Brian Cantwell Smith
brian.cantwell.smith@utoronto.ca
June 1, 2014

The Foundations of Computing

Brian Cantwell Smith

1 Introduction

Will computers ever be conscious? Is it appropriate—illuminating, correct, ethical—to understand people in computational terms? Will quantum, DNA, or nanocomputers require radical adjustments to our theories of computation? How will computing affect science, the arts, intellectual history? A1

For most of my life I have been unable to answer these questions, because I have not known what computation is. More than thirty years ago, this uncertainty led me to undertake a long-term investigation of the foundations of computer science. That study is now largely complete. My aim in this chapter is to summarize a few of its major results.[1]

2 Project

The overall goal has been to develop a comprehensive theory of computation. Since the outset, I have assumed that such an account must meet three criteria:

1. **Empirical:** It must do justice to—by explaining or providing the wherewithal with which to explain—the full range of computational practice; A2

1. This chapter is distilled from, and is intended to serve as an introduction to, a series of books that collectively report, in detail, on the investigation identified in section 2. The study of computing will be presented in *The Age of Significance* (Smith, forthcoming—henceforth AOS); the metaphysical territory to which that study leads is sketched in *On the Origin of Objects* (Smith 1996—henceforth O3).

2. **Conceptual:** It must as far as possible discharge, and at a minimum own up to, its intellectual debts (e.g., to semantics), so that we can understand what it says, where it comes from, and what it "costs"; and A3
3. **Cognitive:** It must provide a tenable foundation for the computational theory of mind: the thesis, often known as *computationalism*,[2] that underlies traditional artificial intelligence and cognitive science. A4

The first, "empirical" requirement, of doing justice to practice, helps to keep the analysis grounded in real-world examples. By being comprehensive in scope, it stands guard against the tendency of narrowly defined candidates to claim dominion over the whole subject matter.[3] And it is humbling, since the computer revolution so reliably adapts, expands, dodges expectations, and in general outstrips our theoretical grasp. But the criterion's primary advantage is to provide a vantage point from which to question the legitimacy of all extant theoretical perspectives. For I take it as a tenet that what Silicon Valley *treats* as computational *is* computational; to deny that would be considered sufficient grounds for rejection. But no such a priori commitment is given to any *story* about computation—including the widely-held recursion- or Turing-theoretic

2. The same thesis is sometimes referred to as *cognitivism*, though strictly speaking the term "cognitivism" denotes a more specific thesis, which takes mentation to consist in rational deliberation based on patterns of conceptualist (i.e., "cognitive") inference, reminiscent of formal logic, and usually thought to be computationally implemented (see Haugeland 1978). A5

3. As explained in A05, the aim is to include not only the machines, devices, implementations, architectures, programs, processes, algorithms, languages, networks, interactions, behaviors, interfaces, etc., that constitute computing, but also the design, implementation, maintenance, and even use of such systems (such as Microsoft Word). Not, of course, that a theory will explain any particular architecture, language, etc. Rather, the point is that a foundational theory should explain what an architecture is, what constraints architectures must meet, etc. A6

conception of computability, taught in computer science departments around the world, that currently goes by the name "the theory of computation."[4] I also reject all proposals that assume that computation can be *defined*. By my lights, that is, computer science is an empirical endeavor.[5] An adequate theory must make a substantive empirical claim about what I call *computation in the wild*:[6] that eruptive body of practices, techniques, networks, machines, and behavior that has so palpably revolutionized late twentieth century life.

The second, "conceptual" criterion, that a theory own up to—and as far as possible repay—its intellectual debts, is in a way no more than standard theoretical hygiene. But it is important to highlight, for two intertwined reasons. First, it turns out that several candidate theories of computing (including the official mathematical "theory of computation," mentioned above, as taught in computer science departments), as well as many of the reigning but largely tacit ideas about computing held in surrounding disciplines, implicitly rely, without explanation, on such substantial, recalcitrant notions as interpretation,[7] representation and semantics.[8] Second, which only makes

4. Indeed, I ultimately argue that that theory—trafficking in Turing machines, notions of "effective computability", and the like—fails as a theory of computing, in spite of its name and its popularity. It is simultaneously too broad, in applying to more things than computers, and too narrow, in that it fails to apply to some things that are computers. More seriously, what it is a theory of, is not *computing*. See §5.2.

5. Methodological issues arise, owing to the fact that we (at least seem to) make up the evidence. Although this ultimately has metaphysical as well as methodological implications, it undermines the empirical character of computer science no more than it does in, say, sociology or linguistics.

6. Adapted from Hutchins' *Cognition in the Wild* (1995).

7. "Interpretation" is a technical notion in computing; how it relates to the use of the term in ordinary language, or to what "interpretation" is thought to signify in literary or critical discussions, is typical of the sort of question to be addressed in the full analysis. A7

8. A notable example of such a far-from-innocent assumption is the

matters worse, there is a widespread tendency throughout the surrounding intellectual terrain to point to computation as a possible *theory of those very recalcitrant notions*. Unless we ferret out all such dependencies, and lay them in plain view, we run at least two serious risks: (i) of endorsing accounts that are either based on, or give rise to, vicious conceptual circularity; and (ii) of promulgating and legitimating various unwarranted preconceptions or parochial (e.g., modernist) biases (such as of a strict mind-body dualism).

The third "cognitive" criterion, that an adequate theory of computation must provide a tenable foundation for a theory of mind, is of a somewhat different character. Like the second, it is more a metatheoretic requirement on the form of a theory than a constraint on substantive content. But its elevation to a primary criterion is nonstandard, and needs explaining. Its inclusion is not based simply on the fact that the computational theory of mind (the idea that we, too, might be computers) is one of the most provocative and ramifying ideas in intellectual history, underwriting artificial intelligence, cognitive psychology, and contemporary philosophy of mind. Some other ideas about computing are just as sweeping in scope (such as proposals to unify the foundations of quantum mechanics with the foundations of information), but have not spawned their own methodological criteria here. Rather, what distinguishes the computational theory of mind, in the present context, has to do with the epistemological consequences that would follow, if it were true.

widespread theoretical tendency to distinguish (i) an abstract and presumptively fundamental notion of "computation" from (ii) a concrete but derivative notion of a "computer"—the latter simply being taken to be any physical device able to carry out a computation. It turns out, on inspection, that this assumption builds in a residually dualist stance towards what is essentially the mind/body problem—a stance I eventually want to argue against, and at any rate not a thesis that should be built into a theory of computing as a presumptive but inexplicit premise.

Theorizing is undeniably a cognitive endeavor. If the computational theory of mind were correct, therefore, a theory of computation would be reflexive—applying not only (at the object-level) to computing in general, but also (at the meta-level) to the process of theorizing. That is, the theory's claims about the nature of computing would apply to the theory itself. On pain of contradiction, therefore, unless one determines the reflexive implications of any candidate theory (of computing) on the form that the theory itself should take, and assesses the theory from such a reflexively consistent position, one will not be able to judge whether it is correct.[9]

More specifically, suppose that mind is in fact computational, and that we were to judge a candidate (object-level) theory of computing from the perspective of an implicit metatheory inconsistent with that candidate theory. And then suppose that, when judged from that perspective, the candidate theory is determined to be good or bad. There would be no reason to trust such a conclusion. For the conclusion might be due not to the empirical adequacy or failings of the theory under consideration, but rather to the conceptual inadequacy of the presumed metatheory.[10]

In sum, the plausibility of the computational theory of mind requires that a proper analysis of a candidate theory of

9. For example, it would be inconsistent simultaneously to claim the following three things: (i) as many do, that scienti.fic theories should be expressed from an entirely third-person, nonsubjective point of view; (ii) as an intrinsic fact about all computational processes, that genuine reference is possible only from a first-person, subjective vantage point ("first-person" from the perspective of the machine); and (iii) that the computational theory of mind is true. If one were to believe in the ineliminably first-person character of computational reference, and that human reference is a species of computational reference, then consistency would demand that such a theory be stated from a first-person point of view—since, by hypothesis, no other way of presenting the theory would refer.

10. Note that the situation is symmetric; reflexive inconsistencies can generate both false negatives and false positives.

computing must consider: (i) what computational theory of mind would be generated, in its terms; (ii) what form theories in general would take, on such a model of mind; (iii) what the candidate theory of computing in question would look like, when framed as such a theory; (iv) whether the resulting theory (of computing), so framed, would hold true of computation-in-the-wild; and (v) whether, if it did turn out to be true (i.e., empirically), mentation and theorizing would, by those lights, also be computational. *All this is required, for reflexive integrity*. To do these things, we need to understand A8 whether—and how—the theory could underwrite a theory of mind. Hence the cognitive criterion.

It is essential to understand, however, that the cognitive criterion requires only that we *understand* what form a computational theory of mind would take; it does not reflect any commitment to *accept* such a theory. In committing myself to honor the criterion, that is, I make no advance commitment to computationaIism's being true or false. I just want to know what it says.

None of this is to say that the content of the computational theory of mind is left open. Computationalism's fundamental thesis—that the mind is computational—is given substance by the first, empirical criterion. Computationalism, that is—at least as I read it—is not a theory-laden or "opaque" proposal, in the sense of framing or resting on a specific hypothesis about what computers are. Rather, it has more an ostensive or "transparent" character: that people (i.e., us) are computers in whatever way that computers (i.e., those things over there) are computers, or at least in whatever way *some* of those things are computers.[11]

11. The computational theory of mind does not claim that minds and computers are equivalent (in the sense that anything that is a mind is a computer, and vice versa). Rather, the idea is that minds are (at least) a *kind* of computer, and furthermore that the kind is *itself computationally characterized* (i.e., that the characteristic predicate on the restricted class of computers that are minds is itself framed in computational terms).

It follows that specific theoretical formulations of cognitivism (whether pro are con) are doubly contingent. Thus consider, on the positive side, Newell and Simon's (1976) popular "physical symbol system hypothesis," according to which human intelligence is claimed to consist of physical symbol manipulation; or Fodor's (1975, 1980) claim, that thinking consists of formal symbol manipulation; or—on the critical side—Dreyfus' (1992) assertion that cognitivism (as opposed to connectionism) requires the explicit manipulation of explicit symbols; or van Gelder's (1995) claim that computationalism is both false and misleading, deserving to be replaced by dynamical alternatives. Not only do these writers make a hypothetical statement about *people*, that they are physical, formal, or explicit symbol manipulators, respectively; they do so by making a hypothetical statement about *computers*, that they are in some essential or illuminating way characterizable in the same way. Because I take the latter claim to be as subservient to empirical adequacy as the former, there are two ways in which these writers could be wrong. In claiming that people are formal symbol manipulators, for example, Fodor would naturally be wrong if computers were formal symbol manipulators and people were not. But he would also be wrong, *while the computational theory of mind itself might still be true*, if computers were not formal symbol manipulators, either. Similarly, van Gelder's brief against computational theories of mind is vulnerable to his understanding of what computing is actually like. If, as I believe, computation-in-the-wild is not as he characterizes it, then the sting of his critique is entirely eliminated.

In sum, cognitive science is, like computer science, hostage to the foundational project:[12] of formulating a comprehensive,

12. Foundationalism is widely decried, these days—especially in social and critical discourses. Attempting a foundational reconstruction of the sort I am attempting here may therefore be discredited, by some, in advance. As suggested in Smith (1996), however, I do not believe that any of the arguments that have been raised against foundationalism (particularly: against the valorization of a small set of types or categories as

true, and intellectually satisfying theory of computing that honors all three criteria. No one of them is easy to meet.

