

Annotations[†]

- A1** :1/1/5:8 The term ‘independently’ is too strong, almost regardless of how it is interpreted. Substantially, 2/3Lisp’s procedural and declarative aspects are intimately related, as suggested by the subsequent “in order that they may be coherently related,” and as extensively discussed in [ch. 2](#).^{0.5} It would be more reasonable to interpret the sentence as implying that declarative and procedural aspects are conceptually or logically independent, though today even that is something I would deny. Not only are 2/3Lisp’s declarative and procedural semantics logically interdependent, but so too, I would now argue, are formal logic’s syntax and semantics.
- Are the concepts of length and area logically independent? Or of first-person and second-person points of view?^{0.7}
- A2** :2/1:ff The extended abstract is here labeled “§2” purely for purposes of this volume. In the dissertation itself it figured as an extended abstract, which is why it overlaps so much with (what is here called) §1.
- A3** :3/1/3:4 It is unclear whether the terms ‘interpreted’ and ‘interpretation’ in this sentence were intended to signify declarative important (what expressions in the calculus denote or represent), procedural consequence (how expressions are processed), or both—what I call full significance. I would like to think that I meant the last, though 3Lisp’s facilities for reflectively “affecting” declarative import are minimal, if they can even be said to exist at all.^{0.8}
- A4** :4/0/6 The fact that normal-formedness was in part (even if barely) semantically defined, rather than be characterized purely “syntactically,”

[†]References are in the form page/paragraph/line; with ranges (of any type) indicated as x:y. For details see the explanation on p.

0.5. In fact I believe that one of the dialects’ distinctive contributions is in showing how to preserve an overall deference of processing to declarative semantics, while at the same time recognizing how, in detail, declarative import can be affected by procedural consequence or behaviour.

0.7. See the discussion of the meaning of “independent of semantics,” taken to be a constitutive component of the notion of formality, in [AQS Volume II](#), for an extended analysis of the meaning of the term ‘independent.’

0.8. Fundamentally, the only way to affect the declarative import (φ) of an expression α in 3Lisp (without structurally modifying it) was to wrap it in a context γ in which procedural considerations would block α ’s default declarative import. The result could lead to a different declarative import for γ ; but at best it could cause α not to have any overall declarative import at all.

was important to me, in an inarticulate but nevertheless resolute effort not to submit to the mechanical restriction.

- A5** .4/0/-5 Whereas 2/3Lisp’s semantical framework was largely ignored, its endorsement of category alignment, turned out, in my informal interactions with programmers and computer scientists, to be much more imaginatively vivid—and largely disparaged. See the discussion in ch. 2, §...

The relation of category alignment to ontological issues of fan-out (dealing with “the one and the many”) gripped me at the time 2/3Lisp was designed, and to an extent still do. The issue is not just the one mentioned in the text, of being able to move seamlessly back and forth between treating a set of arguments plurally or severally, and treating them as a singular unity. More seriously, it will come to the fore in the design of the fan calculus, when dealing with the issue of preserving categorization across a designation relation (for example: taking the number two and the numeral ‘2’ both to be “numeric,” in such a way that even if the denotation relation between them is “closed”—not seen, in the sense that the differences between them are elided in such a way that numeral and number^{0.8} are taken to be the same thing [cf. «...»].—the resulting amalgamation could still be called numeric.

More relevant at the time 2/3Lisp was designed, however, was the fact alluded to in the text. The inability of standard Lisps to deal elegantly with group fan-in and fan-out leads the programmer to have to invoke meta-structural and proto-reflective machinery: quotation marks, explicit calls to the function APPLY, etc. Not only does this practice betray deep underlying conceptual awkwardness; it also distracts from more serious reasons to invoke reflection. So category alignment was embraced in 2/3Lisp partly in order to “clear out” the reflective realm in preparation for its more substantial analysis and reconstruction. (See also «ch. 2, ...».)

- A6** .5/-1:6/1 A brief discussion of Mantiq (including a note on the provenance of the name) is provided in §1 of ch. 2.

It is not now clear to me (2014), however, from notes I made about Mantiq at the time, that the description in the text is entirely accurate. In particular, the phrase “in which what happened took effect in virtue of declarative descriptions of what was to happen”

0.8. And perhaps also a particular numeral type and its tokens.

(5/-1/-2:6/0/1) conveys the untenable suggestion that all Mantiq structures would have only declarative force. As is clear from philosophy of language—to say nothing of the Carroll paradoxes—statement alone is not by itself sufficient to engender action.^{1.5} In addition, as in 3Lisp, the aim for Mantiq was to have events, structures, and phenomena co-exist on an equivalent ontological plane (though at a different semantic level!) with descriptions of those self-same events, structures, and phenomena—i.e., without either having ontic priority (a bit of a conceit; but cf. ch. 2).

It is true, though, that the emphasis in Mantiq was to be on description—something for which I still believe support (and theory) to be woefully missing in computational languages. For numerous reasons (cf. «where?») I do not believe such contemporary formalisms as RDF, XML, OWL, etc., come close to filling the bill.

