Meta in Logic

Donald Perlis

Computer Science Department

Institute for Advanced Computer Studies

University of Maryland
College Park, Maryland 20742

Abstract: An essential function of language and thought is to attribute properties to things, which then provide statements about those things. The study of this is the basis for a meta-language, i.e., a language *about* language. Yet property attributions are things in their own right, and so a meta-language can be regarded as (part of) an object language as well. This fact has numerous repercussions, and has been (at least indirectly) the basis for various studies in logic and artificial intelligence. We attempt to sketch some of this work.

descriptors: meta-mathematics, quotation, modal logic, self-reference, belief, paradox, aboutness, situated logic, reflection.

I. Introduction

Meta refers to aboutness, a stepping-back from something to survey its context and make assertions about it and that context. In this, it is closely related to so-called higher-order reasoning, in which the domain of entities under consideration is expanded to include not only the original objects but also properties of those objects. For instance, instead of simply dealing with the facts that 3 is prime, 5 is prime, and so on, we may wish to consider whether any arithmetic progression has an infinite number of primes. The latter can be formulated in a second-order language for arithmetic, and is in some sense a statement about the expressions of the simpler first-order language, at least to the extent that some wffs of first-order arithmetic can be identified with arithmetic progressions. The notion of a meta-level usually enters the sphere of formal logic by means of a meta-language in which the notion of logical consequence (or of formal proof) is expressed. The metalanguage may be an informal language such as English, or may itself be a formal language. Indeed, it may even be the very language of the original theory itself, and many of the most interesting and useful situations are of this sort. In such cases, the term ‘‘reflection’’ seems especially apt, for there is a sense of a self-dealing or introspection involved. Much of what follows will illustrate how this comes about and what problems are involved.

Specifically, provability of a wff amounts to a property (provability) attributed to a thing (the wff). This is the central example that will run throughout our discussion, and which has been the central case motivating research even when departures from it are taken, as in the case of believability (of a wff), or necessity, or any other property.

Suppose given a theory T in a language L (for illustration we take L and T to be first-order). Then a wff P(b) intuitively ‘‘says something about b.’’ So already in L itself there is at least a superficial notion of aboutness. Thus we might regard P as residing at a meta-level somewhere above the domain where b resides. Indeed, the common way of referring to an interpretation I of wffs of L supports this notion: b is interpreted as an element of the domain of I, whereas P is interpreted as a relation on that domain. Suppose we wish to be more explicit, and express the fact that the thing, P, is being attributed to the thing, b. We might wish to write Attrib(P,b). Now the role of P has shifted subtly: it is no longer a predicate letter but now instead an argument or term, more properly written (if it is to yield a wff of a first-order language L) as Attrib(‘P’,b). Here ‘P’ is not P but rather some other symbol that presumably bears an appropriate relationship to P via suitable axioms. The introduction of a name for P is called the reification of P, suggesting that P has been turned into (or viewed as) a ‘thing’ (from the Latin, re, for thing). It may be better to suppose that P has been turned into a thing-of-discourse, i.e., a term, for it certainly was already a formal object in its own right. Now Attrib(‘P’,b) may have a stronger meta-flavor to it than did P(b), in that Attrib(‘P’,b) seems to be about P(b); but the difference, though important, is slight. Thus the very coupling of symbols into application-pairs as in P(b) already carries the essence of aboutness in it. Note that Attrib may or may not be a symbol in the same language L; typically one regards it as belonging to a meta-language M, but it is not forbidden that M=L.

Historically, interest in a meta-language seems to have arisen with Hilbert’s program to formalize mathematics and prove theorems about that formalization, i.e. meta-theorems. Such meta-theorems are about the notion of formal proof or logical consequence, and hence go beyond the remarks above. However, the first successful effort to incorporate meta-theorems into the self-same language L was made by Godel [1931], and indeed provided names such as ‘P’ for symbols P as a step toward this goal. It turns out that many of the difficulties surrounding the use of meta-languages in artificial intelligence hinge on the presence of such names even without the additional notion of provability being formally incorporated.

We might take as a somewhat typical statement in a meta-language M for L (where M need not be the same as L) the following:

The wff α of L is provable in the theory T.

Here M is English, while L is (say) a first-order language. Often this is written instead as “T |-- α” but this is merely an abbreviation for the English expression, and is not formal despite its concise technical appearance. However, Godel decided to endow L itself with an analogue of |--, via a predicate symbol Thm (actually Bew for Beweis). Then wffs of L include ones such as Thm(“α”) where L now also contains names for its own wffs (‘α’ for α). Clearly some sort of recursive definition is needed, since adding new names augments L so that more wffs can be formed, which then in turn need names of their own, etc. Details are fairly straightforward. (Note that even the less ambitious case of Attrib above requires the same preliminaries of providing names for wffs.)