3 Six Construals of Computing

Some might argue that we already know what computation is. That in turn breaks into two questions: (i) is there a story—an account that people think answers the question of what computing is or what computers are; and (ii) is that story right?

Regarding the first question, the answer is not *no*, but it is not a simple *yes*, either. More than one idea is at play in current theoretic discourse. Over the last thirty years I have found it convenient to distinguish seven **construals** of computation, each requiring its own analysis:

1. **Formal Symbol Manipulation (FSM):** the idea, derivative from a century's work in formal logic and metamathematics, of a machine manipulating symbolic or (at least potentially) meaningful expressions without regard to their interpretation or semantic content;
2. **Effective Computability (EC):** what can be done, and how hard it is to do it, mechanically, as it were, by an abstract analogue of a "mere machine";
3. **Execution of an algorithm (ALG) or rule-following (RF):** what is involved, and what behavior is thereby produced, in following a set of rules or instructions, such as when making dessert; A9
4. **Calculation of a function (FUN):** the behavior, when given as input an argument to a mathematical function, of producing as output the value of that function applied to that argument; A10

. . .

holding an unquestioned and/or uniquely privileged status) amounts to an argument against rigorously plumbing the depths of an intellectual subject matter. In this chapter, my use of the term 'foundational' should be taken as informal and, to an extent, lay (I am as committed as anyone to the fallacies and even dangers of master narratives, ideological inscription, and/or uniquely privileging any category or type). A11

5. **Digital State Machine (DSM):** the idea of an automaton with a finite, disjoint set of internally homogeneous machine states—as parodied in the "clunk, clunk, clunk" gait of a 1950s cartoon robot;
6. **Information Processing (IP):** what is involved in storing, manipulating, displaying, and otherwise trafficking in information, whatever information might be; and
7. **Physical Symbol Systems (PSS):** the idea, made famous by Newell and Simon (1976), that, somehow or other, computers interact with, and perhaps also are made of, symbols in a way that depends on their mutual physical embodiment.

These seven construals have formed the core of our thinking about computation over the last fifty years, but no claim is made that this list of six is exhaustive.[13] At least to date, however, it is these seven that have shouldered the lion's share of responsibility for framing the intellectual debate.

By far the most important step in getting to the heart of the foundational question, I believe, is to recognize that these seven construals are all conceptually distinct. In part because of their great familiarity (we have long since lost our innocence), and in part because "real" computers seem to exemplify more than one of them—including those often-imagined but seldom-seen Turing machines, complete with controllers, read-write heads, and indefinitely long tapes—it is sometimes uncritically thought that all seven can be viewed as rough synonyms, as if they were different ways of getting at the same thing. Indeed, this conflationary tendency is rampant in the literature, much of which moves around among them as if doing so were intellectually free. But that is a mistake. The supposition that any two of these construals amount to the same thing, let alone that all seven do, is simply false.

13. [The footnote that appeared in this place in the original paper has been reproduced here as the sidebar "Additional Construals" on p. …]

For example, consider the formal symbol manipulation construal (FSM). It explicitly characterizes computing in terms of a semantic or intentional aspect, if for no other reason than that without some such intentional character there would

Additional Construals

Especially as the boundaries between computer science and surrounding intellectual territory erode (itself a development predicted by this analysis; see §8), several ideas that originated in other fields are making their way into the center of computational theorizing as alternative conceptions of computing. At least three are important enough to be seen as construals in their own right (though the first is not usually assumed to have any direct connection with computing, and the latter two are not normally assumed to be quite as "low-level" or foundational as the primary seven):

8. **Dynamics (DYN):** the notion of a dynamical system, linear or non-linear, as popularized in discussions of attractors, turbulence, criticality, emergence, etc.;
9. **Interactive Agents (IA):** active agents enmeshed in an embedding environment, interacting and communicating with other agents (and perhaps also with people); and
10. **Self-organizing** or **Complex Adaptive Systems (CAS):** a notion—often associated with the Santa Fe Institute—of self-organizing systems that respond to their environment by adjusting their organization or structure, so as to survive and (perhaps even) prosper.

Additional construals may need to be added, over time. Moreover, there are even those who deny that computation has any ontologically distinct identity. Thus Agre (1997), for example, claims that computation should instead be methodologically individuated: A12

11. **Physical Implementation (PHY):** a methodological hypothesis that computation is not ontologically distinct, but rather that computational practice is human expertise in the physical or material implementation of (apparently arbitrary) systems.

be no warrant in calling it *symbol* manipulation.[14] The digital state machine construal (DSM), in contrast, makes no such reference to intentional properties. If a Lincoln-log contraption were digital but not symbolic, and a system manipulating continuous symbols were formal but not digital, they would be differentially counted as computational by the two construals. Not only do FSM and DSM *mean* different things, in other words; they (at least plausibly) have overlapping but distinct extensions.

The effective computability (EC) and algorithm execution (ALG) construals similarly differ on the crucial issue of semantics. Whereas the effective computability construal, at least in the hands of computer scientists, seems free of intentional connotation,[15] the idea of algorithm execution, as I have characterized it, seems not only to involve rules or recipes, which presumably do mean something, but also (*pace* Wittgenstein) to require some sort of understanding on the part of the agent producing the behavior. A12.5

Semantics is not the only open issue. It is similarly unclear whether the notions of "machine" and "taking an effective step" internal to the EC construal make fundamental reference to causal powers, material realization, or other concrete physical properties, or whether, as most current theoretical discussions suggest, effective computability should be taken as an entirely abstract mathematical notion. Again, if we do not understand this **mind-body problem for machines**, how can we expect computational metaphors to help us in the case of people?

There are still other differences among construals. They differ on whether they inherently focus on internal structure or external input/output, for example—i.e., on whether: (i) they treat computation as fundamentally *a way of being structured*

14. See note 22.

15. At least some logicians and philosophers, in contrast, do read the effective computability construal semantically. This difference is exactly the sort of question that needs to be disentangled and explained in the full analysis.

or constituted, so that surface or externally observable behavior is derivative; or whether (ii) the *having of a particular behavior* is the essential locus of being computational, with questions about how that is achieved left unspecified and uncared about. The formal symbol manipulation and digital state machine construals are of the former, structurally constitutional type; effective computability is of the latter, behavioral variety; algorithm execution appears to lie somewhere in the middle).

The construals also differ in the degree of attention and allegiance they have garnered in different disciplines. Formal symbol manipulation (FSM) has for many years been the conception of computing that is privileged in artificial intelligence and philosophy of mind, but it receives almost no attention in computer science. Theoretical computer science focuses primarily on the effective computability (EC) and algorithm (ALG) construals, whereas mathematicians, logicians, and most philosophers of logic and mathematics pay primary allegiance to the functional conception (FUN). Publicly, in contrast, it is surely the information processing (IP) construal that receives the major focus—being by far the most likely characterization of computation to appear in the Wall Street Journal, and the idea responsible for such popular slogans as "the information age" and "the information highway." A13

Not only must the seven construals be differentiated one from another; additional distinctions must be made within each one. Thus the idea of information processing (IP) needs to be broken down, in turn, into at least three sub-readings, depending on how 'information' is understood: (i) as a *lay* notion, dating from perhaps the nineteenth-century, of something like an abstract, publicly-accessible commodity, carrying a certain degree of autonomous authority; (ii) so-called "information theory," an at least seemingly semantics-free notion that originated with Shannon and Weaver (1949), spread out through much of cybernetics and communication theory, is A14 A15

implicated in Kolmogorov, Chaitin, and similar complexity measures, and has more recently been tied to notions of energy and, particularly, entropy; and (iii) the semantical notion of information advocated by Dretske (1981), Barwise and Perry (1983), Halpern (1987), and others, which in contrast to the second deals explicitly with semantic content and veridicality.

Clarifying all these issues, bringing the salient assumptions to the fore, showing where they agree and where they differ, tracing the roles they have played in the last forty years—questions like this must be part of any foundational reconstruction. But in a sense these issues are all secondary. For none has the bite of the second question raised at the beginning of the section: of whether any of the enumerated accounts is *right*.

Naturally, one has to say just what this question means—has to answer the question "Right of what?"—in order to avoid the superficial response: "Of course such and such a construal is right; that's how computation is *defined*!" This is where the empirical criterion takes hold. More seriously, I am prepared to argue for a much more radical conclusion, which we can dub as the first major result:[16]

C1. When subjected to the empirical demands of practice and the (reflexively mandated) conceptual demands of cognitive science, *all seven primary construals fail*—for deep, overlapping, but distinct, reasons.

4 Diagnosis I: General

What is the problem? Why do these theories all fail?

The answers come at many levels. In the next section I discuss some construal-specific problems. But a general thing can be said first. Throughout, the most profound difficulties A16 have to do with semantics. It is widely (if tacitly) recognized that computation is in one way or another a symbolic or representational or information-based or semantical—that is,

16. This numbering system (C1–C9) is used only for purposes of this chapter; it will not necessarily be used in A05.

as philosophers would say, an *intentional*—phenomenon.[17] Somehow or other, though in ways we do not yet understand, the states of a computer can model or simulate or represent or stand for or carry information about or signify other states in the world (or at least can be taken by people to do so). This semantical or intentional character of computation is betrayed by such phrases as *symbol* manipulation, *information* processing, programming *languages*, *knowledge representation*, *data*bases, and so on. Indeed, if computing were not intentional, it would be spectacular that so many intentional words of English systematically serve as technical terms in computer science.[18] Furthermore—and this is important to understand—it is the intentionality of the computational that motivates the cognitivist hypothesis. The only compelling reason to suppose A17 that we (or minds or intelligence) might be computers stems from the fact that we, too, deal with representations, symbols, meaning, information, and the like.[19]

17. Although the term 'intentional' is primarily philosophical, there are many philosophers, to say nothing of some computer and cognitive scientists, who would deny that computation is an intentional phenomenon. Reasons vary, but the most common goes something like this: (i) that computation is both *syntactic* and *formal*, where 'formal' means "independent of semantics"; and (ii) that intentionality has fundamentally to do with semantics; and therefore (iii) that computation is thereby not intentional. I believe this is wrong, both empirically (that computation is purely syntactic) and conceptually (that being syntactic is a way of not being intentional); I also disagree that being intentional has *only* to do with semantics, which the denial requires. See note 22.

18. Thus computer science's use of (the English words) 'language,' 'representation,' 'data,' etc. is analogous to physics' use of 'work,' 'force,' 'energy,' etc.—as opposed to its use of 'charm.' That is, it reflects a commitment to do scientific justice to the center of gravity of the word's natural meaning, rather than being mere whimsical fancy.

19. Physically, we and (at least contemporary) computers are not very much alike—though it must be said that one of the appeals, to some people at least, of the self-organizing or complex-adaptive-system construal (CAS) is its prospect of providing a naturalistically palatable and

For someone with cognitivist leanings, therefore—as opposed, say, to an eliminativist materialist, or to some types of connectionist—it is natural to expect that a comprehensive theory of computation will have to focus on its semantical aspects. This raises problems enough. Consider just the issue of representation. In order to meet the first criterion, of empirical adequacy, a successful candidate will have to make sense of the myriad kinds of representation that saturate practical systems—from bit maps and images to knowledge representations and databases; from high-speed caches to long-term backup tapes; from low-level finite-element models used in simulation to high-level analytic descriptions supporting reasoning and inference; from text to graphics to audio to video to virtual reality. As well as being vast in scope, it will also have to combine decisive theoretical bite with exquisite resolution, in order to distinguish: models from implementations; analyses from simulations; and virtual machines at one level of abstraction from virtual machines at another level of abstraction, in terms of which the former may be implemented.