A7 .6/1/-6:-3 This project of developing an architecture in which structural identity¹ could serve as a proxy for intensional identity was one of Mantiq’s primary design aims. It was also something on which I had spent a lot of time working, before 3Lisp was designed. The basic idea was to define a rather abstract conception of a structural field (rather like an abstract memory), implemented by a background running concurrent relaxation algorithms, so that structural identity (of the sort that would be tested by an analogue of Lisp’s EQ) would mimic identity of meaning on a plausible if necessarily relatively fine-grained way. For example: internal analogues of such expressions as $(\lambda y . y+3)$ and $(\lambda x . 3+x)$ were intended in Mantiq to be both token and type identical.

There are legions of issues lurking behind this suggestion—including, for example, concerns of the sort articulated in “The Correspondence Continuum” (ch. 10), that different granularities of intensional identity are appropriate in different circumstances. Part of what I had explored, in the emerging Mantiq design, was the exploitation of reflection to obtain, depending on circumstance, more or less fine-grained access to these sorts of structural granularity, in line with the overall philosophy of providing contextually sensitive

1. The text says ‘syntactic’ identity—but it is very clear that I meant the directly-inspectible identity of internal computational structures (elements of what I called the ‘structural field.’)

1.5. Unless, of course, utterance is itself taken to be an action. «...talk about putting process first, etc., as in ch. 2»

ways of making things more or less explicit (so that operator order, for example, or the forms of equivocation addressed in de Morgan's laws, could be “seen” or “not seen,” depending on purpose and perspective, in a flexible way).

- A8** .7/0/1 In the end I did define a reflective version of the λ -calculus, in an (unsuccessful) attempt to communicate to Jon Barwise why reflection was interesting, and how it worked. See §6 of the Introduction and annotation A41 in ch. 4.
- A9** .7/0/-5:-1 As indicated here, the aim of developing 3Lisp was to work out a semantical framework that integrated an understanding of reference and description into an account of computational activity. It was because of this motivating purpose that I felt that the “delivery” of 3Lisp failed, since even though the notion of reflection was positively received, the semantical framework on which it was based was ignored. Cf. the discussion in the Introduction (especially at in §1, and in §6), and throughout ch. 2, where among other things I suggest that this failure stemmed from untenable ontological as well as semantical presuppositions underlying the understanding of reference employed in 3Lisp's design, which (as I knew well, but whose epistemic significance I failed to appreciate) diverged from those that still remain our default theoretical approach to these subjects both in logic and in computer science.
- A10** .7/-1/-9:-7 This comment makes it especially ironic that the effort in formulation proved to be largely unsuccessful. Cf. ch. 2, §....
- A11** .7/-1/-3:-2 It is the claim that writing programs requires that one make one's ideas completely explicit with which I was disagreeing. I certainly believed (and still do today) that constructing a program requires a kind of explicitization that is extraordinarily demanding—far more so than those who have not programmed are ever likely to realize.
- A12** .11/1/-8:-6 Though endorsing the formality condition here, my belief in its truth (of real-world computation) had already begun to erode. By the time that “Reflection and Semantics in Lisp” was published in 1983 (ch. 4 of the present volume), I was close to being willing to deny its truth (though even today I believe that it is based on a profoundly deep insight). See annotation A15 in ch. 4, and Volume II of AOS.
- A12.5** .12/0/4 The statement “a propositional account of the knowledge that the overall process exhibits” was semantically malformed—containing as

it does three iterated intentional stages: “represent a propositional account of the knowledge.” It would have been somewhat better to say that the process would contain ingredients that “propositionally represented” the knowledge exhibited by the overall system—which would have at least reduced the number of stages to two. It is not knowledge that the ingredients are presumed to represent, however, but rather entities and phenomena in the world (e.g., the sun’s setting) that the system can thereby be said to know about. Or so at least reason would suggest—and as the following paragraph implies.

See the discussion at :22/0, and the opening pages of ch. 12, at «.../...».

- A13** :14/0/8 ‘Skeptical’ is the operative word. At the time I was neither prepared to endorse or to deny the representational view, in spite of its ubiquitous allegiance in Artificial Intelligence at the time (Haugeland’s GOFAI did not come in for resounding critique until later in the decade). See “Registration and Registration,” *Indiscrete Affairs*, Vol. II.
- A14** :15/1/8:9 For discussion of the notion of a dual calculus see :1/-1/-5:-2 and :21/0:1 in ch. 3b.
- A15** :16/1/-7:-6 The phrase “bring into...view of my mind’s eye” is intended transparently or extensionally—i.e., to mean bringing “the self that tomorrow I will be” into the realm that my reflective thinking is about. By analogy, when we say that a photographer rotates their camera in order to “bring an apple tree into view,” it is the tree itself that is brought into view, not an image of the tree. To think about myself, therefore, I need to “push myself away,” as it were (cf. o3), separating myself from myself, so that I can be the object—the declarative import—of my reflective thinking.
- A16** :17/0/-2:-1 As is clear from the adjoining parenthetical (“not self-referentially”), the phrase ‘reflecting about,’ in this sentence, was intended in a very general sense, along the lines of a sentence such as “we were reflecting on the situation in the Crimea”—not in the technical sense of self-directed deliberation under investigation here. It would have been better to have chosen another word.
- A17** :18/1/-6:-1 I was less clear on these issues here than I should have been. Cf. the

