacts from an ethical perspective. If morality and ethics constitute a public system, then anything that contributes to its understanding should be open. Others may counter that if we observe analogous human behavior, we only see the external manifestations of those acts and not necessarily the motivators or learning process involved. Extending that argument to LS and DMS of a robot suggests that the robot, and possibly the developer or owner, should decide whether the LS and DMS should be proprietary, shared, or possibly sold. However, we argue against the idea that the human situation should restrict public access to robot software. Our position is further strengthened in the case that the robot has its learning software embedded in an artificial neural network whose neuronal behavior is similar to human neurons. Such an opening, at least to neuroscientists, may further our understanding of human morality as well as "artificial morality."

The nature of the "new code" is important to the argument for other reasons. If the code is in a standard programming language, it will be relatively easy to inspect, either by appropriately trained people or by an automated system. But there is no reason to assume that this will be the case. A common critique of artificial neural networks is that they do not provide explanations of the decisions they make. This lack of explanation is likely to be present in a new configuration of a network that represents some level of learning. If such opaque learning can be incorporated into other artificial neural networks, there is a need for explanation facilities so that its contribution can be evaluated appropriately. It is not clear that FOSS helps here due to the opacity of the "new code". Rather, this suggests that developing a "moral enabler" such as some sort of software that effectively describes the semantics of artificial neural networks ought to be part of the process of developing the LS system.

18.4.1 FOSS as Fragmenter and Aggregator

Floridi invites us to analyze sociotechnical systems to identify aspects that aggregate potentially good actions and fragment potentially bad actions. FOSS used in robots displays these salutary characteristics in the following ways:

1. We consider it a good thing when robot researchers and developers have a tendency share their wisdom about robot control with other researchers and developers. Clearly this sort of sharing exists; for example, whenever robot researchers and developers write for scholarly publications and present their work, this sharing takes place. FOSS encourages and facilitates exactly this sort of sharing. FOSS sharing is at a low level of detail, which can give concrete help to the robot development community directly, and indirectly to robot users and others affected by the use of robots. Thus FOSS aggregates the good actions of sharing information about robot development into the development of autonomous agents that can potentially demonstrate moral actions.
2. A FOSS model is also consistent with the development model of the software that is a product of the learning process. There are clearly numerous contributors. This is especially true in the context of mass-produced robots that communicate with one another. The new software may represent contributions from multiple robots

with multiple owners. Economic incentives play a secondary role to the good that comes from robots that perform better, especially when making decisions with moral consequences.

3. We consider it a bad thing when robot researchers and developers include (either intentionally or accidentally) programmed behaviors in a robot that will tend to harm persons or property. These bad actions can be mitigated more quickly and effectively when robot software is FOSS. The open nature of FOSS thus fragments bad actions by exposing these actions (in this case, programming actions that lead to bad consequences) to examination and criticism. FOSS can also help to fragment the impact of bad software developed by the robots themselves.

18.4.2 PS as Fragmenter and Aggregator

We have argued that the transparency of FOSS used in robots can act as a fragmenter of bad programming actions and as an aggregator of good information sharing actions by robot researchers and developers. PS does not share the transparency of FOSS, but the closed nature of PS does have its own salutary effects as a fragmenter or aggregator. PS used in robots might have these salutary characteristics:

1. A sophisticated, effective, and well-conceived robot could have positive effects for its developers, users, and others. Assuming such a robot could exist, much of its value will come from its complex software. Creating such software requires significant expertise, time, and effort. Each expert action required to develop this good software (and there would be a host of such actions required) can be aggregated into an effort to produce the robot and its beneficial behaviors. Insomuch as PS (it can be argued) protects developers' investments of time, effort, and money in producing the robot software, PS acts as an aggregator of good software development actions.
2. PS can simplify accountability, especially in cases where the software development team is relatively small and well defined. If PS developers embrace their responsibility and accountability for any shortcomings of their software, then PS can fragment bad programming actions by encouraging PS developers to detect and correct their mistakes.
3. A difficulty PS faces is dealing with responsibility and accountability for the learned software. Should learned software face the scrutiny of the developers before it is integrated into a robot, adoption of good changes might be slow.