To meet the second, conceptual criterion, moreover, any account of this profusion of representational practice must be grounded on, or at least defined in terms of, a theory of semantics or content, partly in order for the concomitant psychological theory to avoid vacuity or circularity, and partly so that even the computational part of the theory meet a minimal kind of naturalistic criterion: that we understand how computation is part of the natural world. This is made all the more difficult by the fact that the word 'semantics' is used in an A18 incredible variety of senses across the range of the intentional sciences. Indeed, in my experience it is virtually impossible, from any one location within that range, to understand the full significance of the term, so disparate is that practice *in toto*.

nonintentional but nevertheless specific way of discriminating people-cum-computers (and perhaps higher animals) from arbitrary physical devices.

Genuine theories of content,[20] moreover—of what it is that makes a given symbol or structure or patch of the world be *about* or *oriented towards* some other entity or structure or patch—are notoriously hard to come by.[21] Some putatively foundational construals of computation are implicitly defined in terms of just such a background theory of semantics, but do not explain what semantics is, and thus fail the second, conceptual criterion. This includes the first, formal symbol manipulation construal so favored (and disparaged!) in the cognitive sciences, in spite of its superficial formulation as being "independent of semantics."[22] Other construals, such as those that view computation as the behavior of discrete automata—and also, I will argue below, even if this is far from immedi-

20. In computer science, to take a salient example, the term "the semantics of α", where α is an expression or construct in a programming language, means approximately the following: the topological (as opposed to geometrical) temporal profile of the behavior to which execution of this program fragment gives rise. By 'topological' I mean that the overall temporal order of events is dictated, but that their absolute or metric time-structure (e.g., exactly how fast the program runs) is not. As a result, a program can usually be sped up, either by adjusting the code or running it on a faster processor, without, as is said, "changing the semantics." A18.5

21. Best known are Dretske's semantic theory of information (1981), which has more generally given rise to what is known as "indicator semantics"; Fodor's "asymmetrical-dependence" theory (1987); and Millikan's "teleosemantics" or "biosemantics" (1984, 1989). For comparison among these alternatives see, e.g., Fodor (1984) and Millikan (1990).

22. Because formal symbol manipulation is usually defined as "manipulation of symbols independent of their interpretation", some people believe that the formal symbol manipulation construal of computation does not rest on a theory of semantics. But that is simply an elementary, though apparently common, conceptual mistake. As discussed further in §5, the "independence of semantics" postulated as essential to the formal symbol construal is independence at the level of the phenomenon; it is a claim about how symbol manipulation *works*. Or so at least I believe, based on many years of investigating what practitioners are actually committed to (whether it is *true*—i.e., holds of computation-in-the-wild—is a separate issue). The intuition is simple enough: that semantic properties,

ately evident, the recursion-theoretic one that describes such behavior as the calculation of effective functions—fail to deal with computation's semantical aspect at all, in spite of sometimes using semantical vocabulary, and so fail the first, empirical criterion. In the end, one is inexorably driven to a second major conclusion:[23]

C2. In spite of the advance press, especially from cognitivist quarters, computer science, far from supplying the answers to fundamental intentional mysteries, must, like cognitive science, await the development of a satisfying theory of semantics and intentionality.

. . .

such as referring to the Sphinx, or being true, are not of the right sort to do effective work—so they cannot be the sort of property in virtue of the manifestation of which computers *run*.

At issue in the present discussion, in contrast, is a more logical form of independence, *at the level of the theory* (or, perhaps, to put it more ontologically and less epistemically, independence at the level of the *types*). Here the formal symbol manipulation construal is as dependent on semantics as it is possible to be: *it is defined in terms of it*. And (as the parent of any teenager knows) defining yourself in opposition to something is not ultimately a successful way of achieving independence. Symbols must have a semantics, in other words (have an actual interpretation, or be interpretable, or whatever), in order for there to be something substantive for their formal manipulation to proceed independently of. Without a semantic character to be kept crucially in the wings, the formal symbol manipulation construal would collapse in vacuity—would degenerate into something like "the manipulation of structure" or, as I put it in AOS, "stuff manipulation"—i.e., materialism.

23. As suggested in the preceding footnote, philosophers are less likely than computer scientists to expect a theory of computation to be, or to supply, a theory of intentionality. That is, they would not expect the metatheoretic structure to be as expected by most computer scientists and artificial intelligence researchers—namely, to have a theory of intentionality *rest on* a theory of computation. But that does not mean they would necessarily agree with the opposite, which I am arguing here: that a theory of computation will have to rest on a theory of intentionality. Many philosophers seem to think that a theory of computation can be *independently* of a theory of intentionality. Clearly, I do not believe this is correct.

5 Diagnosis II: Specific

So none of the seven construals provides an account of semantics. Since I take computation to be semantic (not just by assumption, but as an unavoidable lesson from empirical investigation), that means they fail as theories of computation, as well (i.e., C2 implies C1). And that is just the beginning of the problems. All seven also fail for detailed structural reasons—different reasons per construal, but reasons that add up, overall, to a remarkably coherent overall picture.

In this section I summarize just a few of the problems, to convey a flavor of what is going on. In each case, to put this in context, these results emerge from a general effort, in the main investigation, to explicate, for each construal:

1. What the construal says or comes to—what claim it makes about what it is to be a computer;
2. Where it derives from, historically;
3. Why it has been held;
4. What's right about it—what insights it gets at;
5. What is wrong with it, conceptually, empirically, and explanatorily;
6. Why it must ultimately be replaced; and
7. What about it should nevertheless be retained in a "successor," more adequate account.

5a Formal Symbol Manipulation

The FSM construal has a distinctly **antisemantical** flavor, owing to its claim that computation is the "manipulation of symbols independent of their semantics." On analysis, it turns out to be motivated by two entirely different, ultimately incompatible, independence intuitions. The first motivation is at the level of the theory, and is reminiscent of a reductionist desire for a "semantics-free" account. It takes the FSM thesis as a claim that computation can be *described* or *analyzed* in a semantics- A19

free way. If that were true, so the argument goes, that would go some distance towards naturalizing intentionality (as Haugeland says, "... if you take care of the syntax, the semantics will take care of itself").†

There is a second motivating intuition, different in character, that holds at the level of the phenomenon. Here the idea is simply the familiar observation that intentional phenomena, such as reasoning, hoping, or dreaming, carry on in relative independence of their subject matters or referents. Reference and truth, it is recognized, are just not the sorts of properties that can play a causal role in engendering behavior—essentially because they involve some sort of relational coordination with things that are *too far away* (in some relevant aspect) to make a difference. This relational characteristic of intentionality—something I call **semantic disconnection**—is such a deep aspect of intentional phenomena that it is hard to imagine its being false. Without it, fantasy lives would be metaphysically banned; you would not be able to think about continental drift without bringing the tectonic plates along with you. A20

For discussion, I label the two readings of the formal symbol manipulation construal *conceptual* and *ontological*, respectively.[24] The ontological reading is natural, familiar, and based on a deep insight. But it is too narrow. Many counterexamples can be cited against it, though space does not permit rehearsing them here.[25] Instead, to get to the heart of the matter, it helps to highlight a distinction between two kinds of "boundary" thought to be relevant or essential—indeed, often assumed *a priori*—in the analysis of computers and other intentional systems:

† Haugeland (1981a, 23); see also Haugeland (1985).

24. It can be tempting to think of the two readings as corresponding to intensional and extensional readings of the phrase "independent of semantics"—but that is not strictly correct. See A05.

25. See A05 Volume II.

1. PHYSICAL: A physical boundary between the system and its surrounding environment—between "inside" and "outside"; and
2. SEMANTIC: A semantic "boundary" between symbols and their referents.

In terms of these two distinctions, the ontological reading of the FSM construal can be understood as presuming the following two theses:

1. ALIGNMENT: That the physical and semantic boundaries line up, with all the symbols inside, all the referents outside; and
2. ISOLATION: That this allegedly aligned boundary is a barrier or gulf across which various forms of dependence (causal, logical, explanatory) do not reach.

The fundamental idea underlying the FSM thesis, that is, is that a barrier of this double allegedly-aligned sort can be drawn around a computer, separating a pristine inner world of symbols—a private kingdom of ratiocination or thought, as it were—understood both to work (ontologically) and to be analyzable (theoretically) in isolation, without distracting influence from the messy, unpredictable exterior.

It turns out, in a way that is not ultimately surprising, that the traditional examples motivating the FSM construal, such as theorem proving in formal logic, meet this complex condition. First, they involve internal symbols designating external situations, thereby satisfying ALIGNMENT: (internal) databases representing (external) employee salaries, (internal) differential equations modeling the (external) perihelion of Mercury, (internal) first-order axioms designating (external) Platonic numbers or purely abstract sets, and so on. Second, especially in the paradigmatic examples of formal axiomatizations of arithmetic and proof systems of first-order logic (and,

even more especially, when those systems are understood in classical, especially model-theoretic, guise), the system is assumed to exhibit the requisite lack of interaction between the (internal) syntactic proof system and the (external, perhaps model-theoretic) interpretation, satisfying ISOLATION. In conjunction, the two assumptions allow the familiar two-part picture of a formal system to be held: a locally contained syntactic system, on the one hand, consisting of symbols or formulae in close causal intimacy with a proof-theoretic inference regimen; and a remote realm of numbers or sets or "ur-elements," in which the symbols or formulae are interpreted, on the other. It is because the formality condition relies on both theses together that the classical picture takes computation to consist exclusively of symbol-symbol transformations, carried on entirely within the confines of a machine.

The first—and easier—challenge to the antisemantical thesis comes when one retains the first ALIGNMENT assumption, of coincident boundaries, but relaxes the second ISOLATION claim, of no interaction. This is the classical realm of input/output, home of the familiar notion of a transducer. And it is here that one encounters the most familiar challenges to the FSM construal, such as the "robotic" and "system" replies to Searle's (1980) Chinese room argument, and Harnad's (1990) "Total Turing Test" as a measure of intelligence. Thus imagine a traditional perception system—for example, one that on encounter with a mountain lion constructs a symbolic representation of the form MOUNTAIN-LION-043. There is interaction (and dependence) from external world to internal representation. By the same token, an actuator system, such as one that would allow a robot to respond to a symbol of the form CROSS-THE-STREET by moving from one side of the road to the other, violates the independence assumption in the other direction, from internal representation to external world.

Note, in spite of this interaction, and the consequent viola-

tion of ISOLATION, that ALIGNMENT is preserved in both cases: the transducer is imagined to mediate between an internal symbol and an external referent. Nevertheless, the violation of ISOLATION is already enough to defeat the formality condition. This is why transducers and computation are widely recognized to be uneasy bedfellows, at least when formality is at issue. It is also why, if one rests the critique at this point, defenders of the antisemantical construal are tempted to wonder, given that the operations of transducers violate *formality*, whether they should perhaps be counted as *not being computational*.[26] Given the increasing role of environmental interaction within computational practice, it is not at all clear that this would be possible, without violating the condition of empirical adequacy embraced at the outset. For our purposes it doesn't ultimately matter, however, because the critique is only halfway done.

More devastating to the FSM construal are examples that challenge the ALIGNMENT thesis. It turns out, on analysis, that far from lining up on top of each other, real-world computer systems' physical and semantic boundaries *cross-cut*, in rich and productive interplay. It is not just that computers are involved in an engaged, participatory way with *external* subject matters, in other words, as suggested by some recent "situated" theorists. They are participatorily engaged in the world *as a whole*—in a world that indiscriminately includes themselves, their own internal states and processes. This integrated participatory involvement, blind to any *a priori* subject-world distinction, and concomitantly intentionally directed towards both internally and externally exemplified states of affairs, is not only architecturally essential, but is also critical, when the time comes, in establishing and grounding a system's intentional capacities.