This process of representing meta-notions for L internally to L via names is often called arithmetization of meta-mathematics, since in Godel’s hands the name ‘P’ for an expression P was in fact an integer (the Godel number of P), and axioms for arithmetic were used to study properties of expressions of L internally to the theory T in that same language L. Thus the notion of provability in T became at least somewhat expressed within T itself, in terms of Thm. The extent to which this expression of provability could coincide with the true statements of arithmetic in L was given in Godel’s Incompleteness Theorem: as long as T’s set of axioms includes ordinary arithmetic and is decidable and consistent, then the set of theorems of T will either contain some false arithmetical statements or not contain some true arithmetical statements.

Now, if Thm indeed expresses provability in T, then we would expect that whenever P is provable in T, then so is Thm(‘P’) and vice versa:

T |-- P iff T |-- Thm(‘P’)

and indeed this is the case for the definition of Thm given by Godel. Note that this can be conveniently stated in the form of two rules of inference:

P | Thm(‘P’), and Thm(‘P’) | P

For reasons to appear below, we call these the rules of positivity for Thm. Thus it appears that the case for Thm is fairly straightforward. We will see in section III, however, that this is not so.

We will gather our tour of meta in logic into several convenient conceptual clumps as follows: section II deals with what we shall call the positive problem of reflection, section III with the negative problem. Roughly, these deal with the semi-decidable (resp. semi-undecidable) half of determining logical consequences. Representing provability (or consequentiality) at the meta-level runs into difficulties in both of these spheres, but ones that have given way to some progress.

Then in section IV, we view logic and the meta-level from a more general perspective including the environment (or engine) in which a formalism is manipulated. This we refer to as situated logics, which seem to lend themselves to the use of a language with built-in meta-features. Finally, in section V, we discuss how situated logics might be applied to the problem of meaning, in the sense of formal representations that can be said to be about something for the situated logic itself.

II. The positive problem of reflection

Well then, what are the difficulties? Chief among them is the aforementioned need to suitably relate symbols and their names. We already saw that both for Attrib and Thm, such a mechanism was needed. We also saw that for Thm, the relation between it and that which it is about, namely provability, cannot go so far as to fully capture what is true of arithmetic. But can it fully capture even what is provable? Yes and no. Before going into this, let us consider further the idea of relating symbol and name.

The most obvious claim to make relating expressions P and ‘P’ is not one of provability of P (in T) iff Thm(‘P’) is provable (in T), but rather simply that the meaning of P is given by ‘P’, as in the familiar examples

‘Snow is white’ is true iff Snow is white, or

‘Snow’ means snow.

That is, even without the selection of a theory T, a language L carries the understood intention of having interpretations, and a name ‘P’ in L for an expression P in L will be ‘for’ only if there is some formal connection between them. The predicate ‘true’ provides such a tie, although a problematic one, as we shall see.

At this point we should sketch a kind of touchstone of what ideal outcome might be anticipated, even if only to knock it down later. It might appear now that we need only go all the way with the above ideas, applying them to other meta-notions (in addition to Thm) to achieve a nice set of axioms for all significant notions of formal reasoning:

$\text{True}(\text{‘P’}) \leftrightarrow \text{P}$

$\text{Believes}(\text{‘P’}) \leftrightarrow \text{Thm}(\text{‘P’})$

$\text{Necessary}(\text{‘P’}) \leftrightarrow \ldots$

$\text{Probable}(\text{‘P’}) \leftrightarrow \ldots$

$\text{Consider}(\text{‘P’}) \leftrightarrow \ldots$

$\text{Should-Do}(\text{‘P’}) \leftrightarrow \ldots$

$\text{Erase}(\text{‘P’}) \leftrightarrow \ldots$

$\text{Introspect-for}(\text{‘P’}) \leftrightarrow \ldots$

etc.

The astute reader of the Proceedings for the Sardinia Workshop on Meta-Architectures and Reflection will notice many of these notions appearing. We have deliberately avoided filling in defining axioms above, except in the cases of True and Believes (in the latter for illustration we have followed a suggestion of Konolige [1982] to identify an agent’s belief of P with provability of P by that agent).

We now turn to troubles arising from such ambitions. The first is due to Tarksi [1936], and is a variant of the Liar paradox. Tarski showed that the most obvious definition of the predicate True, given above, namely

$$\text{True}(\text{‘}\alpha\text{’}) \leftrightarrow \alpha$$

is inconsistent in a theory T having (sufficient) arithmetic. This is surprising, and also problematic for artificial intelligence, for the notion of an agent knowing a wff α seems to be based on a clear sense of what it means for α to be true. In fact, knowledge of α has often been taken to mean justified true belief of α (though this too is problematic, as shown by Gettier[1963]). However, a simple variation on the Tarski schema above turns out to be consistent [Feferman 1984, Perlis 1985]:

$$\text{True}(\text{‘}\alpha\text{’}) \leftrightarrow \alpha^*$$

where the * is itself a variation on an operator due to Gilmore [1974], and has the effect of replacing in α all occurrences of $\neg\text{True}(\text{‘}\beta\text{’})$ by $\text{True}(\text{‘}\neg\beta\text{’})$.