18.5 Conclusions

We have presented arguments about robot software that is FOSS or PS and how each category of software has different characteristics relative to the aggregation of good actions and to the fragmentation of bad actions. The arguments for FOSS as an aggregator and fragmenter seem stronger in that ownership issues do not interfere with the development process. Our judgment is based on our observation of the history of FOSS and PS in applications that are not connected to robots. Traditionally, FOSS developers have actively shared their expertise with other FOSS developers and the public; furthermore, errors and shortcomings in FOSS have indeed been rectified with the "many eyes" on the explicitly shared source code. Thus the characteristics we have cited for FOSS used in robot software seem likely to emerge in the same way these characteristics have emerged in FOSS software used in other applications.

In the case of PS for robots, it seems likely that, as with PS for non-robots, the profit motive, and the protection that PS affords developers' economic interests, is likely to motivate the creation of sophisticated systems. However, the second argument in support of PS, the simplification of accountability for software errors, seems likely to be offset by PS developers' traditional refusal to take responsibility for any shortcomings in their software. Indeed, while all software licenses are infamous for their efforts to disclaim any warranty, PS software licensing agreements do so without the transparency of FOSS. This greatly weakens the likelihood that PS will include a tendency to fragment bad programming actions for robots. In fact, PS (as it is traditionally practiced by most software developers) is likely to aggregate bad programming actions for robots by hiding programming from the public view, and by giving legal "cover" to developers who wish to escape responsibility for bad software outcomes. These behaviors (limiting software visibility and discouraging accountability) are especially problematic in robots that learn.

In the context of our analysis, FOSS robot software will (in an information ethics sense) tend to be more ethical than PS robot software. Part of our analysis depends on the PS robot software developers continuing to act in ways that current PS developers do. A future where the developers of PS robot software adopted an approach with more features of FOSS software would change our analysis. However, we see no evidence of such a trend so far in existing robot PS development.

Based on this analysis, we claim that the software that is used to drive humanoid robots that interact in environments where people are routinely present ought to be FOSS. As multi-agent systems in a DM, robots programmed in FOSS will be more likely to demonstrate good moral actions if their code is open to many developers. As a policy for a DM system, FOSS is a policy that will encourage aggregation and morally enable individual technologies to act in concurrence for the betterment of society.

Some of the arguments we have presented here with respect to robot software may be applicable to other software uses, but we have not justified that extension in this paper. The sociotechnical context of the software in the DMS and LS in robots differs substantially from most other software due to a unique combination of features. A robot's likely use is in close physical proximity to humans. A mass-produced, self-mobile consumer commodity with humanoid features invites the possibility of new morally relevant actions. This reinforces the claim that the case for openness in the DMS and LS in robots is strong.

References

Floridi, L. (2013a). *The ethics of information*. Oxford: Oxford University Press.

Floridi, L. (2013b). Distributed morality in an information society. *Science and Engineering Ethics, 19*, 727–743.

Grodzinsky, F. S., Miller, K. W., & Wolf, M. J., (2014). Developing automated deceptions and the impact on trust. *Philosophy and Technology*. doi:10.1007/s13347-014-0158-7.

Kravets, D. (2014). Monkey's selfie at center of copyright brouhaha. *Ars Technica*. http://arstechnica.com/tech-policy/2014/08/monkeys-selfie-at-center-of-copyright-brouhaha/. Accessed 11 Aug 2014.

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., Jackson, B. L., Imam, N., Guo, C., Nakamura, Y., Brezzo, B., Vo, I., Esser, S. K., Appuswamy, R., Taba, B., Amir, A., Flickner, M. D., Risk, W. P., Manohar, R., & Modha, D. S. (2014, August 8). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science*, *345*(6197): 668–673. Epub 2014 Aug 7. Accessed 11 Aug 2014.

Turkle, S. (2011). *Alone together*. New York: Basic Books.

UL. (2014). *About UL*. http://ul.com/aboutul. Accessed 10 Aug 2014.

Wang, H. F., & Shen, S. Y. (1989). Group decision support with MOLP applications. *IEEE Transactions on Systems, Man, and Cybernetics, 19*(1), 143–153.

Wolf, M. J., Miller, K., & Grodzinsky, F. S. (2009). Free, source-code-available, or proprietary: An ethically charged, context-sensitive choice. *ACM SIGCAS, 39*, 15–26.