26. Thus Devitt (1991) restricts the computational thesis to what he calls "thought-thought" (t-t) transactions; for him output (t-o) and input (i-t) transactions count as non-computational.

From a purely structural point of view, four types of case are required to demonstrate this non-alignment of boundaries: (i) where a symbol and referent are both internal; (ii) where a symbol is internal and its referent external; (iii) where symbol and referent are both external; and (iv) where symbol is external and referent internal. The first is exemplified in cases of quotation, meta-structural designation, window systems, e-mail, compilers, loaders, network routers, and at least arguably A21 all programs (as opposed, say, to databases). The second, of internal symbols with external referents, can be considered as something of a theoretical (though not necessarily empirical) default, as for example when one reflects on the sun's setting over Georgian Bay (to use a human example), or when a computer database represents the usage pattern of a set of university classrooms. The third and fourth are neither more nor less than a description of ordinary written text, public writing, etc.—to say nothing of pictures, sketches, conversations, and the whole panoply of other forms of external representation. Relative to any particular system, they are distinguished by whether the subject matters of those external representations are similarly external, or are internal. The familiar red skull-and-cross-bones signifying radioactivity is external to both man and machine, and also denotes something external to man and machine, and thus belongs to the third category. To a computer or person involved, on the other hand, an account of how they work (psychoanalysis of person or machine, as it were, to say nothing of logic diagrams, instruction manuals, etc.) is an example of the fourth.

By itself, violating ALIGNMENT is not enough to defeat formality. What it does accomplish, however, is to radically undermine ISOLATION's plausibility. In particular, the anti-semantical thesis constitutive of the FSM construal is challenged not only because these examples show that the physical and semantic boundaries cross-cut, thereby undermining

the ALIGNMENT assumption, but because they illustrate the presence, indeed the prevalence, of effective traffic across both boundaries—between and among all the various categories in question—thereby negating ISOLATION.

And this negation of ISOLATION, in turn, shows up, for what it is, the common suggestion that transducers, because of violating the antisemantical thesis, should be ruled "out of court"—i.e., should be taken as non-computational, à la Devitt (1991).[27] It should be clear that this maneuver is ill-advised; even a bit of a cop-out. For consider what a proponent of such a move must face up to, when confronted with boundary non-alignment. *The notion of a transducer must be split in two.* In order to retain an antisemantical (FSM) construal of computing, someone interested in transducers would have to distinguish:

1. **Physical transducers**, for operations or modules that cross or mediate between the inside and outside of a system; and
2. **Semantic transducers**, for operations or modules that mediate or "cross" between symbols and their referents.

And it is this bifurcation, finally, that irrevocably defeats the antisemantical claim. For the only remotely plausible notion of transducer, in practice, is the physical one. That is what we think of when we imagine vision, touch, smell, articulation, wheels, muscles, and the like: systems that mediate between the internals of a system and the "outside" world. Transducers, that is, at least in informal imagination of practitioners, are for connecting systems to their (physical) environments.[28] What poses a challenge to the formal (antisemantical) symbol ma-

27. See the preceding note.

28. This statement must be understood within the context of computer science, cognitive science, and the philosophy of mind. It is telling that the term 'transducer' is used completely differently in engineering and biology (its natural home), to signify mechanisms that mediate changes in *medium*, not that cross *either* the inside/outside *or* the symbol/referent boundary. A22

nipulation construal of computation, on the other hand, are *semantic* transducers: those aspects of a system that involve trading between occurrent states of affairs, on the one hand, and representations of them, on the other. Antisemantics is challenged as much by disquotation as by driving around.

As a result, the only way to retain the ontological version of the FSM construal is to disallow (i.e., count as non-computational) the operations of semantic transducers. *But that is absurd!* It makes it clear, ultimately, that distinguishing that subset of computation which satisfies the ontological version of the antisemantical claim is not only unmotivated, solving the problem by fiat (making it uninteresting), but is a spectacularly infeasible way to draw and quarter any actual, real-life system. For no one who has ever built a computational system has ever found any reason to bracket reference-crossing operations, or A23 to treat them as a distinct type. Not only that; think of how many different kinds of examples of semantic transducer one can imagine: counting, array indexing, e-mail, disquotation, A24 error-correction circuits, linkers, loaders, simple instructions, database access routines, pointers, reflection principles in logic, index operations into matrices, most Lisp primitives, and the like. Furthermore, to *define* a species of transducer in this semantical way, and then to remove them from consideration as not being genuinely computational, would make computation (minus the transducers) antisemantical *tautologically*. It would no longer be an interesting claim on the world that computation was antisemantical—an insight into how things are. Instead, the word 'computation' would simply be shorthand for antisemantical symbol manipulation. The question would be whether anything interesting was in this named class—and, in particular, whether this conception of computation captured the essential regularities underlying practice. And we have already seen the answer to that: it is *no*.

In sum, introducing a notion of a semantical transducer

solves the problem tautologically, cuts the subject matter at an unnatural joint, and fails to reconstruct practice. That is quite a lot to have going against it.

Furthermore, to up the ante on the whole investigation, not only are these cases of "semantic transduction" all perfectly well-behaved; they even seem, intuitively, to be as "formal" as any other kind of operation. If that is so, then those systems either are not formal, after all, *or else the word 'formal' has never meant independence of syntax and semantics in the way that the* FSM *construal claims*. Either way, the ontological construal does not survive.

Though it has been framed negatively, we can summarize this result in positive terms:

C3. Rather than consisting of an internal world of symbols separated from an external realm of referents, as imagined in the FSM construal, real-world computational processes are *participatory*: they involve complex paths of causal interaction between and among symbols and referents, both internal and external, cross-coupled in complex configurations.

5b Effective Computability

Although different in detail, the arguments against the other primary construals are similar in style. In each case, I have tried to develop a staged series of counterexamples, not simply to show the construal false, but to serve as strong enough intuition pumps on which to base a positive alternative. In other words, the point is not critique, but deconstruction en route to reconstruction. Space permits a few words about just one other construal: effective computability—the idea that underwrites recursion theory, complexity theory, and, as I have said, the official (mathematical) "Theory of Computation." A25

Note, for starters—as mentioned earlier—that whereas

the first, FSM construal is predominant in artificial intelligence, cognitive science, and philosophy of mind, it is the second, effective computability (EC) construal, in contrast, that underlies most theoretical and practical computer science.

Fundamentally, it is widely agreed, the theory of effective computability focuses on "what can be done by a mechanism." But two conceptual problems have clouded its proper appreciation. First, in spite of its subject matter, it is almost always characterized *abstractly*, as if it were a branch of mathematics. Second, it is imagined to be a theory defined over (for example) the numbers. Specifically, the marks on the tape of the paradigmatic Turing machine are viewed as *representations* or *encodings*—representations, in general, or at least in the first instance, of numbers, functions, or other Turing machines.

In almost exact contrast to the received view, I argue two things. First, I claim that the theory of effective computability is fundamentally a theory about the *physical* nature of patches of the world. In underlying character, I believe, it is no more "mathematical" than anything else in physics—even if we use mathematical structures to model that physical reality. Second—and this is sure to be contentious—I argue that recursion theory is fundamentally a *theory of marks*. More specifically, rather than taking the marks on the tape to be representations of numbers, as has universally been assumed in the theoretical tradition, I defend the following claim: A25.5

C4. The representation relation for Turing machines, alleged to run from marks to numbers, in fact runs the other way, from numbers to marks. The truth is 180° off what we have all been led to believe.

All sorts of evidence are cited in defense of this non-standard claim. For example:

1. Unless one understands it this way, one can solve the halting problem;[29]

29. See A05: Volume III.

2. An analysis of history, through Turing's paper and subsequent work, especially including the development of the universal Turing machine, shows how and why the representation relation was inadvertently turned upside down in this way;
3. The analysis makes sense of a number of otherwise-inexplicable practices, including, among other examples: (i) the use of the word "semantics" in practicing computer science to signify the behavior engendered by running a program,[30] (ii) the rising popularity of such conceptual tools as Girard's linear logic, and (iii) the close association between theoretical computer science and constructive mathematics. A26

It follows from this analysis that all use of semantical vocabulary in the "official" Theory of Computation is metatheoretic. As a result, *the so-called (mathematical) "Theory of Computation" is not a theory of intentional phenomena*—in the sense that it is not a theory that deals with its subject matter *as* an intentional phenomena.

In this way the layers of irony multiply. Whereas the FSM construal fails to meet its own criterion, of being "defined independent of semantics," this second construal *does* meet (at least the conceptual reading of) that first-construal condition. Exactly in achieving that success, however, the recursion-theoretic tradition thereby fails. For computation, as was said above, and as I am prepared to argue, *is* (empirically) an intentional phenomenon. So the EC construal achieves naturalistic palatability at the expense of being about the wrong subject matter.

We are thus led inexorably to the following very strong conclusion: what goes by the name "Theory of Computation" fails not because it makes false claims about computation, but because *it is not a theory of computation at all*.[31, 32]

30. See note 20.

31. The fact that it is not a theory of computing does not entail that it

In sum, the longer analysis ultimately leads to a recommendation that we redraw a substantial portion of our intellectual map. What has been called a "Theory of Computation" is in fact a general theory of the physical world—specifically, a theory of how hard it is, and what is required, for patches of the world in one physical configuration to change into another physical configuration. It applies to *all* physical entities, not just to computers. It is no more mathematical than the rest of physics, and thus it should be joined with physics—because in a sense it *is* physics.

We can put this result more positively. Though falsely (and misleadingly) labeled, the mathematical Theory of Computation has been a spectacular achievement, of which the twentieth-century should be proud. Indeed, this is important enough that we can label it as the fifth major result:

C5. Though not yet so recognized, the mathematical theory based on recursion theory, Turing machines, complexity analyses, and the like—widely known as the "Theory of Computation"—is neither more nor less than a *mathematical theory of causality*.

6 Method

Similarly strong conclusions can be arrived at by pursuing each of the other construals. Indeed, the conclusion from the

does not *apply* to computers, of course. All it means is that, in that application, it is not a theory of them *as computers*.

32. That the so-called theory of computation fails as a theory of computation because it does not deal with computation's intentionality is a result that should be agreed even by someone (e.g., Searle) who believes that computation's intentionality is inherently derivative. I myself do not believe that computation's intentionality is inherently derivative, as it happens, but even those who think it is must admit that it is still an intentional phenomenon of some sort. For *derivative* does not mean *fake* or *false*. If "derivatively intentional" is not taken to be a substantive constraint, then we are owed (e.g., by Searle) an account of what *does* characterize computation.

analysis of the digital state machine construal (DSM)—that computation-in-the-wild is not digital—is, if anything, even more consequential than the results derived from either the FSM or the EC critiques. Rather than go into them here, however, I instead want to say a word about method—specifically, about *formality*. For a potent theme underlies all seven critiques: that part of what has blinded us to the true nature of computation has to do with the often pretheoretic assumption that *computation and/or computers are formal*.

In one way or another, no matter what construal they pledge allegiance to, just about everyone thinks that computers are formal—that they manipulate symbols formally, that programs specify formal procedures, that data structures are a kind of formalism, that computational phenomena are uniquely suited for analysis by formal methods, and so on. In fact the computer is often viewed as the crowning achievement of an entire "formal tradition"—an intellectual orientation, reaching back through Galileo to Plato, that was epitomized in the twentieth century in the logic and metamathematics of Frege, Russell, Whitehead, Carnap, and Turing, among others.