2. The claim that “it is almost a great joke that the blindly formal ingredient process should be called an interpreter” betrays the fact—discussed in §5... and ... of ch. 2—that when this document was written I did not fully understand the different ways in which the term ‘program’ is used in computer sci-

discussions of the relation between ingrediential and specificational views of programs and about diverging uses of the term ‘interpreter’ in ch. 2,² as well as ch. 4 and 100 Billion Lines. In addition, there are issues about the relation between personal and sub-personal levels (cf. annotation A26 of ch. 3b, p. 115) which I later came to recognize as profoundly important, but about which in 1981 I was not appropriately aware.

A18 18/-1/-2: In 1980, the year before the dissertation was written, I had presented
19/0 a response to Charles Taylor’s paper “Cognitive Psychology.”³ One of Taylor’s points was that human life is drenched in a background of inexorable implicitness, which he took to be antithetical to (what he took to be) the intrinsic explicitness of computation. While taking no exception to his account of human life, I disagreed with his claim that computation is anything like as explicit as he suggested. Ever since I had started programming in the late 1960s, I had had (and still have) a deep sense that there was far more that is tacit and implicit in the computational realm than is commonly recognized—especially in adjacent fields.

A19 19/0/3 One can say that France is near Marseilles; whether one can mean it is debatable, but it may be possible as well. The sentence should have omitted the epistemic attitude, and been written: “no more reason to suppose it is even a coherent possibility than for France to be near Marseilles.”

A20 20//1:2 The PDP-10, a mainframe computer built by the Digital Equipment Corporation (DEC), formed the backbone of the computational resources then available at the artificial intelligence laboratories at MIT, Stanford, and Carnegie-Mellon. Among other things, it was the (model of) computer on which time-sharing was initially developed; it was also the machine on which Lisp ran, in which all early AI projects were implemented. As a result, it loomed large in the imagination of early AI theorists.

A21 20/2 (Pt. 3): the discussion «...where>...» of the personal/sub-personal distinction...

...
 ence and in philosophy. Yet the passage at 22/1/7:-1 shows that I was more aware that this sentence admits.

3. The response, entitled “The Significance of Computational Psychology,” was presented on March 25, 1980 at a Conference on Artificial Intelligence and Philosophy, held at the Center for Advanced Studies in the Behavioral Sciences in Stanford, California.

- A22** :21/1/3:6 This was written almost 20 years before [Zahavi \(1999\)](#), but that book’s discussion of the fact that there must be a tacit self-referential (reflective?) component to any first-order or base-level thought, in order for explicit reflective thought to be possible, was music to my ears. One of the limitations to the semantical analysis of 2/3Lisp is that this self-referential aspect—what in [Q3](#) I call its inherently deictic character—is not made explicit. But it is certainly there.
- A23** :21/2/7 The “field,” in this line, refers to Artificial Intelligence.
- A24** :21/2/-2:-1 A significant understatement. Cf. [ch. 2](#).
- A25** :22/0/-2:-1 As noted at [annotation A12.5](#), above, cf. the opening sections of [ch. 12](#), which discuss the ambiguity between the two interpretations of ‘knowledge representation’: (i) where knowledge is what is represented, and (ii) where knowledge is what is representational.
- A26** :22/1/6:7 This passage, too (as well as :18/1; see [A10](#), above), would have been better if framed in terms of a personal/subpersonal distinction. Cf. [annotation A26](#) of [ch. 3b](#), p. :115.
- A27** :22/1/7:-1 «Refer to general discussions in [Intro](#)? Also point forward to [100 Billion](#)»
- A28** :23/2/24/0 Needless to say, my own wrestling with representation has continued for many years. Cf. the discussion of ontological problems in [ch. 1](#), and the whole of “Rehabilitating Representation,” in [IA Volume II](#).
- A29** :25/1/17:20 The claim that it would be “unmanageably complicated” even to attempt to build a self-referential system except if guided by a theory of semantics might seem to be challenged by the subsequent development of such systems as Brown ([Friedman «...»](#)) Blond (**Danvy «...»**) and M-Lisp ([Muller «...»](#)). M-Lisp, however, was at least to an extent guided by a theory—that of formal logic. Brown and Blond were not—though in virtue of that very fact it is hard to know whether they should genuinely be counted as self-referential. Both languages are claimed by their authors to be reflective—but as discussed in [ch. 2](#), I do not believe such claims can be substantiated, and perhaps are not even warranted, given their lack of any theory of “representational” or “referential” semantics.
- A30** :26/0/-2 The word ‘yet,’ present in drafts written right up until the point of submission, was for unknown reasons deleted in the submitted version. Since I believe it should have been included, I have inserted (and marked) it here.