Center for the Study of Language and Information
Report No. CSLI-87-71

January 1987

The Correspondence Continuum

Brian Cantwell Smith

CSLI was founded early in 1983 by researchers from Stanford University, SRI International, and Xerox PARC to further research on the development of integrated theories of language, information, and computation. CSLI headquarters and the publication offices are located at the Stanford site.

CSLI/SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

CSLI/Stanford
Ventura Hall
Stanford, CA 94305

CSLI/Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304

The Correspondence Continuum

Brian Cantwell Smith

Intelligent Systems Laboratory, and
Center for the Study of Language and Information
Xerox Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, California 94304

Abstract

It is argued that current semantical techniques for analysing knowledge representation systems (clear use/mention distinctions, strict metalanguage hierarchies, distinct "syntactic" and "semantic" accounts, even model-theory itself) are too rigid to account for the complexities of representational practice, and ill-suited to explain intricate relations among representation, specification, implementation, communication, and computation. By way of alternative, the paper advocates the prior development of a general theory of correspondence, able to support an indefinite continuum of circumstantially dependent representation relations, ranging from fine-grained syntactic distinctions at the level of language and implementation, through functional data types, abstract models, and indirect classification, all the way to the represented situation in the real world. The overall structure and some general properties of such a correspondence theory are described, and its consequences for semantic analysis surveyed.

A slightly earlier version of this paper appeared with the Proceedings of the Sixth Canadian Conference on Artificial Intelligence, Montreal, Quebec, Canada, May 21–23, 1986. The research was conducted at the Center for the Study of Language and Information, supported in part by Xerox Corporation and in part by an award from the System Development Foundation.

The Correspondence Continuum

Brian Cantwell Smith

1. Introduction

Certain genitive phrases of the form 'α of β' are ambiguous. On the *subjective* reading of 'love of children', for example, it is the children who do the loving, as in (1). On the *objective* reading, in contrast, children are recipients of the affection, as in (2).

1. Though bitter from years of adult ridicule, the old man was grateful for the love of children.
2. Though increasingly impatient with his peers, the old man never lost his love of children.

The problem arises when the head noun phrase α ('love') signifies an asymmetric two-place relation, since it's then unclear which argument place is filled by the β term following 'of'. As shown in these examples, the distinction is generally clear-cut, with the intended reading selected by context (this is why it a question of ambiguity, not vagueness).

The phrase "the representation of knowledge" is of this ambiguous type. Oddly enough, though, it's not clear which reading is intended. Is knowledge being represented (objective), or is knowledge doing the representing (subjective)? Both interpretations seem reasonable. For example, suppose we build a medical AI system called DOC using FKRL, our favourite knowledge representation language. On the objective reading, the ingredient structures would be viewed as representing DOC's knowledge, presumably implying that a semantics for FKRL should map FKRL structures onto knowledge (or perhaps onto a set-theoretic model of it, such as a possible-world structure). On the subjective reading, in contrast, DOC's knowledge, embodied in FKRL structures, would itself be taken as representational. In this case semantic analysis would map the representational structures onto the states of affairs in the world that DOC knows about — states of affairs involving drugs, diseases, and diagnoses.

To add to the confusion, it's not even clear exactly what the difference between the two readings would come to, in the knowledge representation case. It seems that a possible world structure modelling belief might be the same as a structure modelling the states of affairs that the belief is about. And yet beliefs and worldly states of affairs aren't the same: the former, for example, are psychological, the latter not (at least in general). Thus, whereas an erudite doctor might be said to possess great knowledge, it would be senseless to say that she possesses great states of affairs.

Some of the confusion has a simple source: *both* 'representation' and 'knowledge' designate asymmetric, relational notions. Furthermore, the two relations are of the same general type; they both characterise phenomena that are *about* something, that refer to the world, that have meaning or content. For example, to say that a series of marks on a page is a representation of Winston Churchill is to say that there is some relation between those marks and the late British Prime Minister. Similarly, to say that your lawyer's knowledge is faulty is to comment on the relation between what what's going on inside the lawyer's head and what's going on outside. Because they are both based on an underlying (asymmetric) relation of content, representation and knowledge are considered to be *semantic* or *intentional* notions (other intentional notions include language, belief, model, theory, specification). But to say that isn't to say very much, at least not yet. It certainly doesn't explain how representation and knowledge differ. Nor does it clarify our starting question of how, in the knowledge representation case, they are supposed to relate.

This paper will try to sort this all out. Specifically, taking semantics as the general enterprise of describing intentional phenomena, I will address the question of what it is to give a semantic analysis of a knowledge representation system. I.e., whereas most semantical analyses focus on *particular* types of semantic entities or structures — possible world structures, partial situations, etc. — my concern will be with the overall framework in terms of which such analyses are conducted. There are several reasons this is an urgent task. The first we have already discussed: as implied by the confusion in the name, there are several interacting intentional notions involved, which should be sorted out. Second, it is increasingly thought necessary to give semantical accounts of proposed representation systems, in order to convey rigour and coherence onto what would otherwise be viewed as ad-hoc symbol mongering. In 1974 Hayes, long a champion of this view, called AI's reluctance to provide semantical accounts for representation schemes "a regrettable source of confusion and misunderstanding" [Hayes, 1974 p. 64], and went on in 1977 to write as follows:

> One of the first task which faces a theory of representation is to give some account of what a representation or representational language *means*. Without such an account, comparisons between representations or languages can only be very superficial. Logical model theory provides such an analysis. [Hayes 1977, pp. 559]

In writing these words Hayes was defending logic against what he took to be the a-semantical orientation of the proceduralist tradition. In this he seems to have succeeded: similar sentiments have since gained widespread allegiance. We should certainly understand anything so popular.

On the other hand, this very success leads to the third reason for the present investigation. I believe that current theoretical tools, particularly the traditional model-theory that Hayes cites and most everyone uses, are inadequate to the knowledge representation task, and need substantial revamping. Perhaps ironically, many of the problems I will canvass are foreshadowed in Hayes' original papers — the relation between so-called propositional and analogue representation, to take just one example, which has yet to be adequately reconstructed. Logical model theory, which doesn't address analogue questions, has if anything gained momentum as the knowledge representer's semantical technique of choice.

Fourth, and finally, many of the lessons learned in the knowledge representation case will hold for all computational systems, and will even impinge on general semantical analysis; so there's a certain universality to the inquiry.

2. A Model of Knowledge Representation

I will adopt a *two-factor* model of knowledge representation, as pictured in Figure 1. An agent, computational or human, is taken to comprise a set of internal structures, states, or aspects, that have some sort of content, and at the same time play a role in engendering the agent's overall behaviour. In order to focus on their internal nature, I will call these structures *impressions*, to distinguish them from *expressions*, elements of an external language. Think of impressions as data structures, as elements of a knowledge representation language, or as partial mental states — not much will depend here on details. The essential point about impressions is that they have two partially independent, though coördinated, properties.

First is what I will call *functional role* (or 'role', for short). Impressions must arise, somehow, in virtue of the history and coupling of the agent to its environment, and must give rise to the system's future activity or behaviour. Furthermore, as well as having these backwards- and forwards-looking aspects, impressions must be causally efficacious in the present — must bump up against each other, or be manipulated by some sort of internal agency, so as to constitute the whole of which they are the parts. So a given impression, such as one expressing the fact that a robot doesn't have much time left until it needs a recharge, might arise from the integration of information gleaned from internal sensor readings, engender inference involving time and expected electrical use, and lead the robot to scramble around the hall in search of an electrical outlet.

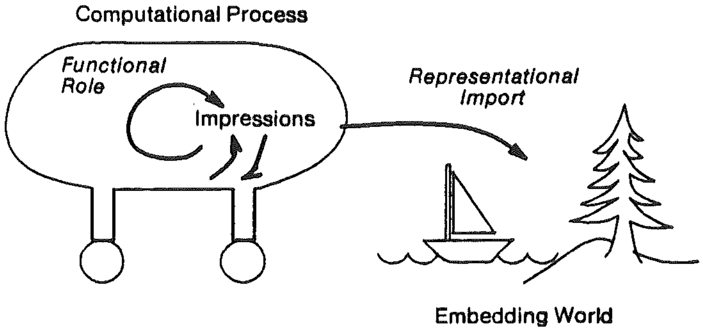

Figure 1: A two-factor model of knowledge representation

Functional role, however, isn't enough. In order to count as representations, as opposed merely to being causal ingredients like the cam shaft in a car, impressions must also stand in a content relation to the states of affairs in the world in which the agent is embedded. I will call this second factor *representational import* (or just 'import', where the meaning is clear).

Representational import isn't an alternative to functional role, or a particular kind: it is something additional. Thus whereas the level of sap in a maple tree arises from a complex history involving the weather, structure of the trunk, etc., and gives rise to complex future behaviour, such as amount of sugar produced, density of new foliage, etc., that's about all there is to say about it. In spite of being correlated with facts in its environment, sap doesn't have any representational import partly because the correlation is too strong (sap can't be wrong), and partly because no concepts are employed (sap doesn't represent the world *as being one way as opposed to another*; it is merely locked into it as a totality). In contrast, for an impression to represent spring's being on the way, there would have to be an additional uniformity relating its structure to the structure of that fact — a uniformity that would be missed in an isolated account of functional role.[1]

For example, suppose I have the impression that water conducts electricity. All kinds of backwards-looking functional roles could have led to this: my own hapless experience trying to heat the bath water with an electric iron; stories I've been told; books I've read; deductions from knowledge of the ionization potential of molecules held together by hydrogen bonds. Similarly, at least within wide limits, there is no predicting what forward-looking role this impression might give rise to: things I might say, or situations I may strenuously avoid, such as climbing onto high-tension wires during rainstorms. The point is that, in spite of this richness of role, including inferential role, there remains a striking and relatively simple uniformity connecting the impression and the fact it represents — the most penetrating regularity in terms of which to explain my behaviour. In brief, it's the connection between the impression itself and the fact that water conducts electricity. This is the regularity of content or representational import.

These two factors must be coördinated in a special way: the states of affairs that the impression represents (its import) and the behaviour that it gives rise to (its role) must be such that the agent can be truthfully said to *know* the fact, which involves being able to act in accord with it, etc. The trick, in spelling this out, is to tie the two roles together into an integral whole without thereby undermining the integrity of the distinction between them — a project that requires combining traditional semantical techniques with the AI and philosophical literature on knowledge as action, pragmatic reasoning, and even causal theories of reference. I won't attempt that integration here, but will merely call the coördinated combination of factors the *full significance* of an impression.

In [Smith, 1982a, 1985] I labelled this two factor orientation to representational significance the *Knowledge Representation Hypothesis*. In the philosophy of mind an analogous view has been labelled a *dual-component* semantics for psychology [Field, 1977, 1978; Loar, 1982; Block, 1985]. Technical variations have appeared under various descriptions; what is perhaps most striking is its familiarity in even the familiar realm of formal logic. In a traditional proof-theoretic framework — say, if the agent was an implementation of a natural deduction theorem-prover for first-order logic — one might view representational import as the *semantics* of an expression, and functional role as its *proof-theoretic* consequence. This last characterisation, however, misleadingly suggests that the full significance of a representation system must satisfy the following two constraints:

1. That the two factors be essentially *independent* (in which case I will call the representational system *declarative*); and
2. That functional role arise solely from *syntactic* properties of the representational structures.

Adherence to a general two-factor analysis, however, in no way commits one to this particularly strong way of dividing things up.[2] 3-Lisp, for example,[Smith, 1982a, 1984] a simple programming language designed within a two-factor framework, explicitly violated both assumptions: import and role were both essentially semantic;[3] it was also shown that they were theoretically explicable only in intimate conjunction.[4] Other analyses, such as that suggested in Barwise and Perry's [1983], propose alternative ways of tying content and behaviour together. In fact it is partly because there are so many ways of getting at roughly the same intuition that I have presented it here somewhat abstractly.

The two-factor nature of knowledge representation is the most important aspect for semantical analysis to clarify. In order to make sense of current semantical techniques, however, we need another distinction, which cross-cuts it. Especially in the philosophical literature, semanticists sometimes distinguish the *meaning* of a structure from its *content* or *interpretation* (not, at least not in any straightforward way, to be confused with the computer science notion of interpretation; see section 5, below, and [Smith, 1984]). The former, very roughly, is what all instances or uses of a given structure type have in common; the latter, what a particular use or instance of that type refers to, or gets at, in all its specificity. Typically, facts about the context or setting provide the additional information that gets from meaning to interpretation. So for example the first person pronoun 'I', under this analysis, has the meaning of referring to whoever uses it: when Mick Jagger says 'I', he refers to himself; if you do, you refer to yourself. This is why two people can scream at each other "I'm right; you're wrong!" — they both use the same sentence, and the meaning is the constant; it is the respective contents that are contradictory. So we might model the meaning of 'I' as the following function of speakers, times, and locations: $[\![\mathrm{I}]\!] = \lambda \mathrm{S,T,L}\ .\ \mathrm{S}$. In a given situation of use (speaker S_0 at time T_0 in location L_0) the intepretation would be S_0.

It is tempting to identify meaning as the semantics of types, interpretation as the semantics of tokens, but the second of these is misleading. John Perry, for example, has imagined a case of two deaf mutes, so poor they must share a single tattered card saying *I'm a poor deaf mute; won't you give me some money.* Standing together at the street corner, they alternately hand the card to passers by. Each time the card is used, the words 'I' and 'me' change referent: one token, constant meaning, changing interpretation. Similarly, consider an analogous computational example: a machine with a single distinguished internal structure used to mean 'now'. The meaning is constant, and the particular structure may persist, but the interpretation changes with each passing nanosecond. Uses, or utterances, are what have interpretations; not concrete instances or tokens.

The meaning/interpretation vocabulary isn't common in the AI or computer science literature, but the circumstantial dependence with which it deals is ubiquitous. Even the simple inclusion of explicit environment and memory arguments in denotational analyses of programming languages (see for example [Gordon 1979]) manifests a sensitivity to the importance to interpretation of contextual factors. In [Smith, 1986] I lay out a whole variety of ways in which the content of computational structures, including impressions, can depend on facts of circumstance or context: internal facts (what program is running, how other internal structures are arranged, etc.), external facts (where the computer is located, whom it's conversing with, etc.), and even some facts that seem to cross the boundary (what time it is). The importance of these kinds of circumstantial dependence will be assumed in what follows.

Furthermore, both aspects of significance — functional role and representational import — can be circumstantially dependent. What ¬FLIES(X) means, when attached to the BIRD node in a default reasoning system, and what inferences it leads a system to draw, can both depend on the presence or absence of other intermediating impressions. I will use *functional meaning* and *representational meaning* to get at the respective factors of an impression's significance abstracted away from details and circumstances of particular instantiation or use. Similarly, *functional content* and *representational content* will refer, respectively, to the actions a use of an impression actually engenders, and to the situation it actually represents.