This history would suggest that formality is an essential aspect of computation. But since the outset, I have not believed that this is necessarily so. For one thing, it has never been clear what the allegiance to formality is an allegiance to. It is not as if "formal" is a technical or theory-internal predicate, after all. People may believe that developing an idea means formalizing it, and that programming languages are formal languages, and that theorem provers operate on formal axioms—but few write "*formal(x)*" in their daily equations. Moreover, a raft of different meanings and connotations of this problematic term lies just below the surface. Far from hurting, this apparent ambiguity has helped to cement popular consensus. Freed of the need to be strictly defined ('*formal*' is not a formal predicate), formality has been able to serve as a lightning rod for a clus-

ter of ontological assumptions, methodological commitments, and social and historical biases.

Because it remains tacit, cuts deep, has important historical roots, and permeates practice, formality has been an ideal foil, over the years, with which to investigate computation.

Almost a dozen different readings of 'formal' can be gleaned from informal usage: *precise*, *abstract*, *syntactic*, *mathematical*, *explicit*, *digital*, *a-contextual*, *non-semantic*, among others.[33] They are alike in foisting recalcitrant theoretical issues onto center stage. Consider explicitness, for example, of the sort that might explain such a sentence as "for theoretical purposes we should lay out our tacit assumptions in a formal representation." Not only have implicitness and explicitness stubbornly resisted theoretical analysis, but both notions are parasitic on something else we do not understand: general representation.[34] Or consider "a-contextual." Where is an overall theory of context in terms of which to understand what it would be to say of something (a logical representation, say) that it was not contextually dependent? A27

Considerations like this suggest that particular readings of formality can be most helpfully pursued within the context of the general theoretical edifices that have been constructed (more or less explicitly) in their terms. Five are particularly important: A28

33. At one stage I asked a number of people what they thought "formal" meant—not just computer scientists, but also mathematicians, physicists, sociologists, etc. It was clear from the replies that the term has very different connotations in different fields. Some mathematicians and logicians, for example, take it to be pejorative, in contrast to the majority of theoretical computer scientists, for whom it has an almost diametrically opposed positive connotation.

34. On its own, an eggplant cannot be either formal or explicit, at least not in its ordinary culinary role, since in that role it is not a representation at all. In fact the only way to make sense of calling something non-representational explicit is as short-hand for saying that it is explicitly represented (e.g., calling eggplant an explicit ingredient of moussaka as a way of saying that the recipe for moussaka mentions eggplant explicitly).

1. The *antisemantical* reading mentioned above: the idea that a symbolic structure (representation, language, program, symbol system, etc.) is formal just in case it is manipulated *independent of its semantics*. Paradigmatic cases include so-called formal logic, in which it is assumed that a theorem—such as MORTAL(SOCRATES)—is derived by an automatic inference regimen without regard to the reference, truth, or even meaning of any of its premises.
2. A closely allied grammatical or *syntactic* reading, illustrated in such a sentence as "inference rules are defined in terms of the *formal* properties of expressions." (Note that whereas the antisemantical reading is negatively characterized, this syntactic one has a positive sense.)
3. A reading meaning something like *determinate* or *well-defined*—that is, as ruling out all ambiguity and vagueness. This construal turns out to be related to a variant of the computationally familiar notion of digitality or discreteness. A29
4. A construal of "formal" as essentially equivalent to *mathematical*.
5. A reading that cross-cuts the other four: formality as applied to analyses or *methods*, perhaps with a derivative ontological implication that some subject matters (including computation?) are uniquely suited to such analytic techniques.

The first two (antisemantical and syntactic) are often treated as conceptually equivalent, but to do that is to assume that a system's syntactic and semantic properties are *necessarily disjoint*—which is almost certainly false. The relationship between the third (determinate) reading and digitality does not have so much to do with what Haugeland (1982) calls "first-order digitality": the ordinary assumption that a system's

states can be partitioned into a determinate set, such as that its future behavior or essence stems solely from membership in one element of that set, without any ambiguity or matter of degree. Rather, vagueness and indefiniteness (as opposed to simple continuity) are excluded by a *second-order* form of digitality—digitality at the level of concepts, in the sense of there being a binary "yes/no" fact of the matter about whether any given situation falls under (or is correctly classified in terms of) the given concept. And finally, the fourth view—that to be formal has something to do with being mathematical, or at least with being mathematically characterizable—occupies something of an ontological middle-realm between the subject-matter orientation of the first three and the methodological orientation of the fifth.

The ultimate moral for computer and cognitive science, I argue, is similar to the claim made earlier about the seven construals: *not one of these readings of 'formal' correctly applies to the computational case*. It can never be absolutely proved that computation is not formal, of course, given that the notion of formality is not determinately tied down. What I am prepared to argue (and do argue in the full analysis) is the following: no standard construal of formality, including any of those enumerated above, is both (i) substantive and (ii) true of extant computational practice. Some readings reduce to vacuity, or to no more than physical realizability; others break down in internal contradiction; others survive the test of being substantial, but are demonstrably false of current systems. In the end, one is forced to a sixth major conclusion: A30

C6. Computation is not formal.

It is an incredible historical irony: the computer, darling child of the formal tradition, has outstripped the bounds of the very tradition that gave rise to it.

7 The Ontological Wall

Where does all this leave us? It begins to change the character of the project. It is perhaps best described in personal A31 terms. Over time, investigations of the sort described above, and consideration of the conclusions reached through them, convinced me that none of the reigning theories or construals of computation, nor any of the reigning methodological attitudes towards computation, will *ever* lead to an analysis strong enough to meet the three criteria laid down at the outset.

It was not always that way. For the first twenty years of the investigation I remained:

1. In awe of the depth, texture, scope, pluck, and impact of computational practice;
2. Critical of the inadequate state of the current theoretical art;
3. Convinced that a formal methodological stance stood in the way of getting to the heart of the computational question; and
4. Sure in my belief that what was needed, above all else, was a *non-formal*—i.e., situated, embodied, embedded, indexical, critical, reflexive, all sorts of other things (it changed, over the years)—theory of representation and semantics, in terms of which to reconstruct an adequate conception of computing.

In line with this metatheoretic attitude, as the discussion this far will have suggested, I kept semantical and representational issues in primary theoretical focus. Since, as indicated in the last section, the official "Theory of Computation," derived from recursion and complexity theory, pays no attention to such intentional problems, to strike even this much of a semantical stance was to part company with the center of gravity of the received theoretical tradition.

You might think that this would be conclusion enough.

And yet, in spite of the importance and magnitude of these intentional difficulties, and in spite of the detailed conclusions suggested above, I have gradually come to believe something much more sobering: a conclusion that, although not as precisely stated as the foregoing, is if anything even more consequential:

c7. The most serious problems standing in the way of developing an adequate theory of computation are as much *ontological* as *semantical*.

It is not that computation's semantic problems go away; they remain as challenging as ever. It is just that they are joined—on center stage, as it were—by even more demanding problems of ontology.

Except that to say "joined" is misleading, as if it were a matter of simple addition—as if now there were two problems on the table, whereas before there had been just one. No such luck. The two issues (representation and ontology) are inextricably entangled—a fact of obstinate theoretical and metatheoretical consequence.

A methodological consequence will illustrate the problem. Especially within the analytic tradition (by which I mean to include not just analytic philosophy, e.g., of language and mind, but most of modern science as well, complete with its formal/mathematical methods), it is traditional to analyze semantical or intentional systems, such as computers or people, under the following presupposition: (i) that one can parse or register the A32 relevant theoretical situation in advance into a set of objects, properties, types, relations, equivalence classes, and so on (e.g., into people, heads, sentences, data structures, real-world referents, etc.)—as if this were theoretically innocuous—and then (ii), with that ontological parse in hand, go on to proclaim this or that or the other thing as an empirically justified result. Thus for example one might describe a mail-delivering robot

by first describing an environment of offices, hallways, people, staircases, litter, and the like, through which the robot is supposed to navigate, and then, taking this characterization of its context as given, ask how or whether the creature represents routes, say, or offices, or the location of mail delivery stations.

If one adopts a reflexively critical point of view, however, as I have systematically been led to do, one is led inexorably to the following conclusion: that, in that allegedly innocent pretheoretical "set-up" stage, one is liable, even if unwittingly, to project so many presuppositions, biases, and advance "clues" about the "answer," and in general to so thoroughly prefigure the target situation, without either apparent or genuine justification, that *one cannot, or at least should not, take any of the subsequent "analysis" terribly seriously*. It is a general problem that I have elsewhere labeled *preemptive registration*.[35] It is A33 problematic not just because it rejects standard analyses, but because it seems to shut all inquiry down. What else can one do, after all? How can one not parse the situation in advance (since it will hardly do to merely whistle and walk away)? And if, undaunted, one were to go ahead and parse it anyway, what kind of story could possibly serve as a justification? It seems that any conceivable form of defense would devolve into another instance of the same problem.

In sum, the experience is less one of facing an ontological challenge than of running up against an **ontological wall**. Perhaps not of slamming into it, at least in my own case; recognition dawned slowly. But neither is the encounter exactly gentle. It is difficult to exaggerate the sense of frustration that can come, once the conceptual fog begins to clear, from seeing one's theoretical progress blocked by what seems for all the world to be an insurmountable metaphysical obstacle.

Like many of the prior claims I have made, such as that all extant theories of computation are inadequate to reconstruct practice, or that no adequate conception of computing is for-

35. Smith (in press). «??»

mal, this last claim, that theoretical progress is stymied for lack of an adequate theory of ontology, is a strong statement, in need of correspondingly strong defense. Providing that defense is one of the main goals of AOS. In my judgment, to make it perfectly plain, despite the progress that has been made so far, and despite the recommended adjustments reached in the course of the seven specific analyses enumerated above, we are not going to get to the heart of computation, representation, cognition, information, semantics, or intentionality, until the ontological wall is scaled, penetrated, dismantled, or in some other way defused.

One reaction to the wall might be depression. Fortunately, however, the prospects are not so bleak. For starters, there is some solace in company. It is perfectly evident, once one raises one's head from the specifically computational situation and looks around, that computer scientists, cognitive scientists, and artificial intelligence researchers are not the only ones running up against severe ontological challenges. Similar conclusions are being reported from many other quarters. The words are different, and the perspectives complementary, but the underlying phenomena are the same.

Perhaps the most obvious fellow travelers are literary critics, anthropologists, and other social theorists, vexed by what A34 analytic categories to use in understanding people or cultures that, by such writers' own admission, comprehend and constitute the world using concepts alien to the theorists' own. What makes the problem particularly obvious, in these cases, is the potential for **conceptual clash** between theorist's and subject's world view—a clash that can easily seem paralyzing. One's own categories are hard to justify, and reek of imperialism; it is at best presumptuous, and at worst impossible, to try to adopt the categories of one's subjects; and it is manifestly impossible to work with no concepts at all. So it is unclear how, or even whether, to proceed.

But conceptual clash, at least outright conceptual clash, is not the only form in which the ontological problem presents itself. Consider the burgeoning interest in self-organizing and complex systems mentioned earlier, currently coalescing in a somewhat renegade subdiscipline at the intersection of dynamics, theoretical biology, and artificial life. This community debates the "emergence of organization," the units on which selection operates, the structure of self-organizing systems, the smoothness or roughness of fitness landscapes, and the like. In spite of being disciplinarily constituting, however, these discussions are conducted in the absence of adequate theories of what organization is, of what a "unit" consist in, of how "entities" arise (as opposed to how they survive), of how it is determined what predicates should figure in characterizing a fitness landscape as rough or smooth, and so on. The ontological lack is to some extent recognized in increasingly vocal calls for "theories of organization."[36] But the calls have not yet been answered. A35

Ontological problems have also plagued physics for years, at least since foundational issues of interpretation were thrown into relief by the developments of relativity and quantum mechanics (including the perplexing wave-particle duality, and the distinction between "classical" and "quantum" world-views). They face connectionist psychologists, who, proud of having developed architectures that do not rely on the manipulation of formal symbol structures encoding high-level concepts, and thus of having thereby rejected propositional content, are nevertheless at a loss as to say what their architectures *do* represent. And then of course there are communities that tackle ontological questions directly: not just philosophy, but fields as far-flung as poetry and art, where attempts to get in, around, and under objects have been pursued for centuries.

So there are fellow-travelers. But no one, so far as I know,

36. A theory of organization is essentially applied metaphysics. A36

has developed an alternative ontological/metaphysical proposal in sufficient detail and depth to serve as a practicable foundation for a revitalized scientific practice. Unlike some A37
arguments for realism or irrealism, unlike some briefs *pro* or *con* this or that philosophy of science, and unlike as well the deliberations of science studies and other anthropological and sociological and historical treatises about science, the task I have in mind is not the increasingly common *meta*-metaphysical one—of arguing for or against a way of proceeding, if one were ever to proceed, or arguing that science proceeds in this or that way. Rather, the concrete demand is for a detailed, A38
worked-out account—an account that "goes the distance," in terms of which accounts of particular systems can be formulated, and real-world construction proceed.

For this purpose, with respect to the job of developing an alternative metaphysics, the computational realm has unparalleled advantage. Midway between matter and mind, computation stands in excellent stead as a supply of concrete cases of middling complexity—what in computer science is called an appropriate "validation suite"—against which to test the mettle of specific metaphysical hypotheses. "Middling" in the sense of neither being so simple as to invite caricature, nor so complex as to defy comprehension. It is the development of a laboratory of this middling sort, half-way between the frictionless pucks and inclined planes of classical mechanics and the full-blooded richness of the human condition, that makes computing such an incredibly important stepping-stone in intellectual history.

Crucially, too, computational examples are examples with which we are as much practically as theoretically familiar (we build systems better than we understand them). Indeed—and by no means insignificantly—there are many famous divides with respect to which computing sits squarely in the middle.

8 Summary

Thus the ante is upped one more time. Not only must an adequate account of computation (any account that meets the three criteria with which we started) include a theory of semantics; it must also include a theory of ontology. Not just intentionality is at stake, in other words; so is metaphysics. But still we are not done. For on top of the foregoing strong conclusions lies an eighth one—if anything even stronger:

c8. Computation is not a subject matter

In spite of everything I said about a comprehensive, empirical, conceptually founded "theory of computing," that is, and in spite of everything I myself have thought for twenty years, I no longer believe that there is a distinct ontological category of computing or computation, one that will be the subject matter of a deep and explanatory and intellectually satisfying theory. Close and sustained analysis, that is, suggests that the things that Silicon Valley calls computers, the things that perforce *are* computers, do not form a coherent intellectually delimited class. Computers turn out in the end to be rather like cars: objects of inestimable social and political and economic and personal importance, but not in and of themselves, *qua* themselves, the focus of enduring scientific or intellectual inquiry—not, as philosophers would say, *natural kinds*. A39

Needless to say, this is another extremely strong claim—one over which some readers may be tempted to rise up in arms. At the very least, it is easy to feel massively let down, after all this work. For if I am right, it is not just that we *currently* have no satisfying intellectually productive theory of computing, of the sort I initially set out to find. Nor is it just that, through this whole analysis, I have failed to provide one. It is the even stronger conclusion that such projects will *always* fail; we will *never* have such a theory. So all the previous conclusions must be revised. It is not just that a theory of

computation will not *supply* a theory of semantics, for example, as Newell has suggested; or that it will not *replace* a theory of semantics; or even that it will not *depend or rest on* a theory of semantics, as intimated at the end of §4. It will do none of these things because *there will be no theory of computation at all*.

Given the weight that has been rested on the notion of computation—not just by me, or by computer science, or even by cognitive science, but by the vast majority of the surrounding intellectual landscape—this might seem like a negative conclusion. (Among other things, you might conclude I had spent these thirty years in vain.) But in fact there is no cause for grief; for the negativity of the judgment is only superficial. In fact I believe something almost wholly opposite, which we can label as a (final) conclusion in its own right:

c9. The superficially negative conclusion (that computing is not a subject matter) makes the twentieth-century arrival of computation onto the intellectual scene a *vastly more interesting and important phenomenon than it would otherwise have been*.

On reflection, it emerges that the fact that neither computing nor computation will sustain the development of a theory is by far the most exciting and triumphal conclusion that the computer and cognitive sciences could possibly hope for.

Why so? Because I am not saying that computation-in-the-wild is intrinsically a-theoretical—and thus that there will be no theory of these machines, at all, when day is done. Rather, the claim is that such theory as there is—and I take it that there remains a good chance of such a thing, as much as in any domain of human activity—will not be a theory of *computation* or *computing*. It will not be a theory of computation because *computers per se*, as I have said, do not constitute a distinct, delineated subject matter. Rather, what computers are, I now believe—and what the considerable and impressive

body of practice associated with them amounts to—is neither more nor less than *the full-fledged social construction*[37] *and development of intentional artifacts*. That means that the range of experience and skills and theories and results that have been developed within computer science—astoundingly complex and far-reaching, if still inadequately articulated—is best understood as practical, synthetic, raw material for no less than full theories of causation, semantics, and ontology—that is, for *metaphysics full bore*.

Where does that leave things? Substantively, it leads inexorably to the conclusion that metaphysics, ontology, epistemology, and intentionality are the only integral intellectual subject matters in the vicinity of either computer or cognitive science. Methodologically, it means that our experience with constructing computational (i.e., intentional) systems may open a window onto something to which we would not otherwise have any access: the chance to witness, with our own eyes, how intentional capacities can arise in a "merely" physical mechanism.

It is sobering, in retrospect, to realize that our *preoccupation with the fact that computers are computational* has been the major theoretical block in the way of our understanding how important computers are. They are computational, of course; that much is tautological. But only when we let go of *the conceit that that fact is theoretically important*—only when we let go of the "c-word"—will we finally be able to see, without distraction, and thereby, perhaps, at least partially to understand, how a structured lump of clay can sit up and think.

And so that, for a decade or so, has been my project: to take, from the ashes of computational critique, enough positive morals to serve as the inspiration, basis, and testing ground for an entirely new metaphysics. A story of subjects, a story of objects, a story of reference, a story of history. A40

37. Social construction not as the label for a metaphysical stance, but in the literal sense that we build them.

For sheer ambition, physics does not hold a candle to computer or cognitive—or rather, as we should now call it, in order to recognize that we are dealing with something on the scale of natural science—*epistemic* or *intentional* science. Hawking (1988) and Weinberg (1994) are wrong. It is we, not the physicists, who must develop a theory of everything.

Annotations[v]

A1 ·1/1/1· The originally published version of this paper[1.5] was preceded with the following Editor's Note:[2]

> "What is computation? Not what current theories of computation say it is, argues Smith, as they one way or another 'implicitly rely, without explanation, on such substantial, recalcitrant notions as representation and semantics,' possibly even suggesting computation as a candidate for a theory of those very notions. Smith distinguishes various accounts of computation, originating in different intellectual areas and aiming at different goals. For example, there is the construal of computation as 'formal symbol manipulation,' embracing the idea of a machine manipulating symbolic or (at least potentially) meaningful expressions without regard to their semantic content. Or there is computation seen as the 'execution of an algorithm,' or the mathematical notion of 'effective computability,' Additional notions of computation include 'digital state machine,' 'information processing,' and 'physical symbol system.' All of these construals fail to meet at least one of three criteria, which a comprehensive theory has to satisfy, according to Smith. The first, an 'empirical' criterion, requires theories of computation to do justice to real life computing, that is, to account for and be able to explain programs like Microsoft Word, what it does, how it is used, and so on. The second is a conceptual criterion, which requires a theory of computation to 'discharge all intellectual debts' such as clarifying the relation between computation and various other notions it depends on or is related to. Finally, the third criterion concerns computation's role in computationalism in that it requires a theory of computation also to be an intelligible foundation for the formulation of the computational theory of mind (whether the latter is true or false is not at stake here). Computation, Smith suggests, is intrinsically intentional—this was what made computation an attractive aspect of computationalism in the first place. Yet, it is this intentional or semantic character of computation that is disguised by the widely held, pretheoretic conception of computation as being entirely formal. Once the involved notion of formality is scrutinized, however, it becomes clear that computation cannot be cor-

1. References are in the form *page*/*paragraph*/*line*. See the sidebar on p. 45.

1.5. Scheutz (2008), pp. 23–582.

2. Ibid., p. 23–24.

rectly classified by any reading of 'formal,' and hence the semantic character of computation is in need of explanation. So, rather than providing one, computation will have to wait for the development of a satisfactory theory of intentionality. But Smith does not stop

Annotation Text References

Text references are in the form page/paragraph/line, with ranges (of any type) indicated as α:β. Thus in a reference *x*/*y*/*z*:

1. *x* is the page number
2. *y* is the paragraph number
   a. *y* omitted ⇒ no ¶; if *z* present, z^{th} line on p. *x*; else all p. *x*
   b. *y* = 0 ⇒ partial ¶ at top of p. (cont'd from previous p.)
   c. *y* > 0 ⇒ y^{th} paragraph starting on p. *x*
   d. *y* < 0 ⇒ count from bottom (-1 is last ¶ starting on p.)
   e. Footnotes indicated as paragraph *n1*, *n2*, etc.
   f. Section headings are not considered ¶s, but indented points are (e.g., bulleted or numbered).
   g. Tables, figures, etc., not considered ¶s; should be referred to explicitly.
3. *z* is the line number
   a. *z* omitted ⇒ no line no.; if *y* present, whole ¶; else whole p.
   b. *z* > 0 ⇒ if *y* present, z^{th} line in ¶; else z^{th} line on p.
   c. *z* < 0 ⇒ count from bottom of ¶ or p. (-1 is last line)
   d. Example: 8/-1/-3 means 3^{rd}-from-last line on p. 8, even if last ¶ (-1) continues onto next p. (9/0/...).
4. Ranges: x_1:x_2, y_1:y_2, and z_1:z_2 mean from x_1 to x_2, etc. (pages, ¶s, or lines). Missing values default from the left:
   a. 23/4/5:7 ⇒ lines 5 through 7 in 4^{th} ¶ on p. 23
   b. 23/2/6:4/3 ⇒ 6^{th} line of 2^{nd} ¶ through 3^{rd} line of 4^{th} ¶ on p. 23
   c. 23/1:24/7 ⇒ 1^{st} ¶ on p. 23 through 7^{th} ¶ on p. 24
   d. 127/2/9:128/4 would mean 9^{th} line of 2^{nd} ¶ on p. 127 through 4^{th} line of 128^{th} ¶—probably not what was intended. To signify 9^{th} line of 2^{nd} ¶ on page 127 through 4^{th} ¶ on p. 128, use 127/2/9:128/4/.
   e. 127//-9:128//4 ⇒ 9^{th} line from bottom of p. 127 through 4^{th} line of p. 128

here. Instead he calls into question the whole set of ontological assumptions underlying computational analyses, culminating in his claim that computation is not a subject matter. Hence, although a satisfactory analysis of computation will have to include a theory of semantics and a theory of ontology, we will never have a 'theory of computing,' because computation does not constitute a distinct ontological or intellectual category. To some this may seem a negative conclusion, but for Smith it opens up the possibility of seeing computers as embedded in a rich practice, which might enable us to see 'how intentional capacities can arise in a mere physical mechanism.'" <= check

A2 ·1/–1/2:3 See fn. 3, p. ·2.