Given these distinctions, our overall question is this: what would a semantical analysis be of the full significance of impressions? In the broadest terms, it will clearly have to distinguish import and role, meaning and content, and show how they all come together into a coördinated whole. But we need details. I'll proceed in steps, concentrating first on representational import. Later we'll return to the question of how to tie it together with functional role.

3. The Present State

Virtually all the theoretical techniques in our current semantical arsenal were developed to deal with representational import. In particular, present practice proceeds roughly as suggested in Figure 2. First, a source domain is identified as the set of elements for which a semantical analysis is to be given. Traditionally, this is called the *syntactic* domain; in the knowledge representation case it is the set of impressions comprising the agent (I'll talk more about the difference in a moment). Second, a semantic domain is similarly identified, roughly taken to be what the elements of the representational domain, expressions or impressions, are about (more about 'aboutness', too, in a bit). Third, the semantic relation between domains, usually called the *interpretation* relation, is then described *extensionally*, in the sense that particular elements of the syntactic domain are mapped, piece-wise, onto the corresponding particular elements of the semantic domain. It may be, in the theorist's actual presentation of the semantic relation, that considerable information about the structure of this relation will be manifested, but strictly speaking this additional structure isn't part of what is provided (or perhaps, to borrow from the *Tractatus*, we could say that it is *shown*, but isn't *said*). Just as for functions and relations more generally, piece-wise correspondence is assumed to be sufficient, at least for theoretic purposes.

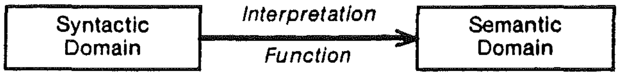

Figure 2: The standard semantical model

So far, however, I haven't said enough to distinguish the extensional analysis of a semantic relation from the extensional analysis of any old relation at all. But in practice more assumptions are adopted. I will label as *model-theoretic* those semantical analyses that accept (which I don't!) the following additional claims:

1. The elements of the representational domain are assumed to be *linguistic*. Debates rage over what language is, but at least the following seems agreed: complex linguistic elements are taken to be linear sequences of some sort (strings, utterances, whatever), with an inductively specified recursive structure founded on an initial

base set of atomic elements called a *vocabulary*, and assembled according to rules of composition specified in a *grammar*. Furthermore, the interpretation relation is usually defined *compositionally*, so that meanings (not contents!) are assigned both to the vocabulary items and to the recursive structures engendered by the grammatical rules, in such a way that the meaning of a complex whole arises in a systematic way from the meanings of its parts. A particularly strong version of compositionality requires that the meaning of a whole be definable, often by function composition, in terms of the meanings of the parts, but other possibilities, such as that the whole's meaning be characterised, or even just constrained, by systems of regularities among the parts, are growing in popularity. We needn't take a position here on details.

2. In a case where the elements of syntactic domain S correspond to elements of semantic domain D_1, and the elements of D_1 are themselves linguistic, bearing their own interpretation relation to another semantic domain D_2, then the elements of the original domain S are called *metalinguistic*. Furthermore, the semantic relation is taken to be *non-transitive*, thereby embodying the idea of a strict use/mention distinction, and engendering the familiar hierarchy of metalanguages. This distinction is motivated by such obvious facts as that the six-character quoted expression ‘"Nile"’ designates a short word, which in turn designates a long river, but from those two facts it does not follow — nor is it true — that the six-character ‘"Nile"’ designates the river.
3. The interpretation relation, as suggested in Figure 2, is typically taken to be a function, implying that the import or content of an expression isn't ambiguous. But ambiguity is a relative term: a linguistic element may look ambiguous if the circumstantial dependence of content hasn't been fully articulated, and may therefore be resolved by the meaning/interpretation distinction. We have already seen how the functional assumption is generalised to handle such complexities: whatever information disambiguates a given use of an otherwise ambiguous expression is included as a parameter of meaning; content is then obtained from the meaning by fixing that parameter. For example, the interpretation of the indexical expression 'I', discussed above, was parameterised on speakers. Similarly, if 'grue' means blue if used before some time T_0,

and 'green' afterwards,[Goodman, 1983] then interpretation would in general be paramaterised on time of use, and 'grue' assigned roughly the following meaning:

$$[\![\text{grue}]\!] = \lambda S,T,L\,.\ \text{if } T<T_0 \text{ then BLUE else GREEN.}$$

Thus the true situation is more accurately pictured by Figure 3, with dependence on circumstantial or contextual factors folded into the interpretation. As mentioned earlier, the discussion [Smith, 1986] was intended to show how facts about both internal and external context can affect interpretation in this way.[5]

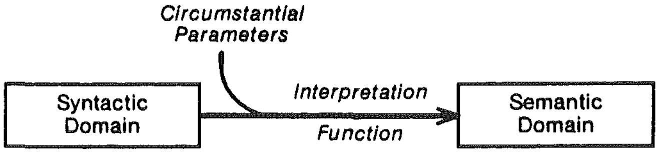

Figure 3: Parameterised interpretation

4. It is not necessary — not even usual — to require that the semantic domain be the real domain that the expressions are about. Rather, the semantic domain is required to be a set-theoretic structure, viewed as a *model* of the real semantic domain.

This last assumption serves a variety of useful functions: it means that semantical analysis remains "purely" mathematical, rather than having to spell out complete metaphysical assumptions about the true nature of the world. So for example a belief or proposition might be modelled as a function from possible worlds to truth-values, without the theorist needing to believe that that is what beliefs *really are* (but of course they *aren't* functions: it is entirely reasonable to ask "what are your friend's beliefs?", absurd to ask "what are your friend's functions from possible worlds to truth-values?"). Similarly, in the semantical analysis of a language used to describe Turing machines, the semantical domain is usually taken to be sets of quadruples, not actual devices complete with tapes, read/write heads, finite state controllers, and so forth. The quadruples are viewed as a *model* of the Turing machine, and — this is the crucial point — *modelling is free*, in the sense that the theorist has license to engage in unconstrained modelling without having to account for it explicitly in his or her theory. To put it another way, modelling is invisible through the standard semanticist's glasses.[6]

Sometimes, of course, when the linguistic or representational elements are genuinely about mathematical objects — theories of arithmetic, for example, or representations of the factorial function — the true interpretation may be one of the model structures (called the 'intended interpretation'). In general, however, and almost universally in the AI knowledge representation case, we are interested in representations of more general states of affairs in the world, such as levels of digitalis in heart patients. So the picture of semantics should be updated as in Figure 4.

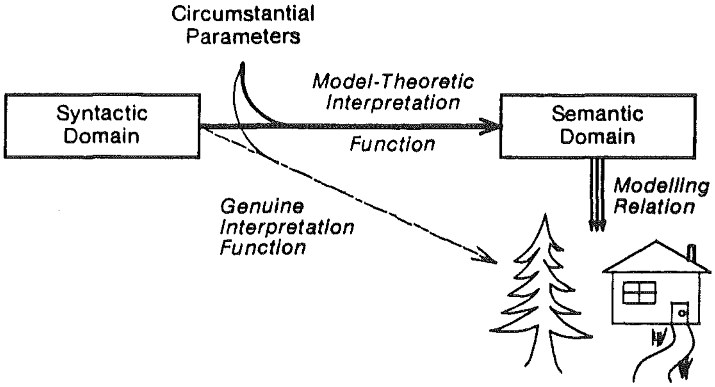
Circumstantial Parameters
Model-Theoretic Interpretation Function
Syntactic Domain
Semantic Domain
Modelling Relation
Genuine Interpretation Function

Figure 4: Parameterised model-theoretic interpretation

Finally, in discussions to follow, we will encounter complex situations that include both modelling and iterated representation of the sort discussed in the second assumption. So it is important to summarise how the standard picture would look in such cases. Since modelling is typically ignored, such a situation would traditionally be *described* as a strict series of non-transitive denotation relations, each analysed piece-wise. Our comments about modelling might suggest that the true situation is more complex, consisting of a series of non-transitive denotation relations, followed by an indefinite amount of promiscuous modelling. But in fact, since there may be promiscuous (i.e. invisible) modelling at each stage of the language hierarchy, as for example when a language is encoded in arithmetic (as is common in recursive function theory, for example), what we really have is this: a strictly non-transitive sequence, each step consisting of a denotation relation followed by an indefinite amount of promiscuous modelling. This situation is pictured in Figure 5.[7]

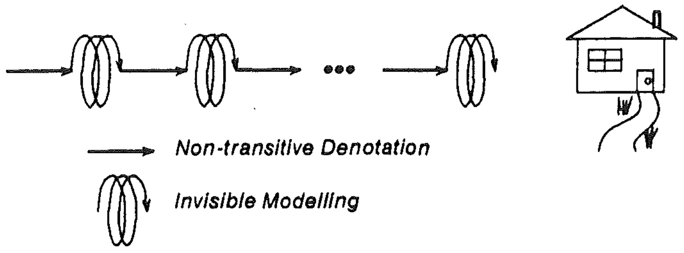

Figure 5: The model-theoretic analysis of iterated representation

4. Impressions

The first step, in analysing the appropriateness of the semantical techniques described in section 3 for the representational problems presented in section 2, is to decide how we are going to treat impressions. Because I specifically introduced the term to cover any internal aspect, state (or partial state), or structure, we want a fairly general answer. Also, it’s a surprisingly complex subject; if we can clear it up first, subsequent semantical analysis will be that much more tractable.

The most important point is this: *we theorists don’t yet have any developed theoretical terminology whose primary function is to describe impressions.* In particular, they aren’t necessarily linguistic objects, since the notion of language arises from the structure of communication and consensual interaction, not causal ingredients. Nor does mathematics provide any directly applicable notions: mathematical structures are abstractions, Platonic ideals, not fragments or constituents of activity. For example, in discussing two-factor semantical analysis in section 2 I talked about impressions being “causally efficacious”; these terms aren’t in the standard mathematical repertoire, nor are pure mathematical objects thought to possess causal powers. I have introduced the term ‘impression’ as a small step towards repairing this deficiency (as I did with ‘structural field’ in the 3-Lisp case), but of course it is simply a general, covering term. What we lack is a theory of types of impressions, types of important relations among impressions, analyses of how impressions can simultaneously cause and represent, and so forth. It is not that we are entirely without terms for such things: ‘data structures’, ‘data bases’, ‘knowledge bases’, ‘data types’, ‘functions’ (in the ‘procedure’ sense), and ‘code’ are all types of impression, as are more specific AI constructs such as semantic nets, inheritance graphs, and taxonomic lattices. Rather, what we need is a general theory, in terms of which these diverse kinds could be characterised.

Lacking a general theory, what do theorists do instead? Different things. Perhaps the most common practice, especially in AI and in the philosophy of mind, is to treat impressions metaphorically — in particular, as analogous to language. Thus in the cognitive case we have talk about "language of thought", "mentalese", "syntactic" theories of mind, etc., as for example championed by Fodor, Stich, and others [Fodor, 1975; Stich, 1985]. AI typically follows the same path, talking about "expressions", knowledge representation "languages", etc., as does anyone who views impressions as "formulae". In philosophy this stance is commonly referred to as the *representational theory of mind*, an unfortunate epithet not because the term 'representation' is inherently so narrow, but because this usage tends, without explicit admission, severely to constrain the notion of representation to its linguistic or even syntactic shadow. Instead I will call it a *linguistic theory of impressions*. Two facts about this theory are important for present purposes: (i) that we recognise its hypothetical nature, the fact that it represents a substantial claim; and (ii) that so long as this language remains metaphorical, we be careful to monitor connotations not necessarily warranted in the new domain.[8] For example, in 3-Lisp I called certain number-designating impressions *numerals*, but the metaphorical nature of the terminology misled me as well as others, causing me to attribute semantical properties to impressions motivated more by linguistic connotation than by genuine functional need (such as adopting a strict use/mention hierarchy, distinguishing 3, '3, and ''3).

People bred in the knowledge representation tradition may find the linguistic approach to impressions obvious, but it isn't universally accepted. As is well known, philosophical debates rage about whether this is the best way to characterise human mental states; more surprising, perhaps, is the fact that a number of alternative views are advocated even within computational circles. First, many people have realised, in opposition to the linguistic claim in its narrowest form, that there is no need for internal structures to be anything like *identical* to written ones. The mildest position of this sort is McCarthy's notion of "abstract syntax", which is really just a way to free impressions from gratuitous details of notation ('abstract' meaning to throw something away). I made a stronger move in the same direction in developing 3-Lisp, using the term "structural field" for the totality of impressions, even though I then described individual impression types using terminology derivative on linguistic analysis. This move was stronger than McCarthy's not only because the granularity of distinction in the 3-Lisp field was less than is usual in even abstract linguistic cases, but also because the mapping between expressions and impressions was contextually sensitive. (Partly for reasons of circularity and structure-sharing, the external notation was neither isomorphic to internal impressions, nor complete. Furthermore, in certain complex cases like closures, the impression structure was far more complex than linguistic notation could readily express.)

Other positions on impressions have been proposed. For example, there is increasing sentiment in various AI quarters that viewing impressions as syntactic or linguistic is not ideal — primarily because it commits the theorist to too fine-grained a set of internal distinctions. Two alternatives are of particular importance. Levesque [1984] retains his allegiance to knowledge representation as a covering notion, but argues for a functional analysis of machine states, with explicit reference to the notion of an abstract data type, as opposed to a view of them as comprising "collections of symbolic structures". In an apparently more radical step, Rosenschein [1985] criticises the entire representational stance, which he characterises as viewing "the state of the machine as encoding symbolic data objects", arguing instead for the notion of a situated automaton, with intentional properties (which he calls "knowledge") defined in terms of "objective correlations between machine states and world states".[9]

Supporting these anti-syntactical proposals, moreover, is the attitude towards impressions adopted in current theoretical computer science. It is an approach that is partially obscured, however, by a facade of potentially distracting theoretical technique. So I will digress from the subject of impressions, for a moment, to examine what computer sciences calls the denotational semantics of programming languages.

5. Programs, Processes, and Indirect Classification

The abstract data type movement in programming language design, and the denotational approach to programming language semantics, are best understood as attempts to characterise the structure of computational processes in other than linguistic terms. They are motivated by the following obvious fact: when we develop computational processes, we *cannot build processes directly*, but instead cause them to come into existence by writing programs. AI programmers often gloss the distinction between the program and the process, viewing programs as functional ingredients inside processes (i.e., take programs to be impressions). This assumption, partly motivated by the widespread use of interpreted, interactive languages like Lisp, is betrayed in such informal parlance as "the program is still running", "the program reads in a number and then prints out the answer", etc.

Nonetheless, *programs* — textual objects edited by EMACS and other editors, printed out on the screen, etc. — don't do anything; they are inert. Rather, these passive structures are used by interpreters and compilers (about which more in a minute) to engender behaviour with appropriate properties.

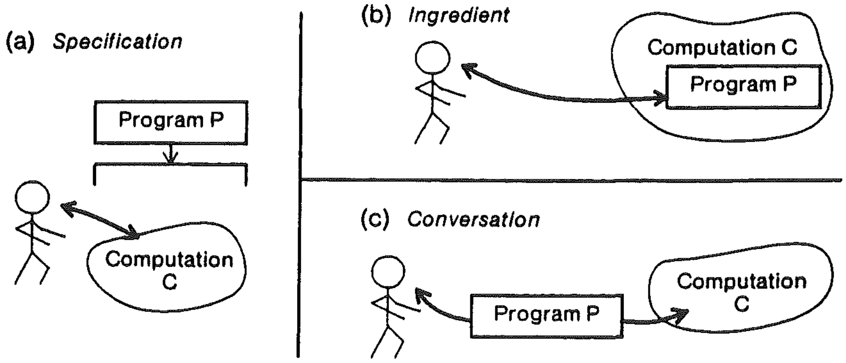

Figure 6: Three perspectives on programs

The situation is depicted in Figure 6. The AI and knowledge representation community typically views programs, along with elements of knowledge representation languages, as constituents of or elements within computational processes — i.e., as impressions. I will call this the *ingrediential* view, as suggested in (b). By far the more standard computer science conception, in contrast, is what I will call the *specificational* view, pictured in (a): programs taken as specifications or descriptions of computations, albeit as special descriptions that can be viewed as *prescriptions* by the machine or interpreter. (Different from both is a third, *conversational,* view that we will examine in section 6.)

Thus — and this will greatly affect our semantic analyses — traditional computer science takes semantics as the job of mapping programs onto processes (it is only under this conception, furthermore, that "interpreters" are properly named). Concerned as we are with knowledge representation, our task is different: to describe the relation between those processes and the worlds in which they are embedded. It follows that, in the traditional terminology, the *semantic* domains of traditional programming language analyses should be the knowledge representer's so-called *syntactic* domains. Confusion over this point amounts to the commission of a use/mention error — exactly the sort of thing that careful semantical analysis is at pains to eliminate.[10]

It may seem odd to look for impressions in the semantic domain of a semantic analysis of a programming language. Denotational semanticists, after all, typically deal in semantic domains consisting of abstract mathematical structures — functions, sets, numbers, partial orders, and the like — which don't seem very much like causally efficacious impressions. But this apparent discrepancy is explained by the fact that traditional denotational semantics is model-theoretic, and, as we've already seen, the model is not the true domain of interpretation, but some other structure, typically abstract, set in correspondence with it. As suggested earlier, this technique enables theorists at least partially to avoid exactly the metaphysical questions we are interested in: questions about the true nature of impressions themselves.[11]

Not all questions are avoided, of course, since the structure of the model is intended in some way to correspond to the structure of the impressions. The question is how the correspondence goes (i.e., what is the relation between a set-theoretic structure and an FKRL impression?). To get at the answer, note that modelling is an instance of the rather general practice of describing a set of complex phenomena only by setting them in relation to another, presumably more familiar, set of structures. Barwise and Perry [1983] call this "indirect classification". An observer establishes (perhaps implicitly) a relation between the domain in question and some other domain, and then describes particular phenomena in the first domain only with reference to some corresponding phenomena in the second. An obvious case is the folk classification of people's thoughts and beliefs: we describe what P believes by describing the situation that would be the case if what P believes were true. When I ask you what you think, you are literally incapable of saying, since we in English have no vocabulary, or intuitions, about the structure of what is inside people's minds. Rather, you are liable to say something like "I was thinking *that Palo Alto is too far from Finland*". The thought process, in other words, is described indirectly, by adverting to a fact (Palo Alto's being too far from Finland) that would be the case if my thought were true.

The examples we looked at in discussing model-theoretic semantics were just like this: we establish an association between something and something else, and get at the something else by referring to the something. So for example we set up a correspondence between Turing machine states and quadruples, which lets us describe a particular state with a particular quadruple. Furthermore, the example illustrates an important general property of all indirect classification: what is *specific* about a given entity in the primary domain is set in correspondence with what is *specific* about the corresponding entity in the classificatory domain. Thus you don't have to encode, in the domain of quadruples, the fact that Turing machines have tapes, or that the third element of the quadruple corresponds to the mark under the read/write head, or that the numbers 0 and 1 are being used to classify a mark or a blank, or anything else *that is true for all the relevant cases.* All that is required is that a *particular* quadruple contain enough information to determine the *particular* state, transition, etc., that it is being used to classify.

What distingushes the denotational approach to programming language semantics from arbitrary indirect classification, and leads to potential confusion, is the practice of *identifying* the classificatory entity with what is thereby classified. This identification isn't necessary; one could classify Turing machine #23 with quadruple #1437 without going on to claim a Turing machine *is* a quadruple. The identification is considered to be acceptable when the two structures are thought isomorphic, but isomorphism is always relative to an assumed metric of equivalence. In our case, where we have a second semantical factor (functional role) lurking in the background in need of explanation, we cannot afford to identify two things that differ in any way: what looks isomorphic from the point of view of representational import may be crucially different in terms of functional role. For example, as we have already pointed out, no abstract mathematical structure is even a candidate for the kind of efficacious causality we will need in order to connect impressions with action. Distinct but isomorphic mathematical structures may be used to classify embodied mechanisms with very different causal powers. So we need to proceed extremely cautiously.

We'll encounter further issues about modelling in the next section, but for now let's return to programming languages. First, since it provides the most freedom, is least biased with respect to impression structure, and is most compatible with current computational theory, in what follows I will informally adopt the specificational view of programs (in spite of its being contrary to the dominant view in AI and cognitive science). Thus programs will be viewed as inert linguistic entities, built up of expressions; processes, in contrast, are active, manifest behaviour, and are composed of impressions. Second, my claim is that the way to understand denotational semantics in computer science is as an analysis of the program–process relation that *indirectly classifies computational processes in terms of abstract mathematical models.* The situation is pictured in Figure 7.

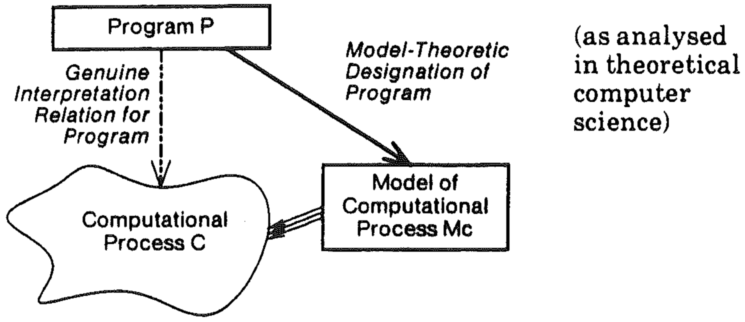
Program P
Genuine Interpretation Relation for Program
Model-Theoretic Designation of Program
(as analysed in theoretical computer science)
Computational Process C
Model of Computational Process Mc

Figure 7: The model-theoretic analysis of the semantics of programs

In terms of this picture, I can now explain the theoretical distraction I alluded to earlier, in introducing this section. It arises from the combination of two problems: failing to distinguish between the specificational and ingrediential views of programs, and being seduced by model-theoretic properties of the model (its abstract, mathematical character) into thinking it must model content. As a result, one is apt to identify the model M_C of the computational process C with the model M_W of the state of affairs W that the process is genuinely about, as shown in Figure 8.

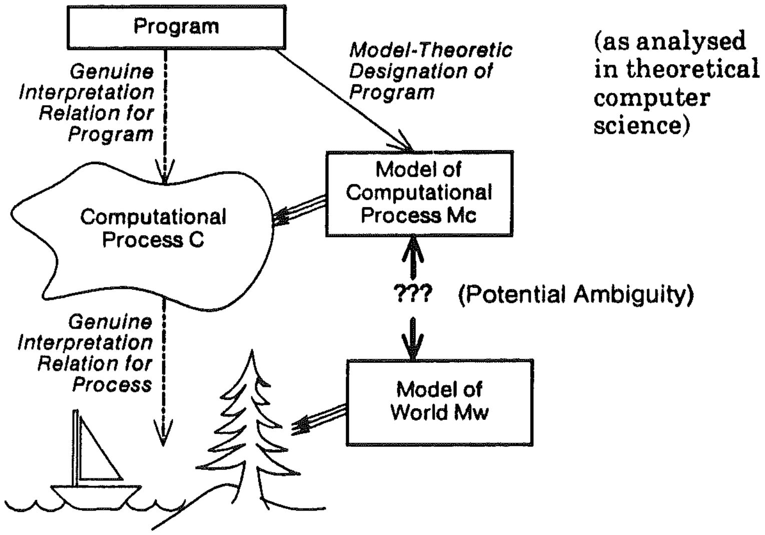
Program
Genuine Interpretation Relation for Program
Model-Theoretic Designation of Program
(as analysed in theoretical computer science)
Computational Process C
Model of Computational Process Mc
??? (Potential Ambiguity)
Genuine Interpretation Relation for Process
Model of World Mw

Figure 8: Models vs. interpretations of computational processes

The fact that the programming language tradition calls its analyses semantical, in other words, and the fact that their classificatory domain is abstract, may mislead AI researchers into thinking that the semantic domains of programming languages model the *content* of the computational processes that the programs engender. But this is false, at least in general. *There is simply no assumption, in the standard semantical analysis of programming languages, that computational processes are themselves semantic or intentional entities* — i.e., no *further* semantic relation is admitted or described. All that is explained is the relation between program and engendered computational process.

In the AI case, however, and particularly when dealing with knowledge representation systems, we assume that the ingredients inside the processes we are interested in, which we are calling impressions, are themselves intentional (this was the essence of our adopting a representational, as opposed to a merely functional, stance in section 2). Even if we were to adopt a model-theoretic approach in our semantical task, we would be interested in the relation between impressions in C (or in the model M_C) and the model M_W.

We have already said that there is no *a priori* reason to assume that these two models M_C and M_W will be the same. But a stronger thing can be said: if one assumes that M_C is an adequate model of process C, and that M_W is an adequate model of what C is about, then *to identify M_C and M_W is to assume that the representational import relation of knowledge representation systems is one of isomorphism.* Far from treating impressions as a language, this would be to treat them as a simulacrum of the world. Or to put the same point another way, to adopt, as a model of a knowledge representation system's semantics, a denotational analysis of the programming language used to specify it, is either to assume that the primary representation relation, between process and world, is one of isomorphism, or else — even worse — to ignore that relation completely (thereby maintaining a solipsistic stance towards computations themselves). Either way, this would be an unhappy result: false, and terrifically misleading.

Let's look at some examples. In purely mathematical cases, M_C and M_W may truly coincide. For example, suppose I write a program to calculate the factorial function. We may presume this literally means the following: that I write a program to specify a process that is about numbers and the factorial relation. In this case W is a structured domain of numbers and functions. Similarly, a denotational semanticist in computer science would almost surely use the same structures (numbers and the factorial function) as an abstract mathematical model (M_C) in terms of which to classify the process. Not only can M_C and M_W be identified; in this situation M_C, M_W, and W are identical.

This identity, however, relies on special properties of the exai.iple. Suppose in contrast that, in designing a robot to pull off bank heists, we represent (in FKRL) the fact that anything to the right of the robot is neither to the left of it nor straight in front. In order to motivate an appropriate M_C, we need to understand the relation between FKRL programs (now viewed as specifications) and FKRL impressions. So imagine the notation for FKRL programs was reminiscent of logical notation, and that we could write down something like the following:

$$\forall x\, [\, \mathrm{Right}(x) \Longrightarrow (\neg \mathrm{Left}(x) \wedge \neg \mathrm{Front}(x))\,]$$

Suppose, furthermore, that this FKRL expression is more specific than the corresponding impression in two ways. First, there is to be no fact of the matter, in the resulting impression, about what particular variable was used in the program; it might equally well have been y, or z. Second, although matters of lexical notation force one of the conjuncts to be first ($\neg\mathrm{Left}(x)$, in this case), we will assume that impressions are internally realised as unordered sets. Thus the following expression would have generated an indistinguishable impression:

$$\forall w\, [\, \mathrm{Right}(w) \Longrightarrow (\neg \mathrm{Front}(w) \wedge \neg \mathrm{Left}(w))\,]$$

Given these assumptions, we can then take on the task of providing a semantical analysis of FKRL programs — which is to say, an analysis of the relation between FKRL specifications and FKRL impressions — using the model-theoretic approach of indirect classification. It is unlikely that we would do no more than constrain the models of this impression to those that satisfy the logical implication, since we can presume that more fine-grained details of the impression's structure will play a functional role in licensing inference (such as the fact that the negation signs haven't been pulled out to the front, as they have in the semantically equivalent $\neg\exists w\, [\mathrm{Right}(w) \wedge (\mathrm{Front}(w) \vee \mathrm{Left}(w))]$). So we would probably be tempted to classify it using something like a term model, with the set of all equivalent expressions (i.e., all those with different variables and/or re-ordered conjuncts). It might be, however, that for some reason we were warranted in taking a more abstract approach, and, with reference to an interpretation function that mapped Right, Front, and Left onto three distinct unary predicates, and were to classify the impression in terms of the set of all models satisfying the given implication.

To relate this to Figure 8, let's call the quantified expression E, the engendered impression I, the first classification C_1, and second C_2, and the impression's interpretation W, as suggested in Figure 9. (The arrow goes through the head of the robot since the computational process, which includes impression I, is actually assumed to be its mind; the real interpretation relation is between that mind and the policeman in front of it.) It should be obvious, first, that C_1 and C_2 are both more abstract than E, in the information-theoretic sense of being less rich. Second, C_2 is in turn more abstract than C_1, since this model makes fewer distinctions (identifying all semantically equivalent expressions). Finally, both C_1 and C_2 have properties that are not properties of I itself, nor do they model or classify properties of E (or W). For example they are both sets, even though none of E, I, or W is.

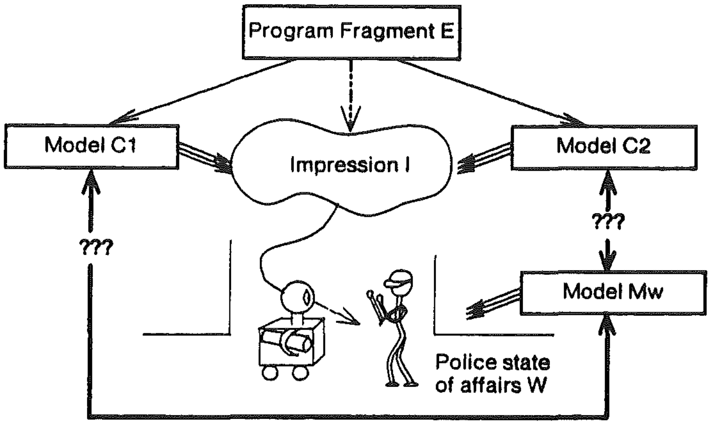

Figure 9: Indirect classification of the semantics of FKRL programs

The question, then, is whether either of C_1 or C_2 is a candidate for being a model of W, the representational import of I. And the answer — to bring this all home — is *no*. The problem is that "to the right of", in the world, isn't a one-place relation: something is "to the right" only relative to the position and orientation of the robot itself. Thus C_2 won't do as a model of representational content, since it doesn't contain enough information to determine, for example, whether the impression I is *true*. If we wanted to model W, then various additional circumstantial factors — including the position and orientation of the robot — would have to be brought in explicitly, since in dealing with W we need to deal with actual position in the world (that's where we find the police).