A3 ·2/1/–1 The 'cost' metaphor is from Latour («ref»); see also 03, ch. 2.

A4 ·2/2 The cognitive criterion would have been more clearly explained as requiring, of any candidate theory of computation, that it provide a *reflexively* tenable foundation for the computational theory of mind. See the discussion at ·4/–1:·6/0, and also annotation «...» (p. «...»).

A5 2/n2 It is unfortunately common, in philosophy of mind, to take the phrase 'computational theory of mind' to *imply* cognitivism, based on the combination of two mistaken ideas: (i) that computation is necessarily formal symbol manipulation (just one construal of computing; see §3, pp. ...ff), and (ii) that formal symbol manipulation in turn implies a syntactically and semantically compositional "conceptualist" architecture.

A6 2/n3/2 Of the items in this list, the notion of *implementation* deserves special mention. While it is no requirement on a comprehensive theory of computing that it provide an explanation of any particular implementation, the basic idea of implementation is so fundamental to computational practice that a comprehensive theory needs to provide an explanation of what it is for α to be an implementation of β—the general constraints that such an α must meet, the relation of implementation to more general issues of ontological or mereological constitution (type- and token-reduction, supervenience), etc. Among the issues that would need to be addressed, at least three stand out: (i) whether or not an implementation relation can be adequately characterized in purely physical/causal terms; (ii) whether the conditions on implementation are purely *behavioural*, or instead implicate issues of internal constitution or structure; and/or (iii),

related but not identical to the others, whether, if α implements β, the relation between α and β is in any constitutive or necessary way *intentional* (whether α must be able to be semantically interpreted in the same way as β, whether implementation bears any conceptual or logical relation to representation, etc.). See «…15/0 …» for a suggestion that implementation is, in fact, an intentional relation, though I make no explicit argument there in support of such a view.

See ch. 5 for a challenging but nevertheless illustrative case of implementation, and §… of ch. 2 for a discussion of the objectifyiing nature of implementation and its relation to reflection.

A7 · /n7 I have no idea why, though it adverts to its uses in everyday language and in critical theory, this footnote does not even mention the understanding of 'interpretation' that permeates logic and philosophy of language. It is the latter which has been in primary focus in my analysis of computation since the beginning: the idea that the "interpretation" of a sign or signifier (name, variable, etc.) is what it semantically signifies, stands for, denotes, represents, etc.

As suggested in the Introduction, and argued in more detail in ch. 2 (see especially §«…»),[3] the computational understanding of interpretation has increasingly parted company with the classical semantical notion, starting as early as Turing's original 1937 paper, while—confusingly—retaining largely overlapping vocabulary. The computational notion is by and large constrained to be operational and mechanistic, whereas no such condition impinges, or *could* impinge, on the general semantical notion of naming. See also fn. 20, p. ·16, and the extensive discussion of the interpretation of programs in ch. 2.

A8 ·6/0/-4:-3 Reflexive integrity is discussed at length in AOS, but in essence it is a simple idea. If a theory is reflexive (applicable, among other things, to itself), then it is an elementary requirement of theoretical integrity that whatever the theory *claims* about theories should be *exhibited by the theory making that claim*. See fn. 9 on p. 5. Similarly, if it were to be argued, for example, in some theory θ, that theories and the understandings they undergird are never purely rational, but must inevitably include an emotional/affective component, then on pain of reflexive integrity the author of θ should be ready to admit that θ itself must have an emotional/affective dimension. And so on.

3. See also AOS, especially Volume III, where the history of this parting of intellectual ways is identified and historically disentangled.

What makes reflexive integrity challenging is not its internal logic, which is relatively straightforward, but the fact that theoretical advances in reflective analyses can lead to discontinuities in understanding, requiring something of an epistemic (if not Kierkegaardian) "leap" to embrace. The situation is analogous (though by no means the same as) making a shift from realism to a strong version of constructivism: *theories* of both realism and constructivism will differ, based on whether they in turn are understood realistically or constructively, implying that the epistemic process of shifting from a realist to a construvist point of view requires an analogous sort of epistemic leap.

A9 ·8/–2 It is not evident that algorithm execution (ALG) and rule-following (RF) should be conflated in the way indicated here—though neither of the two terms is clear enough in either popular or technical usage to make distinguishing them straightforward. What matters is not the difference between the labels, both of which are used ambiguously, but a critical distinction, applicable in both cases, and mentioned briefly in ·12/1, between: (i) a conception with both mechanical/causal and semantic/intentional dimensions, according to which the behaviour arises from some sort of explicit mechancial "following" of a concrete, physically-effective representation or encoding of a rule or set of steps, in such a way that the resulting behaviour *semantically* satisfies—i.e., normatively accords with the meaningful content of—the rules or set of steps therein represented; and (ii) a weaker variant, with neither causal nor representational implications, according to which the resulting exhibited behaviour merely *honors* or *satisfies* the given rule or rules, or *exhibits* behavior that accords with that which is algorithmically specified (with no implication that its doing so results from those rules being represented or expressed).

Because of the mechanical connotations of the term 'execution,' the former (stronger) version is probably the default reading of "algorithm execution." The latter (weaker) interpretation may be more often associated with the term "rule-following," but the difficulty with it is that it is manifestly *too* weak to serve as a substantial construal of computation. All scientifically described phenomena, at least arguably, are rule-following in this weak sense. Famously, for example, at least to a first order of approximation, the planets and

stars behave in ways that honor Newton's rules of gravity and motion, and thus "follow those rules" in that weak sense—but are in no way rendered computational by that fact. That is not to say that the weak characterization is empty; it remains debatable whether it can be discharged as fully ontological, for example, or whether it bears an epistemic or intentional taint in virtue of the idea of "honouring" or "manifesting behaviour in accord with." Nevertheless, to endorse such a weak conception of rule-following as a *constitutive characterization of computing* would evacuate the notion of substance. Were it true, the resulting computational theory of mind would amount to no more than a thesis that the mind is, as it were, "regular"—which may or may not be true, but is a different question from whether we are computers.[4]

A10 :8/-1/2:3 There is some question of whether this should be generalized from "mathematical function" to "mathematically *modeled* function." The issue is addressed in A05; cf. also ch. 2, §..., and ch. 12.

A11 :8/n12 Cf. ch. 3 of 03 (entitled "Irreduction"), especially its final four sentences:

> "Yes, we need something that will satisfy our yearning for foundations. And *no, it must not be grounded in* α*, for any* α.[5] But there is another possibility. Why can we not just be grounded, *simpliciter*?"

The metaphysical proposal made in 03 represents a (not entirely successful[6]) attempt to paint a picture of world that meets this cri-

4. Whether *computers* are regular is not a question to which the answer is obvious a priori—or, for that matter, a posteriori.

5. 03/83/0/-3:-1. As 03 was being written, a printing error caused a draft, distributed to my students for discussion, to read: "And *no, it must not be grounded in __, for any __*." I was struck that the incorrect version seemed to convey the intended meaning better than the 'α' version, and so since then I have tended to rely on it in talks, teaching, etc.

6. The challenge is that the notions of connection and disconnection on which 03 relies, and to some extent the (perhaps entailed) notions of proximity and distance, can be challenged as playing exactly the sort of "foundational" role that are rejected by the irreduction mandate. Several things could be said in response: (i) that the entire 03 proposal is intended at best to be *a* story of all that there is, not *the* story, implying that other stories might have equal or comparable merit, collectively triangulating on what is ultimately (but ineffably) the case, as suggested in the final figure in the book (03/375/fig 12-1); (ii) that the epistemological aspect of the vision, by no means independent

terion: of being grounded without being grounded in any α. Or, as one might put it, it strives to depict a world with *foundation* but not *foundations*.

A12·10/5-2/-2:-1 Note that Agre's definition eviscerates the computational theory of mind of any substance—or rather, strictly speaking, renders it false, since *our* minds were not artificially constructed. (The point is that Agre places no ontological conditions on what it is to be a computer. The project of artificial intelligence will thus succeed, according to Agre, whenever we learn how to construct minds, *whatever they are*.)

I, too, will argue that computation is not ontologically distinct; see c8 in §8, p. ...., below, though our conclusions differ in both substance and emphasis. Agre seems to favor retaining the word 'computation,' but using it for physically constructed artefacts (whether he would countenance log cabins and oil refineries and the like as computers because they are physical artefacts arising out of human material implementation is not clear, though it would strike me as perverse, to say nothing of being discrepant with common usage). My own recommendation, as suggested in §8, is that—at least for theoretical purposes—we dispense with the term 'computation' altogether.

A12.5 ·11/1 See annotation «A9» at ·....

A13 ·12/1/-2:-1 The term 'information highway' is long since passé, and by now (2014) at best sounds quaint; the same fate may soon befall 'information age.' Other informational framings, however—the idea that the internet is an information resource, that information forms the substance of the digitally-mediated economy, etc.—for now at least remain strong. See AOS Volume IV. ✔

A14 ·12/-1/5:8 The historical analysis of information as nineteenth-century notion, and its characterization as a publicly-accessible, authoritative, somewhat corpuscular commodity, is due to Nunberg.[7]

. . .

of its ontological dimension, is framed so as to support unliimited challenge, revision, etc., so that even if connection and disconnection do tread on the forbidden 'α' (or '__') territory, that does not render the notions either immutable or immune to challenge; and (iii) that—which is especially noteworthy in the present content—the notion of connection is aimed squarely at what the notion of effectiveness means in computer science, and so is far from arbitrary. Nevertheless, what 03 failed to do, which even by its own lights it should have, is to "derive" connection and disconnection from a careful process of immanent induction.

7. «...ref...»

A15 ·12/–1/–4: ·13:0 Alas the grammatical problems with this sentence are in the original. At a minimum, clauses (ii) and (iii) should begin with 'as in.'

A16 ·13/–1/3:4 This claim illustrates the point mentioned in §… of the Introduction, and explored in some detail in ch. 2: I initially felt that the most dominant problems among all theoretical difficulties facing an account of computing were semantical—in spite of the very strong ontological conclusions reached below (e.g., in claims c…-c…, q.v.).

A17 ·14/0/–4 Should 'the cognitivist hypothesis,' in this sentence, be replaced with 'the computational theory of mind'? As noted in fn. 2 (p. ·2) cognitivism, at least as that term is used in cognitive science and AI,[8] is a narrower thesis than the idea that minds are computers, suggesting a positive answer. Or at least that is so if one accepts the formal symbol manipulation (FSM) construal of computation. Someone who embraces a non-intentional characterization of computing might endorse a computational theory of mind without believing that any intentional properties bear on the question of whether or not minds are computers (since computing, according to them, need not have any such properties).[9] While reading this sentence in terms of a cognitivist conception is undoubtedly too narrow, therefore, broadening it to an unrestricted computational theory of mind would be too wide.[10] *(cont'd)*

Similar considerations apply to the use of the term 'cognitivist' in the first sentence of the subsequent paragraph (15/0/1).

A18 ·15/–1/–6:–3 Cf. the discussion of overlapping technical vocabularies in §… of the Introduction, p. ….

8. I.e., when the conceptualist or propositional ingredients taken to be constitutive to mind as assumed to be implemented in something like a compositional formal symbol system, as in formal logic.