There isn't any formal problem with adding circumstantial parameters; we saw how to do that in section 3. The point, rather, is that these circumstances *affect the semantic relation between process (I) and world (W), not the relation between program (E) and process (I).* In fact it is crucial, in order to get at the proper regularities in the process, that the circumstantial relativity *not* be included in the wrong place. It is far more likely that the machine's behaviour will revolve around regularities framed in terms of what's in front of it, to its right, or to its left, not in terms of what is in a given position. If the robot's external circumstances were introduced in the E–C_2 relation, then the resulting C_2 would fail, as a model of I. For example, it would be of no help in explaining matters if I somehow broke and caused the robot always to ignore things on its left, since "on its left" wouldn't be a notion in this modified C_2.

In general, of course, nothing prohibits a theorist's classifying something by its content (as we did in the factorial case); in fact that, arguably, is exactly the mechanism underlying the propositional attitudes of folk psychology ('knows that', 'believes that', 'hopes that', etc.).[12] The point is only that we must not assume that *all* indirect classification is of this type. More seriously, it is not, by and large, the right way to understand the impression structure of circumstantially dependent agents.

6. Impressions, Expressions, Complications

I said in section 4 that there isn't any accepted, direct way of describing impressions. So far we have seen two quite different alternatives: a metaphorical approach, using the language of linguistic expressions (section 4), and an indirect approach, classifying them in terms of abstract mathematical structures (section 5). Before leaving the subject, we must recognise a third. It is common in informal practice, and standard in the programming language approach called 'operational' semantics, to describe the impressions and behaviour of a given computational process in terms of the corresponding impressions or behaviour of a lower-level machine on which the process is implemented. This relation is depicted in Figure 10. For example, we might describe FKRL impressions by presenting the Lisp code we have used to implement them.

In some sense this approach just causes the problem to recur, since questions remain about how to describe the implementing machine, but it is often possible to refer either to a familiar underlying machine,[13] or to model the input/output behaviour of that machine in terms of ordinary mathematical functions. The relation between traditional denotational and operational semantics of programming languages, therefore, is primarily one of abstraction: the denotational account, by using coarse-grained functions as classificatory devices, gets at less detail than the operational account. Nonetheless, it is standard practice to prove the two types of account equivalent, strong evidence that they are different theoretical ways of getting at the same phenomenon: the nature of the computational process itself (and not at its semantic import!).[14]

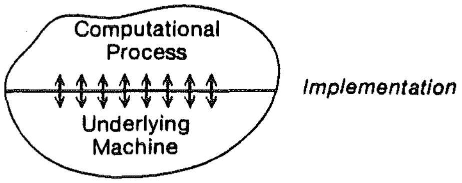

Figure 10: The implementation relation

The importance of this third approach, for our purposes, is its introduction of implementation as yet another intentional relation for semantical analysis to contend with. As with representation and belief, implementation is a directed, asymmetric, intentional notion: to say of X that it is an implementation is to imply a Y such that X is an implementation *of Y*. Furthermore, the implementation boundary is opaque to other semantical relations — i.e., it cannot be viewed as invisible modelling, or easily composed. For example, if we implement FKRL impressions in Prolog, and if the representational import of Prolog impressions can truthfully be given as standard first order model-theoretic semantics,[15] then it would not follow that the representational import of FKRL was the representational import of Prolog. At best the *interpretation* of Prolog impressions — the elements of Prolog's semantic domain — would be *FKRL impressions themselves.*

It is almost time to summarise the distinctions we've made, and assemble a coherent overall picture. Before doing that, however, we must tie up two loose ends. First, in the previous section we distinguished the representational content of impressions from the entities that theorists use to classify them indirectly, identifying a *modelling* relation between the two. But we haven't yet taken this observation to its obvious conclusion: modelling, like representation, specification, knowledge, implementation, etc., is itself a semantic, intentional, notion. Like many other things we've seen, a model isn't a model all on its own; models are models *of* something. A balsa airplane, for example, might be a model of a real airplane no longer around, or of one being designed. Similarly, the sets of quadruples we've talked of are models of a Turing machine; 0 and 1 are often used as models of Truth and Falsity. Thus we need, ideally, to give a semantical analysis of the modelling relation, if techniques of modelling or indirect classification are ever used. I.e., in the terms of Figure 9, we need semantic analyses of the C_1–I (or C_2–I) and M_W–W relations, as well as of E–I and I–W.

Second, all the computational processes we have looked at so far are limited in the following obvious way: we've imagined them acting in the world (driving around, computing factorial), but we haven't provided them with any *communicative* abilities. They can't talk. In order to be realistic, therefore, we should complicate our pictures yet one more time, as indicated in Figure 11. In order to contain the complexity, I've omitted all models and indirect classification from the diagram, showing only the genuine intentional relations that actually obtain in a given case. I will use the general term *notation* for the relation between expressions and impressions that they give rise to or express, and the more specific *internalisation* or *externalisation* to get at each direction of information flow. The analog, in the human case, is the relation between the sentences we speak and hear, and the impressions in our minds (mentalese or whatever) to which they correspond. To the extent that impressions are viewed linguistically, internalisation might be analysed as a species of translation, but it is important not to bias terminology in advance.

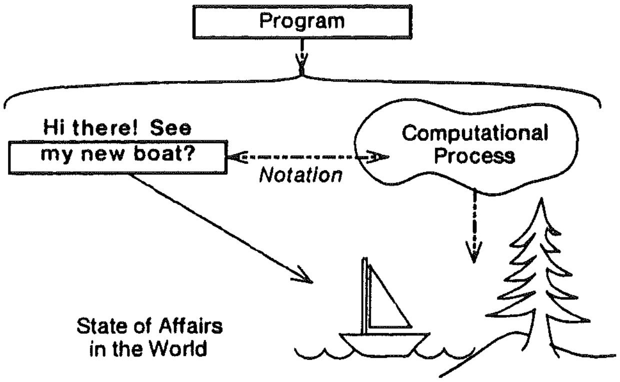

Figure 11: Programs that specify communicating agents

Issues of notation tie back to an issue we left unresolved above. Very often, the languages computer systems "speak" — query languages for data bases, editing commands for word processors, manipulation protocols for spread sheets — are visibly distinct from the programming languages used to create them. Many AI programming languages, however, such as Lisp, Smalltalk, Logo, and recent versions of Prolog, are primarily interactive, suggesting the third model of programming suggested in Figure 6 (c), above. Furthermore, the increasing incidence of "user-friendly" computers suggests that this interactive model of computer language will only spread In addition, since it is the correct model for natural language, people will be biased towards an interactive stance to the extent that people understand computer languages by analogy to their native linguistic skills. Thus we have a genuinely triple ambiguity in the term "program", which only raises the chances of true semantic confusion. Ironically, confusion between the specificational and interactive models of programming, coupled with the fact that the program–process relation is mediated by what is called an interpreter, has lead many computationalists to think of internalisation as the fundamental semantic relation — thereby embracing exactly the view that Lewis deridingly calls "markerese semantics" [Lewis, 1972]. On the other hand, AI practice suggests what Lewis's analysis does not: that internalisation is a substantial intentional relation in its own right. If nothing else, more adequate vocabulary might facilitate better interdisciplinary communication.

We are ready, then, to summarise four major themes in the investigation so far. First, we distinguished functional role and representational import, and set ourselves the long-range goal of an integrated account of full significance, consisting of partially independent but coördinated accounts of each semantical factor. Second, we claimed that we don't yet have adequate vocabulary for talking directly about impressions, and therefore avail ourselves of three alternative approaches: (i) using metaphorical terminology, such as the language of linguistic expressions; (ii) using indirect classification, typically in terms of abstract mathematical structures; and (iii) abstracting over implementations, which makes the problem recur. Differences among these alternatives, and differences in the fields in which they are popular, have obscured our ability to agree on underlying impression structure itself.

Third, setting aside considerations of functional role, we identified the following important relations, each at least a candidate for its own semantic analysis:

1. The *specification* relation, between a program and the process or impressions it engenders;
2. The *internalisation* and *externalisation* relations, between expressions used by a system to communicate with its users, and the impressions they give rise to or express;
3. The *implementation* relation, between impressions at one level of description, and other lower-level impressions in terms of which they are implemented; and

4. The *primary representation* relation, between impressions (process) and the states of affairs in the world with which the agent is concerned.

All of these four can be called *genuine*, in the sense that they are all a necessary part of the life of the representational agent in question — they haven't been posited solely for purposes of theoretical analysis. Other relations between the same structures could be added, of which the most important is probably the relation between communicative expressions (language) and the world — the subject, in the human case, of natural language semantics. I will adopt these four relations, however, as primary, because they are all candidates for full two-factor accounts. Put another way, they are all of a *causal* nature, in a way that the direct relation between language and the world is not. Note also that impressions participate in all four relations (which puts extra pressure on our ability to describe them in their own right), being the semantic domain in the first three, the so-called "syntactic" domain only in the last.

As a fourth theme, in addition to identifying these genuine semantical relations, we uncovered numerous relations of modelling or indirect classification, cross-cutting all of the above. To distinguish them from the genuine relations, I will call them *theoretic*, since they are introduced for the purposes of us, the theorists, rather than for the agent itself. Nonetheless, if we as theorists employ them, they too must be semantically understood. If we were to use model-theoretic techniques to understand the four genuine relations listed above, we would bring to eight the total number of interacting correspondence relations. The complexity can get a little daunting. It is no wonder that it is sometimes hard to tell, when presented with a "semantic analysis", just what it means.

All these results contribute to the general series of challenges I am mounting against straightforward model-theoretic semantics. The first specific challenge was implicit in our two-factor analysis itself, and its concomitant rejection of the independence of functional role and representational content. The second arose when we removed the constriction that impressions be syntactic or linguistic in nature, and embraced instead a much wider range of representational possibilities. The third challenge stems from the multitude of genuine intentional relations just cited — specification, internalisation, implementation, representation, etc. — more than one of which will require its own two-factor analysis. The fourth derives from the fact that standard theoretical techniques of indirect classification and modelling introduce, at the level of theory, a whole spectrum of additional correspondence relations, at least distractingly similar to semantic relations, if not semantic relations in their own right. If we don't understand them they will pollute our attempts to clarify the semantic relations we are primarily interested in.

Nor are we done raising challenges. In the next section I will turn to a fifth, coming to a sixth at the end of the paper.

7. The Correspondence Continuum

I said in section 3 that the model-theoretic tradition characteristically assumes a non-transitive denotation relation, motivated by clear linguistic cases: an English description of a French description of dessert, for example — such as "the four words *neige*, *la*, *à*, and *oeuf*, in reverse order" — is a description of language, not a description of something to eat. At the same time, we saw traditional analyses freely compose modelling relations, as for example when a number encoding a description of a Turing machine is identified with the Turing machine in question. This free composition goes hand in hand with modelling's traditional invisibility.

Unfortunately, however, these two cases — non-transitive denotation, and transitive modelling — don't cover the whole spectrum of semantic relations. In the general case, intentional relations combine in much more complex ways. We'll look at three examples. First, suppose I remark on a photograph you have taken of one of my favourite sailing ships, and you then present me with a copy made by photographing the original. It would be pedantic for me to maintain, on grounds of use/mention hygiene, that the copy is not a photo of the ship, but rather a photo of a photo of a ship. For most purposes, the relation between the copy and the original print is sufficiently close that I can harmlessly compose the two correspondence relations (copy–original and original–ship), yielding a result (copy–ship) essentially identical to the second. But not for all purposes: if, on close inspection, I claim that there is a tear in the ship's sails, you might appropriately reply that *no*, the tear, rather, is in the original photograph that the copy was made from. Or I might be interested in the quality of your photographic technique, and use the copy as a representation of your original work. The ability to compose, or to "look through" it to what is represented, can depend on the purpose to which a a semantic relation is put.

Second, imagine connecting an FKRL system to a visual recognition system, consisting of a TV camera, special-purpose line-finding hardware, a figure-recognition module, etc. In such a case one might be tempted to say that the configuration of pulses on the cameras represented in the intensity of incoming light, and that the resulting FKRL impression represented the object under view. Yet although the former objects play a causal role in supporting the latter, it's not clear how the two representation relations fit together — the second seems to "leap completely over" the first. In spite of systematic correspondences among the constituent structures, the representation relations seem curiously independent. It's as if the structural correspondences compose, but the representation relations don't.

Third, in designing 3-Lisp, I distinguished impressions called numerals from canonical impressions denoting them, in spite of the fact that the denotation relation was an exact isomorphism. I did so because, trained in avoiding use/mention confusions, and viewing impressions as analogous to language, I thought representation relations *could not* compose. Various colleagues suggested that this strictness bordered on pedantry, and recommended that I simply identify the two impressions. Others even suggested that I identify both of them with the number designated, since as far as they could see the impression–number relation was also one of isomorphism.[16] But my allegiance to semantic strictness was strong: as shown in Figure 12, I refused to say that the two-character expression written 23 represented the number twenty three; rather, when speaking carefully, I said that it *notated an impression that designated that number*. Similarly, I was forced to say that '23 notated a handle impression that designated a numeral impression that designated a number. And so on and so forth.

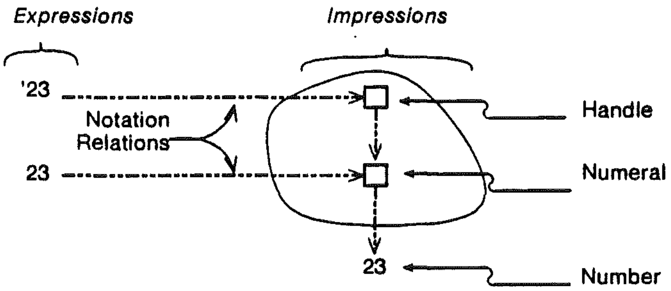

Figure 12: 3-Lisp's plethora of representation relations

3-Lisp was certainly semantically clean, but in retrospect some of its rigidity seems gratuitous, even if I remain opposed to any *identification* of strings with impressions, or of impressions with numbers. It is overwhelmingly convenient to be able to point to a figure on a computer screen and say, simply, that it represents a number (furthermore, one might even be *correct* in doing so). And yet at the same time there are occasions when it is crucial to distinguish among expressions, impressions, and numbers.

All of these examples illustrate my fifth challenge to traditional model theory: neither strict non-transitivity, nor indiscriminate identification, is always appropriate. In each cited case, as so often happens, theoretical technique isn't up to the demands of practice. The true situation is more accurately pictured in Figure 13. The idea is this: a given intentional structure — language, process, impression, model — is set in correspondence with one or more other structures, each of which is in turn set in correspondence with still others, at some point reaching (we hope) the states of affairs in the world that the original structures were genuinely about.

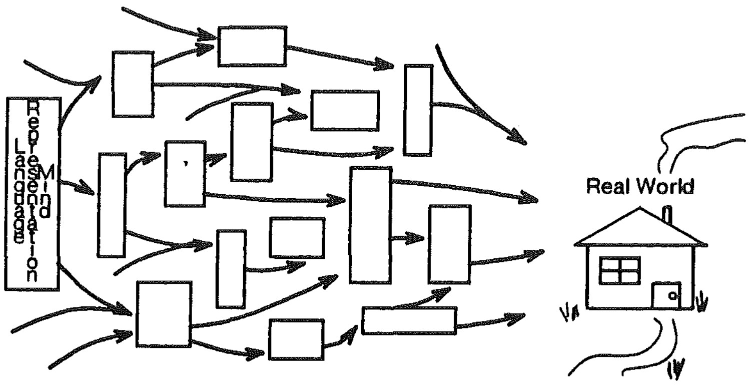

Figure 13: The Correspondence Continuum ("Semantic Soup")

It is this structure that I call the 'correspondence continuum' — a semantic soup in which to locate transitive and non-transitive linguistic relations, relations of modelling and encoding, implementation and realisation, the rest. Several points are important. First, I will not presume, in the general case, anything about composition, relative structure, circumstantial dependence, or any other traditional issue: such questions will have to be answered individually, based on particular facts about specific cases. Sometimes, and for some purposes, these representation relations will happily compose; other times not. Sometimes *some* properties (such as ambiguity!) will be preserved even across a whole string of such correspondence relations, even though other properties (such as one-to-one correspondence of objects) are lost. In the next section I will begin to sketch out an analysis of correspondence relations that will show how this might go.

Second, one shouldn't think of this as necessarily a single dimension; the diagram is meant to be able to accomodate the multiple dimensions of representation (notation, representation, specification, etc.). As we have just seen in 3-Lisp's case, and as we saw so often in the last section, part of the task, in analysing the semantics of computational processes, is to tie together different correspondence relations that are neither totally independent, nor arranged in a simple linear order.

The general picture given in Figure 13 is intended as a replacement for the simplistic diagram of Figure 2, even for the most basic intentional relations. In the remainder of the paper I will try to address a few of the numerous questions it raises.

Here's one, for starters. Which, if any, of these correspondence relations should be counted as genuinely semantic, intentional, representational? Surely not all. For example, to take another visual example, at the very moment I write this there is a series of correspondences of some sort between the signal on my optic nerve, the pattern of intensity on my retina, the structure of the light waves entering my eye, the surface shape on which the sunlight falls, and the cat sitting near me on the window-seat. And yet it is the cat that I see, not any of these intermediary structures. A causal analysis of perception, that is, would require a cascade of correspondences, but in this case only the full composition, *but not any of the ingredients*, would count as a genuine representation (though it doesn't follow that these intervening structures are thereby any less important). Similarly, even if I indirectly classify impressions with functions from possible worlds to states of affairs, and then map those mathematical structures onto genuine situations in the world, the agent itself will attend only to the situations in question, entirely unaffected my abstract classifying structures.

Both of these cases, and many of the phenomena cited in the last section, suggest that the number of important correspondence relations greatly outstrips the number that are of a genuinely semantic or intentional nature. Such arguments lead to a simple and obvious conclusion: *correspondence is a far more general phenomenon than representation or interpretation.*[17] First, it permeates theory, in terms of indirect classification and modelling. Second, it permeates practice, as manifested in such notions as implementation, encoding, realisation, presentation, specification, internalisation, and externalisation, as well in as our initial concerns of representation and knowledge. Third, although not all these correspondence relations should be counted as fully intentional, there is no chance that we will understand semantics unless we are first clear on how they all fit together. So my recommendation is that we peel correspondence away from more difficult semantic issues, and make it a subject matter in its own right.[18]

Let's look, then, at what a theory of correspondence might be like, before returning to semantics and to knowledge representation.

8. A Sketch of a Theory of Correspondence

In broad outline, the structure of correspondence is quite simple. First, two domains are identified, which I will presume consist of a pre-determined collection of situations, objects, properties, and relations. I'll call them *domain* and *co-domain* (though this isn't category theory), and say that an element of the domain *corresponds to* an element of the co-domain. Furthermore, without introducing any assumption of symmetry, I will speak most generally of correspondence relations, rather than functions, and make room for circumstantial parameters in the usual way. The situation is pictured in Figure 14. (The resemblance to Figure 2 is obvious; we can now see that Figure 2 was right for correspondence, wrong — because too simple — for semantics more generally).

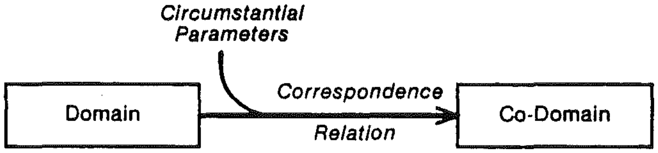

Figure 14: The general structure of correspondence

Given these two domains, specific correspondence relations are defined between states of affairs in each domain — not between the domains themselves, nor between objects, properties, or situations on their own, but between things *being a certain way* in one domain, and things *being a certain way* in the other. Thus, the light's being red corresponds (or so we hope) to cars' stopping. Similarly, we might say that the sequential concatenation of the numeral 2, the sign +, and the numeral 3 corresponds to the addition function's being applied to the numbers 2 and 3, which in turn corresponds to the number five.[19] Even in cases where there is a simple correspondence of objects, as when the numeral 3 stands for the number three, it is really the object's *being that and not some other numeral* that corresponds to the number's *being that and not some other number.*

The numeral may have all sorts of other properties — such as consisting of one curved and one straight line — which don't correspond to anything in the co-domain at all.

There are several reasons to require an explicit specification of domains, and to lay responsibility for the correspondence relation on states of affairs. In general, objects exemplify infinitely many properties, and participate in infinitely many relations — in this sense the world is overwhelmingly rich. Even questions of object identity don't escape this richness, as precise attempts to define numerals quickly reveal (does the expression '124 + 124' contain 1, 2, 3, or 6?). It is therefore necessary, in characterising a particular correspondence relation, to identify in advance the particular set of objects, properties, and relations in each domain that are constitutive of the significant states of affairs — what I will call a prior *registration* of the domains — and then to identify, with reference to that registration, how states of affairs in the domain correspond to states of affairs in the co-domain. This is partly because states of affairs, at least as I am using the notion,[20] are individuated by the relations and properties they instantiate (the states of affairs of a number's being the sum of 2 plus 2, and of its being the positive square root of 16, are different). But it also seems true to common sense, as the red light example suggests.[21]

(As well as adopting these two theoretical assumptions, there is another which I will explicitly set aside. Many writers, including theorists as far back as Peirce, have expressed the deep intuition that representation is a 3-place, not 2-place, relation, involving not only representation and represented, but also *interpreter*, *observer*, or, in Peirce's case, *interpretant*. Thus a text, and probably even a simple map, is taken not to be a representation on its own, but to represent only for some other agent or purpose (or both). I sympathise, in the representational case, but we are talking here about a simpler notion of correspondence, where the question is much less clear. For example, one could view a binary correspondence relation between X and Y as a relation that an interpreter posits or reacts to, in taking X to represent Y. Thus your map may not represent New York unless you or some other person takes it to do so, but that act of taking it to represent New York involves attributing or establishing a *binary* correspondence relation of a certain type — of a type, furthermore, that might be characterised in terms of the theory I am proposing. In addition, given my general recognition of the importance of circumstantial dependence, it isn't obvious that the role of interpreter has been excluded. But however this goes, my present purpose is to define a project, not report on its conclusion. Such questions should ultimately be answered by theory, not prejudged.)[22]

I will call the relevant states of affairs in the domain and co-domain the *source* and *target*, respectively. So the source expression “72° 10' E, 44°20' N” might correspond to a bucolic target in northeast Vermont. Correspondence relations will, in general, be defined in terms of source and target *types*, in such a way that instances of the source type would correspond to instances of the target type in some determinate fashion. For example, the mapping from sets of quadruples to Turing machines would be established so that a particular quadruple's having certain elements would correspond to the controller of the corresponding Turing machine's satisfying a particular transition function. This approach makes sense of the intuition about modelling suggested in section 5: that what is *specific* about one state of affairs (source) determines what is specific about another (target).

In setting out an initial analysis of this sort, [Smith, forthcoming(c)] I call a particular correspondence relation *iconic* if each object, property, and relation in the source corresponds, respectively, to an object, property, and relation in the target. A particularly important case of iconicity occurs when a source object, property, or relation corresponds to itself in the target: I will say in such a case that the target structure is *absorbed* in the source. For example, left-to-right adjacency is absorbed in the grammar rule EXP ::= OP(EXP,EXP) for a simple term language for arithmetic. Similarly, to suppose that the necessity of set membership, in a model-theoretic analysis of modality, models necessity in the world is to assume, counter-factually, that necessity is absorbed. In contrast, a target property or relation is said to be *reified* if it is corresponded to by an object in the source. Thus for example the syntax of predicate calculus reifies predicates because it represents them with (instances of) predicate letters, which are registered as objects, at least in standard syntactical analyses.

A correspondence relation is called *polar* when a positive source (something's being the case) corresponds to a negative target (something's not being the case), or vice versa. Hotel lobbies provide an example, where a key's being present in the mail slot indicates the fact that the client is gone. A relation is called *typological* if it can be defined without reference to distinguished individual objects in the domain or co-domain. Thus the standard Cartesian relation of ordered pairs of real numbers to points on a plane fails to be typological on four counts: origin, orientation of x-axis, unit length, and something to distinguish left and right orientation, such as a normal to the plane. Finally, when either or both domains are analysed mereologically — in terms of notions of part and whole — either or both ends of the correspondence can be defined *compositionally*, in the sense that what corresponds to (or is corresponded to by) a whole is systematically constituted out of what corresponds to (or, again, is corresponded to by) its parts. If the part/whole relation is itself absorbed, a very strong version of compositional correspondence obtains, where parts of a source correspond to parts of that source's target.

Many other such relations can be defined, ranging from this simple sort up through more complex cases having to do with sentences, quantification, use, circumstantial dependence, etc. The intent here is not solely to develop a theoretical typology (though that is often useful, especially early in theoretical development), but eventually to identify an algebraic basis of correspondence in terms of which to analyse arbitrary relations. Given such an algebra, for example, and an analysis of two relations C_1 and C_2 in terms of the orthogonal set of basic features, it should be possible to predict the exact structure of the composed relation $C_1 \circ C_2$. Thus we would expect the composition of two iconic relations to be iconic, iconic relations to be both left and right identities (with respect to this algebra), and so on and so forth. Note, however, that the appropriateness conditions for composition are very strong: $C_1 \circ C_2$ makes sense only if the targets of C_1 are of *exactly the same type* as the sources of C_2. Traditional isomorphism won't do, since isomorphism is just another correspondence relation C_3; the combination would have to be analysed as $C_1 \circ C_3 \circ C_2$.

As the isomorphism example suggests, a correspondence theory of this sort would provide theorists (I primarily have semanticists and computer scientists in mind, but of course the account would be general) with an extraordinarily fine-grained pair of glasses with which to analyse arbitrary structured relationship between domains. Every conceivable coding, representation, modelling, implementation, and isomorphism relation would be made blatantly visible. Whereas category theory can be viewed as highly abstract, in other words, correspondence theory would be exactly the opposite: unremittingly concrete. This doesn't mean that abstract objects couldn't be studied within such a framework, of course, only that *no further abstraction by the theory* would be permitted unless explicitly accounted for (beyond that provided by the initial registration of the domains). Thus, whereas a model-theoretic analysis of the interpretation of the English word 'cat' might map it onto a mathematical set, a correspondence-theoretic account could not do so. You could have a correspondence-theoretic analysis of the relation between the word 'cat' and the set-theoretic structure used by model theory to classify it, but that is quite a different thing.

It is a consequence of this granularity that many standard mathematical techniques, such as that of identifying structures "up to isomorphism", would be inapplicable. But this result is to be expected: since the whole point is to avoid gratuitous modelling, and to explain arbitrarily fine-grained distinctions, the theory cannot indulge in any loss of detail.

As well as focusing on the detailed structure of specific correspondences between states of affairs, an adequate theory would have to address general questions about particular relations, such as whether every source in the domain corresponded to exactly one target, whether every target had a source corresponding to it, etc. It would be natural, that is, to define correspondence versions of such standard notions of totality, completeness, and ambiguity. But this starts to feel a little odd, because of its familiarity. Are we just reinventing traditional mathematical accounts of functions and relations? How do our categories of correspondence relate to such standard notions as isomorphism, homomorphism, injection?

The answer appears to be the following. It has often been pointed out that standard so-called *extensional* analyses of functions and relations, in terms of piece-wise pairings, ignore the structure of the connection between the domain and co-domain, even though that structure is often important in practice — such as when the function is to be computed, or the relation recognised, or when the connection is *causal*, defined in terms of the constituent properties. When we write functions down in natural or formal languages, however, or embody them in machines, we typically betray a great deal of additional information. Thus the standard term designating the factorial function

```
if n = 0 then 1 else n*factorial(n-1)
```

implicitly suggests a way of computing factorial, even though that information is lost in the standard extensional analysis, which would map this term onto the infinite set of ordered pairs <0,1>, <1,1>, <2,2>, <3,6>, etc.

In the general case the information conveyed by a functional description can be sorted into three kinds, as suggested in Figure 15: information about the structure of the domain, about the structure of the co-domain, and about the structure of the relation between the two (the first two clearly merge when, as is often the case for simple functions, the domain and co-domain are the same).

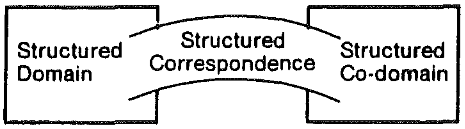

Figure 15: The three structures of correspondence

Recognising the importance of this other information, various people have attempted to develop what are called *intensional* analyses of functions, relations, etc. The idea, or so it's claimed, is to make this extra information explicit. But from our point of view there is something curious about the way in which this is traditionally done. Because these efforts have arisen in the context of computation, recursive function theory, and a general concern with procedures, the approach is in fact not one of making these three kinds of information explicit, but rather of making explicit *the structure of an algorithm for computing the function* (or relation). Thus Moschovakis [1984] has proposed treating an algorithm as a first class mathematical entity in its own right, and a variety of writers have at least argued for dealing directly with procedures, such as those recommending procedural treatments of semantics [Woods, 1981].