9. It may not seem quite entailed by a non-intentional construal of computing that no intentional properties can bear on the question of whether people are computational, since the computational theory of mind inevitably places additional constraints, having to do with what *kind* of computers people are. As explained in «…», however, those additional constraints should be computational in nature, which if computation is held to be non-intentional would block intentional properties entering at this stage.

10. Someone who embraces a non-intentional construal of computing, such as effective computability (EC) or digital state machines (DSM), owes us an explanation of why we should have any in-advance sympathy for the thought that we are computers. It is not that such an intuition might not be forthcoming; it just cannot lean on the fact that "we, too, deal with representations, symbols, meaning, information, and the like."

A18.5 ·16/n20 Ch. 2 contains a much more extensive analysis of the issue discussed in this note.

A19 §5a (·18/-1:·26/3) For more on formal symbol manipulation (FSM), the topic of this subsection, see AOS Volume II, devoted in its entirety to an analysis of this construal.

A20 ·19/1/-7:-4 Some might argue that no form of relationality could be intrinsic to intentionality, on the grounds that relational properties are necessarily *extrinsic*, and thus not intrinsic to anything. But the argument is invalid, trading on ambiguities in the meaning of 'intrinsic.' Relationality could still be intrinsic to intentionality in the sense of being *essential* or *necessary* to it. All that follows, if nothing can be intentional without exhibiting relational properties, is that *being intentional* must not be an intrinsic property in the classic sense.

The point grows complex only when one attempts to identify what it is that being intentional is *not intrinsic to*. Assume, uncontroversially, that it is essential to a thought's *being a thought* that it be intentional, and also, more controversially (as suggested here), that it is essential to something's being intentional that it exhibit relational properties. Are we to conclude that being a thought is not an intrinsic property of…the thought? Not quite—or anyway that would be an infelicitous way to put the point. Strictly speaking, it would be better to say something along the following lines: that being a thought is not an intrinsic property of *that patch of reality that, if one were to advert to relational facts, could correctly be labeled "a thought."*

There is no fundamental problem with this line of reasoning—at least to the extent that the words *intrinsic*, *extrinsic*, *relationality*, etc., mean anything. If one believes in relationality, that is, then one should recognize intentionality as being necessarily relational.

I say "if one believes in relationality" because I myself do not—in part because, as argued in 03, I do not even believe that "being an object" is an intrinsic property of…objects. Except, as recommended above, if we are to speak strictly, then we should say: *of those patches of reality that, if one were to permit adversion to relational facts, could correctly be labeled 'objects.'* But the verbal awkwardness is telling—and suggestive of why the whole intrinsic/extrinsic vocabulary should be set aside. If not even an object's *being the object that it is* is an intrinsic fact about it, it is murky to know what it would mean to claim that some *other* property P is intrinsic to *it*.

* * *

Arcana aside, a serious point underlies these deliberations. It is a major theme of both 03 and A05: (i) that some form of *disconnection*, involving a lack of (or weakness in) effective or mechanical coupling, is essential to intentionality; (ii) that the official theory of computation is a theory of *effective* connection or coupling;[11] and (iii) that "effective" is the only legitimate substance to (i.e., only grain of truth to be found in) the "syntactic" or positive reading of 'formal,' as that term is used in the formal symbol manipulation (FSM) construal of computation. It follows that there cannot be a mechanical or causal theory (or formal in the "syntactic" sense) theory of intentionality or computing. Or to put the point more broadly: neither intentionality nor computing, in my view, can be understood from within the mechanical restriction (cf. ch. 1, §…).

A21 ·23/1/8:9. See ch. 2 (especially §s …) for an extended discussion of whether programs' constituent symbols are viewed, or should be viewed: (i) as designating entities in a programs' *task domain*—which will often, though not always, be external to the computer (salaries, orbits, people, etc.); or (ii) as designating internal entities—such as data base entries, memory locations, etc. (entities that may, in turn, represent or model those external real-world[11.5] entities). The 2Lisp and 3Lisp dialects, subject of chs. 3–5, take the former view; most of computer science adopts the latter (hence the 'arguably').

A22 ·24/n28 /3:6 These disciplinarily specific meanings of 'transducer' are yet another instance of the cross-disciplinary terminological confusion discussed in §4 of the Introduction.

A23 ·25/1/9:11 Needless to say, which operations should count as "reference crossing" (i.e., which operations mediate between a sign or symbol and what it designates) depends on what one takes the designation of that sign or symbol to be. The ambiguity about the semantics of program symbols suggested above (cf. annotation A21, as well as the extensive discussion in §… of ch. 2) shows not only that there *is*

11. See §5b, and especially c5 on p. ·29.

11.5. Computational internal entities are part of the real world, needless to say; I use the term in the (regrettably) common sense only to make the point.

no agreement, but also that *there has not needed to be any agreement on the issue*, underscoring the claim in the text that no computationalist has ever needed to distringuish computational operations that do from those that do not cross such semantic boundaries. (See also ch. 12.)

A24 ·25/1/13 Counting is an instance of a reference-crossing operation because it takes as input an exemplified cardinality—an actual *number* of entities (seven people, say)—and produces as output a *symbol* designating that cardinality (the numeral '7', or the word 'seven'). Cf. AOS, Volume II.

A25 §5b (·26/-2:·29/3) For more on Turing machines, effective computability (EC), etc., see AOS Volume III, which consists of a detailed analysis of this second construal.

A25.5 ·27/2/5:7 It is notable, and potentially distracting, that mathematical models of computability differ from mathematical models of physical phenomena in not being framed in terms of physical units. This fact is superficially explained, I believe, by the fact that computational regularities hold of physical arrangements more abstractly individuated than is typical in physical theory. Much more substantial, however, is a profound and unresolved underlying issue: all theoretical results in computer science rest on a basis of a largely unrecognized[11.7] and wholly unexplicated *individuation of discrete physical states*. It is this ontological parsing that allows the mathematics to be erected without reference to specific physical quantity.

Evidence of the complexities and confusions that can result from using non-standard state individuation are rampant in the literature, and include Searle's (second) argument against the plausibility of strong artificial intelligence, Putnam's claim that a rock has the computational power of a Turing machine, etc.[11.8] To my knowledge, however, no one has yet proposed a satisfactory solution to the problem (though perhaps Gandy (19...) can be taken as an early attempt). My own view is that state individuation is ultimately a result of intentional processes of registration—a claim that substantially upends prospects of naturalization.[11.9]

11.7. What is unrecognized is not the state individuation itself, but the fact that it remains so theoretically unreconstructed.

11.8. Searle (19...), Putnam (19...).

11.9. See AOS Volume III for an extensive discussion of physical state individuation, and 03 re the intentional nature of registration.

A26 ·28/2/–3 While the popularity of Girard's linear logic was distinctively increasing when this paper was written (2001), it is unclear in 2014 that this is any longer so.

A27 ·31/2/8 See Dienes & Perner (1999) and Davies (2001, §6) for discussions of different uses of the terms *explicit* and *implicit*, relating them to correlative distinctions between and among: procedural vs. declarative knowledge; conscious vs. unconscious knowledge; tacit forms of knowing; actual expression or "direct statement" vs. functional consequence, logical implicature, and/or conceptual presupposition; the availability or unavailability of a representation to be the object of meta-level representation; etc. Within computer science, perhaps the most common interpretation of the distinction has to do with the computational cost of drawing an (explicit) conclusion: as Levesque famously put it (1984): "a sentence is *explicitly* believed when it is actively held to be true by an agent and *implicitly* believed when it follows from what is believed."[12] Using computational cost to distinguish waht is explicit from what is implicit is also endorsed by Kirsh (1990), his classic paper on different uses of the terms in cognitive science.

By and large, all these discussions take implicitness and explicitness to be properties of *representations*—though what it is to represent is itself highly problematic. Thus Kirsch (1990, p. 347) suggests that systems may sometimes know things without representing them at all,[13] much as other writers (e.g., Rosenschein 1985, Halpern 1995) characterize computer systems as carrying information and knowing things non-representaitonally. Nevertheless, even in the case Kirsch cites, the overall focus remains representational.

A28 ·31/–1/–2:–1 Although the five readings of 'formal' listed on p. ·32 are the ones that have figured most prominently in the development of theoretical edifices, a number of more idiosyncratic suggestions arose in the course of the interviews mentioned in note 33. One of the most intriguing was a suggestion that 'formal' means *authorized*—as for example in "a formal invitation from the White House." The reso-

12. Emphases in the original. "Actively held to be true" means explicitly represented and functionally located in the appropriate way.

13. His example is of a vision system whose ability to derive 3D shapes from stereo 2D images depends on a fact about the world that it "assumes" without representing at all: that objects in the world change shape smoothly and continuously.

nance between its evocative connotations and the privileged status of formality in theoretical computer science is a little uncanny.

A29 :32/3/3 The term 'construal' on this line (and also three lines further on, in list item 4) should be replaced by 'reading.' I was not referring to the seven "construals" of computation under theoretical scrutiny.

A30 :33/1/7 The phrase "in the full analysis" is a reference to AOS.

A31 :34/1/2 By 'personal' here I mean approximately autobiographical, as is common usage these days—i.e., as having to do with the author. Today I would have instead used the term 'individual,' or rewritten the sentence entirely in order to avoid the current (and unfortunate, in my view) individualist and even egocentric connotations of the term. We would do better, in my view, to retain 'personal' to mean the opposite of 'impersonal'—i.e., as having to with *persons*.

A32 35/-1/7 For a discussion of this use of the term 'register' see «ref Rehab» and 03.

A33 :36/1/10 I have come to prefer Cussin's term 'preemptive registration,'[14] used here, over 'inscription error,' the phrase I used in 03 for essentially the same phenomenon. See in particular «03; chapter...»

A34 37/-1/2 At the time this was written I knew little of cultural or critical theory, or of science and technology studies (STS), or of feminist epistemology, in all of which the impositional nature of concepts and categories has been the target of stinging analytic critique. Reference to those literatures would have mitigated the need to advert solely to conceptual clash as demonsrating the dangers of imposing a conceptual frame on a subject matter. But the overall point would have remained.

Cf., however, the discussion of the "metaphilosophical" orientation of many of the discursive traditions in §... of the Introduction.

A35 38/1/-5:-4 The primary concern I had in mind here had to do with what in the Introduction I characterize as *blanket mechanism*—an uncritical assumption that fitness landscapes, self-organizing systems, emergent phenomena, etc., must be constitutively characterised in causal or mechanistic terms—rather than, to take the evident contrast, referentially, semantically, intentionally, and/or normatively.

A36 :38/n36 Or as I have put it elsewhere «where?», "a theory of organization is metaphysics with a business plan."

14. «Ref...or at least explain».

A37 ·39/0/3 Cf. the discussion of the limits of discursive critique in §… of the Introduction.

[NB: the term 'foundation' on this line was misprinted as 'foundational' in the original published version.]

A38 ·39/0/–4:–3 The aim of the fan calculus mentioned briefly in §… of the Introduction, is to serve as a substrate (a "kernel calculus," in the terms of §«…» of ch. 2) in terms of which to provide such a detailed account. At least at the time of this writing, however, and in spite of a number of draft sketches and public talks, such a calculus remains more dream than reality.

A39 ·40/–2/–1 Needless to say, instead of 'natural kinds' this should read "instances of a natural kind."

A40 ·42/–1 The metaphysical sketch presented in 03 was exactly based on such a project: to mine computational systems and experience in order to envision (albeit sketchily) a metaphysical account of the world adequate to their inclusion.