Obviously there's nothing wrong with explicating the notion of an algorithm. But there is no reason to suppose that this project, even if successful, will make explicit the three kinds of information cited above. For example, no matter how explicit I am in giving you directions for driving across Boston, the structure of the city will at best be borne implicitly in the resulting descriptions of routes. Imagine trying to reconstruct a Boston city map by sorting through every route travelled by a long-time cab driver, gradually culling information about the town from such sequences as "drive two blocks up Trapelo Rd, turn right on Grove", or heroic attempts explain how to get from Somerville to the airport without using the tunnel. Making the *algorithm* specific won't even make explicit the structure of the relation it computes, let alone the structure of the related domains.

In contrast, a correspondence theory can be viewed as almost a dual project: it would provide an informationally rich account of the structure of the relation between structured domains, though it would remain entirely silent on any question of *computing* this relation. It would get at the three relevant structures (of domain, co-domain, and correspondence) directly, rather than taking them to be indirectly manifested by specific ways of going from a given domain element to its corresponding co-domain element.

As for which project has a better claim on being an "intensional" analysis of functions and relations, I cannot say — nor, presumably, does it matter. For one thing, the very theory of correspondence I am proposing will among other things obviate the worth of such terms as "intensional" and "extensional". More important is to recognise the essential difference, and compatibility, between the two accounts. As suggested in Figure 16, the distinction between fine-grained ("intensional") and coarse-grained ("extensional", or piece-wise) analyses is orthogonal to the question of effectiveness or computation. We can thus classify the standard set-theoretic model of functions and relations as coarse-grained and non-effective, recursive theory as coarse-grained but effective, and the theory of algorithms as fine-grained and effective. A theory of correspondence then occupies its rightful place as the fourth possibility: a fine-grained but non-effective theory of relationship.

	Computational	*Non-Computational*
Fine-grained	Algorithms	**Correspondence**
Coarse-grained	Recursive Functions	Functions & Relations

Figure 16: Analyses of relationship

The location of a correspondence theory in this diagram is well suited to the semantic purposes for which we have needed it. One of the most fundamental facts about most genuine semantic relations, such as reference, is that *they aren't computed*, in any coherent sense of that word. When I say "Bach died in 1750", and thereby refer to a long-dead composer, nothing *happens* in order to make the reference work; it just *is*. It is thus entirely to be expected that semantical examples should push us towards a fine-grained but non-computational analysis of structured correspondence.

9. Semantics Revisited

The availability of a correspondence theory would change semantical analysis in at least these ways:

1. As promised, the following traditional notions would be replaced: a strict hierarchy of (meta-)languages, invisible but promiscuous modelling, and the notion of an absolute use/mention distinction.
2. It would provide the theorist with sufficient equipment to analyse such otherwise unanalysed notions as encoding, and to discern and thereby avoid problems of gratuitous artifacts.
3. It should provide, for the first time, adequate vocabulary in terms of which to analyse and assess such non-linguistic representational structures as images and analogue representations.

4. It enables us to explain some lurking problems and unexplained worries that have plagued traditional approaches.

I will look at each of these briefly.

First, in dismantling the absolute use/mention distinction I don't propose to license automatic composition of all correspondence relations. Rather, the intent of the algebraic basis of correspondence, sketched in section 8, is to enable us to see what sorts of properties will propogate through iterated correspondences, which ones won't. The popular closed-world assumption, for example, is in essence an assumption that object identity is absorbed; it should be straightforward to verify whether this property is preserved across one or more correspondence relations in question. Similarly, the assumption that words have referents could be justified, even by someone committed to the logical priority of mental impressions, just in case the internalisation and representation relations could be unproblematically combined. Even in written natural language, use vs. mention apparently shades off into matters of degree; thus we have: Bean Town is lively; Bean Town is so-called for historical reasons; they call it *Bean Town*; "Bean Town" is what he said; "Bean Town" is a nickname (but not, curiously, "Bean Town" is smudged).[23] Particular analyses of use and mention would depend on the semantic relations employed; once again letting go of the strict theoretic distinction paves the way for accomodating a wealth of familiar facts.

As well as undermining use/mention distinctions, the correspondence continuum challenges the clear difference between "syntactic" and "semantic" analyses of representational formalisms — an especially important consequence given the allegiance the distinction commands. On the face of it, it might seem that we are simply removing an important method of discriminating accounts, which would be a negative result; the claim, though, is that no simple "syntactic"/"semantic" distinction gets at a natural joints in the underlying subject matter.

For example, many writers have claimed to provide semantical analyses using models set-theoretically constructed out of basic syntactic elements such as sentences, ground terms, etc. A typical AI case is found in Moore and Hendrix's proposal for a semantical model for belief [Moore and Hendrix, 1979]; similarly, term models are often used in giving semantical analyses of logic-based programming languages, such as in Goguen and Meseguer's EqLog [1984]. Although stamped with the official "semantics" insignia, they are often used as abstract models of (i.e., to classify) syntactic or computational properties, such as interreducibility of terms in a rewrite system (α-interconvertibility in the λ-calculus, for example), effective derivability, etc.

My point is not to indict this practice, nor even to dispute its semantical claim. Rather, the point is this: *if* one is commited to a simple binary "syntactic"/"semantic" distinction, as on the traditional view, then such proposals would have to be counted as syntactic, and hence as false advertising, since for example the semantical interpretation of a formula such as DEAF(BEETHOVEN) would have only to do with syntax, nothing to do with the old man himself. On the continuum view we are proposing, in contrast, there is plenty of room for just such analyses as these. Of course there is a price: virtually nothing, on this account, follows immediately from labelling an analysis semantical. Rather, the theorist should make plain exactly what kinds of facts or properties the models in question are being used to classify, or what kinds of semantic relation are being analysed: computational, representational, whatever. But — and this is the important point — the space of possibilities is not constricted in advance by the nature of the theoretical framework.

The second main semantical consequence of the new approach arises from its fine-grainedness, which thereby facilitates direct views onto otherwise invisible relations. These last fall into two kinds: subject-matter relations that have heretofore evaded satisfactory analysis, like encoding and implementation, and theoretic relations like modelling, that have affected and sometimes distracted analysis. With respect to this fine-grainedness of approach, correspondence theory can be understood, in its relation to traditional semantics and model theory, as analogous to the relation between situation theory [Barwise **1986a**] and traditional set theory. In both cases, the classical system makes far fewer distinctions than at least some analyses demand. Thus situation theory, like other property theories, populates the world with properties, relations, facts, states of affairs, and the like, thereby embracing a much richer ontological foundation than the set theory we are used to. My brief against traditional model-theoretic analyses of languages and modelling is similar to Barwise and Perry's against set theory: it glosses much of the very detail we need to understand. The two enterprises of situation theory and correspondence theory, furthermore, are related in much stronger ways than by analogy. Any candidate correspondence theory will have to be based on a much richer ontological foundation than is espoused in set theory, for at least the following reason: in virtue of its explicit rejection of invisible modelling, correspondence theory will have to be able, in its own right, to cope directly with the full registrations of domain and co-domain.

For example, suppose someone wanted to use correspondence theory to assess the familiar representation relation between pairs of real numbers and points on a plane. In the model-theoretic tradition, the first job would be to develop models of both phenomena. However, since ordered pairs are an eminently good model both of themselves and of points, the representation relation would look to be one of identity. In order for a correspondence theory to see the relation, it would have to license both ordered pairs of real numbers and points on a plane as legitimate, distinct, entities — as *first class citizens*, to use the computational phrase. Thus a set-theoretic base would simply not work.

Given an adequate ontological foundation, however, and a concomitant account of correspondence, one should be able to repair some well-recognised lacks in current computer theorising, all of the "too coarse-grained" variety. The broad metric of Turing equivalence is a particularly blatant example, since virtually every imagined computer language is of equivalent power. The problem is that the very notion of Turing equivalence rests on promiscuous modelling; in showing one machine equivalent to another, you don't *really* show them to be the same; rather, you show that you can *implement* one in the other. More seriously, all sorts of rather close correspondence relations — implementation, encoding, modelling, etc. — have apparently fallen between the cracks, being "closer", so to speak, than is typical of the representational import of language, but still distinct from identity. The hope is that a proper categorisation of correspondence will be an important first step towards more adequate foundations and more subtle comparisons.

The third semantical consequence has to do with the potential integration and unified treatment of a wide variety of apparently disparate kinds of representation. Ever since the earliest days of AI there have been debates about the relative merits and properties of so-called analogue, pictorial, or imagistic representations, vis. a vis those that are sentential, propositional or, as Sloman calls them [Sloman, 1975], "Fregean". Maps and diagrams are paradigmatic examples of the former; natural language sentences and formulae in first-order logic, of the latter. In spite of a diverse literature probing these distinctions and explicating cross-cutting distinctions buried in them,[24] however, no comprehensive framework has emerged in which to reconstruct the underlying insights. It is difficult not to notice that writers on these topics often refer back to Wittgenstein and Peirce, who wrestled with these issues before the development of modern semantical technique.

This literature conveys an unmistakable picture of complexity inherent even in the most paradigmatic examples. For example, Sloman [1975] attempts to differentiate analogic and Fregean representation by supposing that the former manifests a certain kind of correspondence (he doesn't constrain it) between the *part* structures of representation and represented. On the face of it, this would seem to amount to a structural correspondence between relations, of the sort we saw in discussing iconicity, coupled with a mereological registration of both source and target domains. The pure characterisation, in other words, seems exactly the sort that a correspondence theory should be able to explicate. Sloman's proposal, however, seems much less successful as a way of clearly discriminating between analogue and propositional representation. For example, as many have pointed out (see for example Pylyshyn's discussion [1978]), it doesn't have the intended bite unless one ties down the notion of 'part'. For a bar chart to remain analogue, the conception of part in the target domain must be taken quite liberally; on the other hand, such sentences as "Adrian, Adelia, and Aaron arrived in that order" seem to employ part relations in source (sentence) structure to signify part relations in the target (what is described). So the distinction isn't so clear. Furthermore, there is no doubt that even paradigmatic analogue representations or images represent only with respect to a correspondence relation (see for example Fodor [1975]), so the constraint on mereological correspondence would need to be spelled out, in exactly the way that our algebra of correspondence types suggests.

Without delving into specific examples, several general things seem clear. For one thing, the persistent intuition that representations come in a wide variety of kinds seems exactly right. For another, analysing these kinds will require exactly the sort of fine-grained correspondence theory we are proposing. Finally, it is unlikely that common examples will sort into any small, mutually exclusive, set of nameable classes. Instead, we should license a full range of types of correspondence, kinds of circumstantial dependence, and varieties of registration (continuous, discrete, compositional), in terms of which subsequently to characterise pictures, maps, graphs, schedules, models, images, and so forth, as well as sentences, formulae, and elements of language. The latter group, one would guess, will in general be more *complex* than the former, and may involve additional kinds of circumstantial dependence, compositional structure, or relational complexities such as polarity. But they surely won't be totally distinct.

In section 7 I introduced the phrase "correspondence continuum" to connote the interacting complex of difference correspondence relations we often find connecting representation and represented. However, I equally intended the words to suggest the different kind of continuity arising here: of a full range of variation of type of representational structure.

A simple example will illustrate how continuous these types can be. Modern construction blueprints contain what, to the uninitiated, can be a bewildering range of symbols, ranging from obviously analogue outlines of room shapes, through suggestive icons indicating plumbing and kitchen fixtures, heaters, etc., through slightly stylised icons for electrical outlets, light switches, etc. (with a number of slashes to indicate number of individual outlets, an 'S' to mark whether they are switched, etc.), through general purpose furniture icons with simple inscribed names (desk, bed, etc.), through icons with manufacturer's annotations ("Vermont Castings", "Wolf", etc.), through intermixed sketches, diagrams, and annotations on construction technique, all permeated with arrows, English comments, stamps of approval, scribblings to cancel out parts of the specification, and so on and so forth. That there is a rich variety of representation seems without doubt; that a theoretical scalpel could carve the assemblage into a few neat categories, extraordinarily unlikely.

The moral is unchanged: current representational practice outstrips current semantical technique. Recognising that our current apparatus was developed primarily in service of very particular representational systems employed for logic and mathematics, we should instead embrace what Ken Olson has suggested [Olson]: a return to as various and thick a structure of correspondence relations as Peirce ever imagined. Unlike Peirce, however, we can avail ourselves of the full battery of rigorous mathematical methods, axiomatic systems, and so forth, that have been developed since his time. We might even, with such a project, be able to rescue some of the richness of the "semiotic" tradition from what has been perceived to be its vagueness and descriptive complexity.

The fourth and final consequence listed at the beginning of this section has to do with lurking problems in the traditional approach. Those problems, however, arise from fundamental metaphysical questions, and will as such be addressed in the next section.

10. Theories, Models, and Metaphysics

Figure 13 painted a continuum of relations, starting on the left with the linguistic or representational structure under analysis, and progressing in some fashion towards the "real world" on the right. I have suggested that a correspondence theory would provide us with an ability to characterise the relations among the structures comprising this whole, but I haven't addressed the question of how one would locate oneself in the resulting continuum. If, as I have suggested, the practice of calling certain relations "syntactic" and others "semantic" isn't helpful, is there any other way to distinguish one analysis from another? Or, to put the same question the other way around, can we say anything about traditional approaches? How are they located on this as-yet rather unstructured map?

Four things can be said. First, if this picture is even roughly correct, it predicts that we will encounter structures at various stages across the continuum — relatively more "linguistic" or "syntactic" ones, closer to the primary representational source on the left, others midway across, perhaps having to do with meaning or semantic uniformities, and others relatively more metaphysical or ontological, closer to the full buzzing confusion on the right. That the distinction becomes a matter of degree, rather than a binary decision, makes sense of various traditional debates and disagreements. In particular, it is somewhat of a theoretical relief.

To be specific, many people (I am one) have worried about the metaphysical foundations of particular model-theoretic analyses of language,[25] feeling that the proposed model structures reflect, at least in part, the structure of language, not the structure of the world the language is about. For example, consider an analysis (such as a term model) that posits distinct one, two, and three-place relations for various different uses of the verb 'break' (as in 'the window broke', 'the hockey puck broke the window', and 'I broke the window with a hockey puck'). Or imagine an analysis that distinguishes the Pope's saying Mass from the *fact* of the Pope's saying Mass. Or imagine (not hard!) debates about the metaphysical reality of possible worlds, with some people saying that they are real, others saying that they are merely theoretical devices with which to classify language, others claiming that arguments about the reality of semantical constructs miss the point, which is after all to prove various mathematical facts about the linguistic structures themselves. Or suppose someone were to doubt, on metaphysical grounds, the received wisdom that positive and negative facts were on a par, feeling instead that this symmetry is a device of language, not a fixture in the world.

If one were to adopt the traditional binary view, then all such questions have to be settled one way or the other. I.e., you would have to reject an otherwise appealing semantical analysis if the semantical structures it proposed were metaphysically unconvincing. On the kind of view I am suggesting, however, the whole continuum of possibilities is exactly what one would expect. You could accept a term model semantics, for example, but understand it as living rather close to the left hand side, and then ask for further relations to ground it further to the right. The structure of the continuum, that is, gives you a way of accepting your fellow theorists' intellectual contributions, even while disagreeing with their metaphysical predilections.

Second, there are several ways one might locate a particular correspondent structure in a given semantical analysis. For example, it was pointed out early on that much of the semantical contribution of linguistic use arises from circumstances of utterance, not directly from the structure of the sentence used (as in the "I'm right; you're wrong!" example). One of Barwise and Perry's chief points about language is that this property, which they call *efficiency*, is necessary to the proper functioning of communication. It is natural, then, to imagine an analysis of language use that spelled out this circumstantial dependence. It is also easy to imagine, as a semanticist, wanting to avoid the recalcitrant metaphysical problems that arise when you try to map specific vocabulary items onto the world itself (see below). So the following approach might suggest itself: develop a correspondent structure midway between utterances and the world, in such a way that the entire circumstantial dependence of language, up to questions about the metaphysical foundations of vocabulary, has been discharged. The resulting structure is liable to be infinite, but of course that isn't a theoretical problem.[26]

In particular, this seems a productive way to understand the semantical structures posited both by possible world semantics and situation theory. There are important differences between the two proposals, of course, some of which we can describe: possible world semantics *models* what it calls the interpretation of sentences, whereas situation theory (at least in recent variants) try to deal with interpretation directly. But the point is to reject as too simplistic the question of whether the structures they each propose are to be viewed as *the structure of the world*, albeit highly idealised, or as *the structure of language*, albeit decontextualised. Instead, they can both be understood as intermediate analyses.

Third, it is important to dispel a false assumption about how correspondence relations will go, as we move from left to right. As many writers have noted, far more distinctions are made in the syntax of most formal languages than in the model-theoretic structures posited as their interpretations. The most extreme example is the traditional interpretation of all sentences as denoting one of two values: Truth or Falsity. But the general situation is much more common: different spellings with the same content; different procedures designating the same function; etc. Similarly, logical proof theory, defined in terms of syntax (towards the left) pays attention to far more details than does traditional model theory (though of course proof theory doesn't pay attention to all details, such as to when a formula was written, or to whether parentheses or brackets were used). All of these examples suggest, in general, that correspondence relations will gradually lose information, as they move towards the right, as suggested in

Figure 17. This assumption is for example embedded in approaches that use initial and final algebras as interpretations for programming constructs.

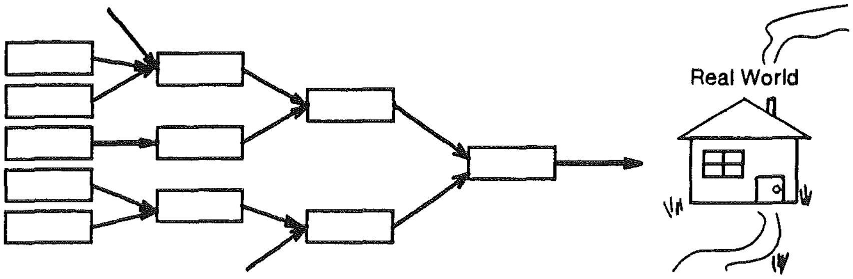

Figure 17: The "losing information" view of semantics

Considerations of circumstantial dependence, however, and some metaphysical arguments, suggest that this neat structure may be an artifact of formal languages, not a general truth of semantics (Barwise, in fact, [1986b, p. 331], defines "formal" languages to be exactly those that are not circumstantially dependent in this way). In the general case, in other words, semantics should not be viewed as a way of moving from fine- to coarse-grained linguistic distinctions. This stance is clearly false if circumstance is ignored: different uses of the word 'I', as we have pointed out so often, can refer to indefinitely many different people, as can 'now' refer to arbitrarily many different times. But more complex phenomena suggest other structures. For example, imagine an analysis of natural language, along the lines suggested above, that ignores different people's sense of the reference of some term — 'guilt', say, or 'like' — about which inter-personal agreement is rare. If there is a fact of the matter, when a given person says "she likes feeling guilty", as to what aspect or property of the world is thereby named, then it follows that the *real* connection from utterance to world will discriminate more finely than our chosen semantical analysis.

I choose this example because I can imagine that it would be a serious mistake to try, in the analysis of language, to compensate for such differences by writing them in terms of an explicit parameter for something like "speaker's conceptual scheme" — what I will call *registration scheme*, to connect it to our previous conception of a "pre-registered" correspondent domain. I.e., for certain purposes we may not *want* to capture all the richness of the representation, nor all the richness of the world, nor all the richness of the connection between the two. But from this fact one cannot conclude that richness recedes as one moves to the right.

Fourth and finally, there remains the very serious metaphysical question of how any analysis at all is going to deal with the right hand end: the world itself. In fact our continuum seems to suggest that one of the great appeals of the model-theoretic semantical approach — for natural language, AI, and other systems — is that it stops the analysis half-way across the continuum. As suggested above, there are those who worry that the resulting models are still infected with the structure of the languages they purport to analyse, but this has its advantages. Theorists who disagree wildly on the actual structure of the world itself, if that even means anything coherent, can nonetheless agree on a model-theoretic structure. More specifically, one would expect proportionally more agreement — among realists, skeptics, idealists, and theorists of every conceivable metaphysical stripe — to the extent that one's semantic analysis established a correspondence to a structure further towards the left. In fact any two people who agreed on an analysis *all the way towards the right* would by definition be of exactly the same metaphysical persuasion; that's what such agreement would mean.

The strongest claim I will make about metaphysical grounding will arise in the next and final section, when I return to the semantics of knowledge representation, but a preliminary point can be made here. It has to do with semantics as an instance of theoretical inquiry. To start with, make the following two relatively noncontroversial assumptions. First, assume that we human theorists, when we use language, are somehow able to refer to the world itself, even if we don't yet know how. I.e., assume something like the most modest form of realism possible: just that there is a world, that we're in it, and that our words somehow enable us to get at it. This is all perfectly compatible with everyone's carving it up in radically different ways, as dictated by nature, nurture, or just plain whim. Second, assume that theories are linguistic vehicles with which we communicate our understanding to our fellow person. Or assume that theories are linguistic entities claimed to be true; it doesn't matter.

Once granted these two assumptions, the following is an immediate conclusion: to the extent that our theories are legitimate instances of language, we are able to refer to the world. It follows that, as theorists, we don't lack ways of getting to the right hand end of the diagram. I, for example, can get there right this minute with the phrase "this lukewarm cup of coffee to my right". The problem, of course, is that I don't necessarily know various things: not only how it is that I manage to refer to the cup, but also the way in which I have thereby referred to it. So the metaphysical problem for semantical theorists is not one of *referring* to the world by using theoretical language, but rather something closer to the opposite: there is no way of referring to the world *except* by using language. Neurath's boat once again.

This much is obvious. What's important about it is that it is true *all the way across the continuum*: we have no way to refer to the representational structure on the left, or to any intermediating correspondent structure, outside of language either. It only feels more problematic towards the right because it is there that we encounter a natural tendency to want to escape our own particular conceptual schemes, especially if we and the representational structure in question part company. What he calls 'duty' she calls 'guilt'.

This may indeed may be a real limitation: the chances of explaining, all the way to the right, the semantical interpretation of a system whose conceptual scheme differs radically from one's own, is probably essentially nil. Radical indeterminacy of translation, if there is such a thing, surely has what we might call radical indeterminacy of semantics as a sub-species. But there are more interesting conclusions, as suggested in Figure 18.

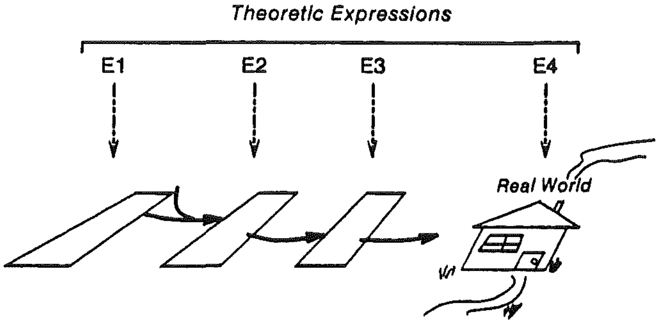

Figure 18: Semantics of theories of correspondence

To the extent that theorist's language and representation overlap on registration scheme, the problems are clearly that much less. This is the happier case, of course, but it has this curious consequence: as analysis moves towards the right, it will *look*, to an outside observer, as if the representation in question is gradually being translated into the theorist's own language. I.e., we might say that the noun 'chat' (towards the left) is modelled by the objectified CAT relation (middle), which in turn characterises the set of real cats (right). I.e., quotation on the left, reification or nominalisation in the middle, ordinary use on the right. But this is just as it should be; it is predicted by the diagram.

There is absolutely no reason to conclude, from this observation, that semantics inherently involves translation.

On the other hand, to the extent that theorist's registration scheme is his own, it will be so all the way across the diagram. Just because the theorist registers the representational structure itself in terms of a given set of properties and relations (say, as having a particular syntactic form), there is no reason to believe that the representational system registers itself in this way — if indeed there is any reason to suppose that it registers itself at all. I.e., if, as I am inclined to suppose, registration involves representation (as well as *vice versa*), then the subject system will register *only* what is to the right; the rest is registered *only for theoretical purposes*.[27] As before, conflict can occur only at the right hand end, but only because that is the only thing that *both* system and theorist register.

In sum, the idea that semantics involves translation is a superficial rendering of the much deeper though perfectly straightforward fact that semantical analysis, like all theoretical investigation, is carried on in language, left through middle through right.

11. Knowledge Representation Revisited

Although we may seem to have strayed a fair distance from knowledge representation, its demands have been our constant motivation. First, we've seen that the semantical competition between 'representation' and 'knowledge' was merely the tip of a rather large iceberg: without even trying to enumerate an exhaustive list, half a dozen other intentional notions were added to the semantical roster. Second, with respect to appropriate semantical technique, I argued for the prior development of a comprehensive theory of correspondence, and sketched some preparatory philosophical foundations. One way to view this proposed theory is as a branch of mathematics that would immeasurably aid semantics in two ways: by clarifying the semantical project itself, and by providing conceptual vocabulary in terms of which to classify genuinely semantic relations.

On the other hand, I have tried to say plainly that a theory of correspondence would not itself be a theory of semantics, or representation, or knowledge; in fact, in spite of all the ground we've covered, I have said virtually nothing here about the essence of any such notions. Even section 9, which tries to sketch some of the structure in which semantics would proceed, still does nothing to resolve this piece of homework. Nor can I do more here. My only intent, by way of a last conclusion, is to make one brief foray in this direction, which will tie the whole analysis back to the primary distinction made at the outset, between representational import and functional role.

The point is simple. I said that functional role and representational import must be coördinated: the agent must be able to act sensibly in terms of what it represents, and (perhaps) represent what it can act sensibly towards. This coördination can be viewed as a kind of "coming together" of knowledge (second factor) and action (first factor). Thus, suppose, knowing the paper is almost over, I reject the lukewarm coffee on my right in favour of good brandy in the cupboard. I'd like the impression that represents the brandy to engender the action of my crossing the room, pouring out a glass, and raising it to my lips. What is of paramount importance, for our purposes, is the following fact: in the terms of the continuum diagram, this coming together of representation import and action (which is one kind of functional role) must be *all the way to the right.* I want to drink *what's in the world*, not a model or indirect classification of a cup of brandy, nor a term model of 'brandy' expressions, nor a set-theoretic assemblage of sentences or impressions containing representations of the property of being brandy. Whatever "stuff itself" is, that's what my actions must be directed towards.

This observation, merely a theoretical consequence of the dual facts that action takes place in the world, and that functional role is a kind of action, is the grounds for our sixth and final challenge to the model-theoretic tradition, promised earlier. Because computer systems participate with us in the world — stop our cars, launch our weapons, deliver our mail — it is imperative that our analyses of the representational import of impressions take us all the way to the real world situations towards which the engendered action will be directed. Tooth decay among children won't be reduced by a computer's injecting a mathematical model of fluorine into a set of possible worlds. In order to see the coördination between functional role and representational import, that is, both parts of our two-factor analysis of significance must reach all the way to the right. Let's call an analysis that reaches out that far a *grounded* account.

So far, then, the only coördination requirement I will put on theories of full significance is that they be grounded. At least for the moment, that will have to be requirement enough.

Acknowledgements

A slightly earlier version of this paper appeared with, but not in (because it was late), the Proceedings of the Sixth Canadian Conference on Artificial Intelligence, held in Montreal, Quebec, Canada on May 21–23, 1986; the current version differs only by minor typographic edits and clarifications. Thanks to Jon Barwise, Jim des Rivières, and John Lamping for comments on an earlier draft. The research was supported by Xerox Corporation, and by a System Development Foundation award to the Center for the Study of Language and Information.

Notes

1. Saying just what distinguishes representational from purely functional ingredients is a difficult philosophical problem. My own emphasis on the two criteria cited here — a certain "disconnection" between representation and what is represented, and the claim that a representation must represent the world *as being a certain way* — is discussed in [Smith, forthcoming (a)], chapter 4, and in [Smith, forthcoming (b)]. The issue has been addressed by many writers in the philosophy of psychology, such as Fodor, Searle, and Stich, especially in assessing the relation between proposed functional and representational theories of mind. Computational readers will note, however, that many of these philosophers get at representation by analogy to computation, whereas my own view is approximately the opposite: that we must get at computation by first understanding representation. There is more overlap in subject matter than concurrence in views.

2. David Israel has challenged the view, almost universally held in AI, that the notions of *proof*, *deduction*, *inference*, etc., even in mathematical logic, should be conceived in syntactic terms. This syntactic orientation is not even universally accepted within what is called formal logic, since it rests on only one of many possible readings of the term 'formal' (see [Smith, forthcoming (a)]).

3. Reasons why the functional (procedural) parts count as semantic are spelled out in [Smith, forthcoming (a)].

4. First factor derivability ($\vdash$) and second factor satisfaction are traditionally tied together through entailment ($\models$) and proofs of soundness and completeness, but these particular notions are coherent only as a kind of global constraint on what are otherwise locally independent factors. The kind of "intimate conjunction" employed in 3-lisp, and being imagined here for more general models of reasoning and computation, is one of much more local interdependence. As pointed out in [Smith, 1982b], computational practice already encompasses a wide range of such local interactions; see also [Smith, 1987].

5. Functional parameterisation deals with circumstantial dependence, but in a specific and limited way. In particular, by assuming that the linguistic element, plus circumstantial facts, together determine the interpretation, it implies that this is the direction of "information flow" — that understanding proceeds from knowledge of language, plus knowledge of circumstance, to knowledge of content. In practice, however, the flow can easily run in the other direction: someone hearing an utterance may know about the situation being described, and use that information to determine the

structure of the linguistic element, or of such circumstantial factors as discourse structure. For these and other reasons a genuinely relational theory of meaning and content would be preferable [Barwise and Perry, 1983]; I use the functional analysis here only because of its familiarity, and because my current argument isn't particularly sensitive to the distinction.

6. Sometimes, as for example in Montague semantics, the syntactic domain is modelled as well, but I won't worry about that here — it is merely an extension of the same points we are making.

7. I don't, at least in this paper, intend these remarks to challenge the appropriateness of these techniques for the intellectual project for which they were developed: the metamathematical inquiry into the foundations of mathematics. My complaint here is only about its adequacy for use in AI, knowledge representation, and any other situation in which the true state of affairs being represented is one in the real and messy world of everyday life.

8. Richard Boyd [1979] argues persuasively that metaphorical scientific language can play a role, especially initially, in enabling a community to establish increasingly substantial reference to a new domain. On such an account, the use of linguistic terminology to discuss impressions might, over the years, gradually lose its metaphorical overtones, and take on full-fledged referential connection to this new domain. But, as Boyd himself points out, in order for this process to take hold, the metaphor must start out being at least partially correct. My concern in this particular case, as the rest of this section tries to suggest, is that many of the connotations of the use of linguistic language to describe impressions are in fact unwarranted.

9. History is often repeated, we are told, but here it is being repeated in reverse direction. The gradual shift from functionalism to representationalism in the philosophy of mind is apparently being played out backwards in AI, which started with a very strong representationalist stance, and is steadily moving away from it, towards what are explicitly admitted to be purely functional accounts.[Levesque, 1984; Newell, 1982] My own view is that both traditions, in opposite order, suffer from the lack of a full fledged theory of representation. Based on the idea that the only rigorous concept of representation is a narrow, purely syntactic, version, they oscillate between its gratuitous detail and consequent semantic implausibility, on the one hand, and contextually insensitive and menacingly behaviourist pure functionalism, on the other. I believe both are inadequate, and conclude that we should free representation from its syntactic strictures, rather than rejecting the notion entirely.

10. Although I will eventually challenge the idea of a rigid use/mention distinction, that doesn't mean that many so-called "use-mention confusions", such as this, aren't serious.

11. Some readers will object that computer science analyses treat computational processes only in terms of surface behaviour — input/output relations — without positing any internal structure at all, let alone impressions. But this isn't so clear, not only because I have defined impression in a rather general way, but also because this view assumes a purely "extensional" reading of the semantical analyses themselves. As has been argued by Fodor and others in the mental case, some sort of representational ingredients will often be posited by theory merely in order to state the proper behavioural regularities. The abstract data types of denotational analysis can be viewed purely as theoretic entities, without classificatory import, but an argument would have to be made that they don't represent impression structure; the mere fact that they aren't *claimed* to do so isn't sufficient.

12. Folk psychology faces exactly the same problem we have just surveyed. In particular: (i) it classifies people's mental states by content; (ii) the purpose of these classifications is to explain how people behave and what they do; and (iii) the content of people's mental states is determined in part by their circumstances. These facts have led some writers, such as Stich [1985] to conclude that folk psychology will never be scientifically reconstructable, but this seems to be an unwarranted pessimism; the problem, rather, is to see how folk psychology compensates for the external circumstantial dependence.

13. As usual, and as the example about Lisp code suggests, practice is in fact one level more complex than this analysis suggests: one gives the operational semantics of a programming language L, viewed specificationally, by translating expressions types of L into complex expressions types of programs, written in an implementing language L', that implement L. The language–process relation for L' is what is usually assumed.

14. There was some misunderstanding, when 3-lisp was introduced, [Smith 1982, 1984] about the two semantical factors in terms of which it was analysed and designed (Ψ and Φ, they were called, but they corresponded directly to first and second factors in the framework being presented here). Unfamiliar with the two-factor framework, many computer scientists assumed they were merely new names for operational and denotational accounts, respectively. This was false, of course, but in retrospect the confusion can be attributed to three things: (i) the fact that 3-lisp was designed on an "ingredient" view of programs, whereas, as described in the text, programming language analysis is typically carried on within the

specificational tradition; (ii) 3-lisp's represented "world" was constrained to being one of pure mathematical abstractions and internal structures (since it was presented as a computational model of introspection), so that the *domain* that 3-lisp impressions represented was the same one that would normally be used for both operational and denotational semantics — i.e., the domain of impressions and of the obvious mathematical models of them; and (iii) because of this restricted domain, the interpretations of 3-lisp impressions weren't dependent on external circumstances, so that the clear difference between model and interpretation, noted at the end of section 5, didn't apply.

These three reasons conspired together; it has only been in the last few years that the various intricacies of their relationship have come clear.

15. Which I doubt, for reasons that can easily be explained using terminology we've already introduced. As classically understood, standard first order logic is both declarative and syntactic, in the sense of section 2. Real-life Prolog programs, however, violate the assumed independence of factors: their role affects their import. Lacking techniques for spelling this out (i.e., techniques for providing explicit two-factor analyses), most computer scientists who give semantics for Prolog programs in fact provide model-theoretic analyses of functional role, using term models and such, in the sense explained in section 5. Logicians, expecting analyses of representational import, quite reasonably find these reconstructions odd. Furthermore, to the extent that it is functional role, not representational import, that is retained, Prolog's claim to clear *semantics* is thereby undermined.

    Note that a model-theoretic analysis of functional role (first factor), on the ingredient view of programs, is liable at least partially to coincide with a mathematical model of representational content (second factor) of the programs used (on the specificational model) to describe them. The subject matter is rife with such potential semantical confusions.

16. In point of fact only one factor of the full significance was an isomorphism.

17. This implies, of course, that there must be much more to representation than correspondence. Hence footnote 1; correspondence on its own requires neither disconnection nor registration.

18. Strictly speaking I don't believe this, for two reasons. First, my metaphysical predilection is to attribute the notions of object, property, and relation to a collaborative interaction between mind and world, so that the world alone needn't be held responsible for objects' boundaries and kinds (naive realism), nor need they be viewed as pure constructs of cognition (variants of solipsism or idealism). Second, I am at least prepared seriously to entertain the hypothesis that minds, fundamentally, are embodied

representational processes. In conjunction these two views raise the following "chicken and egg" problem: if minds are required in order to know how the world is structured, and if minds are representational, then representation must seemingly be studied before correspondence, in order to establish the categories in terms of which the correspondences will be articulated. On the other hand, for reasons spelled out in the text, I think the chances of getting representation right without a prior theory of correspondence are rather limited.

These considerations interact with another distinction. Which person is being held responsible for the categorisation of the domains in question: the agent under study, or the theorist? I assess the interaction among these issues in [Smith, forthcoming (b)]; the net result is simply the rather predictable conclusion that the two notions (correspondence and representation) must be viewed as something of an indissoluble pair. This conclusion, however, doesn't in any way challenge the view being expressed here: that they aren't the *same* notion.

19. Note that this phrasing suggests iterated correspondence: expressions to function applications, and from there to values. The connection between iterated correspondence and so-called 'intensional' analyses of functions and relations is discussed at the end of this section.

20. My intention is to employ the term in a way compatible with its technical use in Situation Theory [Barwise, 1986a], although nothing in the text requires that particular analysis.

21. The theoretical stance of taking registration as prior to correspondence, and correspondence as at least partially independent from representation, is not one I am ultimately satisfied with; see footnote 18, and [Smith, forthcoming (b)]. It seems well motivated, though, at least as a way of getting to the next stage in semantical clarity.

22. In cases where a third agent — an interpreter — is present, a possible solution is presented to the problems raised in footnotes 18 and 21: the agent can register both representation and represented. But there are two problems with this. First, of course, we have to ask how agents register, which brings the problem back to roost. Second, it is a strong and possibly false claim that interpreters register signs and language they use (as opposed to mention).

23. Introspection suggests that quotation marks are primarily, if not always, used to refer to linguistic types. As a possible counter-example, Geoff Nunberg has suggested " 'Fiat lux' started this whole mess' ", but at best that refers to an utterance of the Latin sentence different from the (enclosed)

one used to refer to it. It does seem that quoted expressions cannot be used to refer to their constituting internal tokens.

24. A representative series of articles by Dennett, Fodor, Kosslyn & Pomerantz, Pylyshyn, and Rey can be found in part two ("Imagery") of Block's [1981]. See also Sloman [Pylyshyn, 1984; Sloman, 1975] and Pylyshyn [1984] chapters 7 & 8.

25. The difficulties are blatant in term models, evident in Kripke style possible world structures, but still apparent, at least to my mind, in the structure of the situation-theoretic universe [Barwise, 1986a].

26. John Etchemendy once suggested that the situation-theoretic universe could be viewed in this way — as the world's only non-situated language.

27. The theorist, of course, can either be us, or else the system introspecting on itself; see [Smith, 1986].

References

Barwise, Jon [1986a], "Situations, Sets, and the Axiom of Foundation", Alex Wilkie ed., *Logic Colloquium 84* Amsterdam: North Holland.

——— [1986b], "Information and Circumstance", *Notre Dame Journal of Formal Logic*, Volume 27 Number 3, July 1986.

Barwise, Jon, and Perry, John [1983], *Situations and Attitudes*, Bradford Books, Cambridge, MA.

Block, Ned, ed., [1981], *Readings in Philosophy of Psychology, Vol 2*, Harvard University Press, Cambridge, MA.

Block, Ned [1985], "Advertisement for a Semantics for Psychology", *Midwest Studies in Philosopohy X*, P. A. French, T. E. Uehling and H. K. Wettstein, eds.

Boyd, Richard [1979], "Metaphor and Theory Change: What is 'Metaphor' a Metaphor For?", in A. Ortony, ed., *Metaphor and Thought*, Cambridge University Press, Cambridge, MA, pp. 356–419.

Brachman, Ronald J., and Levesque, Hector J., eds., [1985], *Readings in Knowledge Representation*, Morgan Kaufmann, Los Altos, CA, 571 pp.

Field, Hartry [1977], "Logic, Meaning, and Conceptual Role", *Journal of Philosophy* Vol. 74.

——— [1978] , "Mental Representation", *Erkentniss*, Vol. 13.

Fodor, Jerry [1975], *The Language of Thought*, Thomas Y. Crowell Co.: New York. Paperback version, Harvard University Press (1979), Cambridge, MA.

Goguen, Joseph A. and Meseguer, Jose [1984], "Equality, Types, Modules and Generics for Logic Programming", CSLI Technical Report CSLI–84–5 Stanford University, Stanford, CA.

Goodman, Nelson [1983], *Fact, Fiction and Forecast*, Harvard University Press: Cambridge, MA.

Gordon, Michael [1979], *The Denotational Description of Programming Languages: An Introduction*, Springer-Verlag: New York.

Hayes, Patrick J. [1974], "Some Problems and Non-Problems in Representation Theory", *Proc. AISB Summer Conference*, University of Sussex, pp. 63–79. Reprinted in [Brachman and Levesque, 1985], pp. 3–22.

——— [1977] "In Defence of Logic", *Proc. IJCAI–77*, Cambridge, MA pp. 559–565.

Levesque, Hector [1984], "Foundations of a Functional Approach to Knowledge Representation", *Artificial Intelligence*, Vol. 23, pp. 155–212.

Lewis, David [1972], "General Semantics", in D. Davison and G. Harman, eds., *Semantics of Natural Language*, D. Reidel, Dordrecht, Holland, pp 169–218.

Loar, Brian [1982], "Conceptual Role and Truth Conditions, *Notre Dame Journal of Formal Logic*, Vol. 23(3).

Moore, Robert C., and Hendrix, Gary G. [1979], "Computational Models of Belief and the Semantics of Belief Sentences", SRI International Technical Note 187 Menlo Park, CA.

Moschovakis, Yannis [1984], in *Lecture Notes in Mathematics* vol 1103 on Model Theory, Heidelberg: Springer-Verlag.

Newell, Allen [1982], "The Knowledge Level", *Artificial Intelligence* Vol 18 (1), pp. 87–127.

Olson, Kenneth [1985], personal communication at CSLI.

Pylyshyn, Zenon [1978], "Imagery and Artificial Intelligence", in C. W. Savage, ed., *Perception and Cognition: Issues in the Foundations of Psychology, Minnesota Studies in the Philosophy of Science*, vol. 9. University of Minnesota Press, Minneapolis, pp. 19–55. Reprinted in [Block, 1981], pp. 170–194.

——— [1984], *Computation and Cognition: Toward a Foundation for Cognitive Science*, The MIT Press/a Bradford Book, Cambridge, MA.

Rosenschein, Stanley J. [1985], "Formal Theories of Knowledge in AI and Robotics", SRI International Technical Note 362 Menlo Park, CA.

Sloman, Aaron [1971], "Interactions Between Philosophy and Artificial Intelligence: The Role of Intuition and Non-Logical Reasoning in Intelligence", *Artificial Intelligence* vol 2 pp. 209–225.

——— [1975], "Afterthoughts on Analogical Representation", *Proc. Theoretical Issues in Natural Language Processing*, Cambridge, MA pp. 164-168. Reprinted in [Brachman and Levesque, 1985] pp. 431–439.

Smith, Brian C. [1982a], *Reflection and Semantics in a Procedural Language*, Technical Report MIT/LCS/TR-272, M.I.T., Cambridge, MA, 495 pp. See also [Smith, 1985].

——— [1982b] "Linguistic and Computational Semantics", *Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics*, Toronto, Ontario, June 1982.

——— [1984], "Reflection and Semantics in Lisp", Conference Record of 11th POPL pp. 23–35, Salt Lake City, Utah. Also available as Xerox PARC Intelligent Systems Laboratory Technical Report ISL–5, Palo Alto, California, 1984.

——— [1985], "Prologue to *Reflection and Semantics in a Procedural Language*", in Brachman and Levesque [1985], pp. 31–39.

——— [1986], "Varieties of Self-Reference", in *Theoretical Aspects of Reasoning about Knowledge: Proceedings of the 1986 Conference*, Morgan Kaufmann, Los Altos, CA. Also available as CSLI-87-7?, Stanford University, Stanford, CA 94305. Revised version to appear in *Artificial Intelligence*. ~~72~~ 73

~~73~~ 74

——— [1987], "The Semantics of Clocks", *Synthese*, forthcoming.

——— [forthcoming (a)], *Is Computation Formal?*, MIT Press/A Bradford Book, Cambridge, MA [1987].

——— [forthcoming (b)], "Representation and Registration".

——— [forthcoming (c)], "Categories of Correspondence".

Stich, Steven [1985], *From Folk Psychology to Cognitive Science: The Case Against Belief*, MIT Press/A Bradford Book, Cambridge, MA.

Woods, William A. [1981], "Procedural Semantics as a Theory of Meaning", in A. Joshi, B. Webber, and I. Sag (eds.), *Elements of Discourse Understanding*, Cambridge University Press, Cambridge.