The next problem that surfaced for a formal treatment of rational agents was discovered by Montague [1963], and can be viewed as dealing with belief as follows. The axiom schema (where we use Bel for belief)

$$\text{Bel}(\text{‘}\alpha\text{’}) \rightarrow \alpha$$

together with the rule

$$\alpha \mid \text{Bel}(\text{‘}\alpha\text{’})$$

is inconsistent with (sufficient) arithmetic. This result apparently has had a negative impact on formal work in knowledge representation. However, several positive responses can be made. First, let us see how Montague’s theory arises as a plausible formalization of belief, especially in light of Konolige’s alternative one using Thm above. There are two general perspectives here. One is that the underlying language L and theory T are those *of* the agent, so that each axiom and theorem is something believed by the agent and Bel is the agent’s way of reflecting on its own beliefs. The other is that L and T are *our* tools for analyzing the agent, and that T then is our theory about the agent so that T’s theorems are facts we may establish and Bel(‘α’) is our way of expressing that α is believed by the agent. It turns out that what we will say below applies almost equally to either of these interpretations, with minor modification the reader can supply; so we will adopt the former view for illustration.

If we suppose ourselves faced with an ideal reasoning agent R, then we may as well suppose that R has a predicate Bel of its own, to record to itself that certain expressions are believed. Now R, being ideal, makes no mistakes and so believes only truths. But if R is ideal, should not R also know (or believe) this? That is, should not R have as a fact available to its own reasoning, that all its beliefs are true? This however is Montague’s first axiom (in schema form). Now, this already flies in the face of the Thm interpretation of Bel, since a result of Lob [1955] See [Boolos and Jeffrey 1980]. shows that Thm(‘β’) $\rightarrow \beta$ is a theorem (of a given theory T for which Thm is Godel’s predicate) iff β is already provable in T. That is, in general we *cannot* have Thm(‘β’) $\rightarrow \beta$ and so cannot identify Thm and Montague’s Bel. On the other hand, Montague’s axiom schema rings true, so we must look further. What about his rule of inference?

The rule α | Bel(‘α’), in our interpretation, simply means that the agent R, having proven α, can now affirm that he believes α. This seems harmless and desirable. Thus Montague has indeed presented us with a puzzling situation. It seems that ideal reasoners cannot exist. Now this would be bad enough, but it turns out that a notational variant on the schema and rule, namely a modal logic in which Bel is a modal operator, *is* consistent! That is, replacing Bel(‘α’) with Bel α in the schema and rule (and modifying the underlying language and theory according to the standards of modal logics, the problems (seem to) disappear.

What is going on? Does modal logic somehow lend itself to reflective reasoners with self-beliefs, and not first-order logic? Cannot a reasoner use first-order logic to reason *about* its reasoning in good meta-spirit? Well, several comments are in order. First, if we are willing to relinquish the ideal of a perfectly infallible reasoner, then the schema is no longer so convincing. Indeed, a standard mark of wisdom is the recognition of the possibility of error, so that one might even prefer the *negation* of Montague’s schema:
(x)(Bel(x) & ¬True(x)). Second, des Rivieres and Levesque [1986] have shown that a syntactically careful translation of consistent modal theories will preserve consistency of suitable first-order variants, although at the expense of some intuition. Perlis [1987] showed that a trick like that for True above can be employed as well for Bel to salvage most of Montague’s theory without contradiction. Asher and Kamp [1986] also have an analysis of this problem, based on approaches of Gupta [1982] and Herzberger [1982] to the paradoxes of truth.

In a vein similar to Montague’s, Thomason [1980] has provided an even more dramatic failure of our intuition, in the following paradox: If an agent R believes (a suitable theory of) arithmetic and also R’s beliefs satisfy the following conditions (given by the predicate Bel):

Bel(‘α’) $\rightarrow$ Bel(‘Bel((‘α’)’)
Bel(‘Bel((‘α’)’ $\rightarrow \alpha$’)
Bel(’α’) for all valid α
Bel(‘$\alpha \rightarrow \beta$’) $\rightarrow$ (Bel(‘α’)$\rightarrow$Bel(‘β’))

then R is inconsistent in the sense that R will believe all wffs.

One can levy much the same critique for Thomason’s theory as for Montague’s. In particular the second axiom schema above attributes to R a naive belief in R’s own infallibility, even in light of the known paradoxical tendencies of self-referential systems. A more cautious reasoner might take a less broad-brushed stand on the truth of its own beliefs. The issue that Thomason and Montague exploit is precisely one of vicious self-reference that is built into the language they employ; if R is to be very smart, R should realize this and therefore couch its beliefs accordingly. This is the thrust of the results of Feferman and Perlis.

Richard Weyhrauch private communication, October 1986 pointed out that the underlying spirit of omniscient (ideal) reasoners seems perfectly consistent with the further assumption of negative introspection, i.e., if a wff α is *not* a theorem of the agent, then the agent should not only fail to believe α, but also should be able to *prove* that it does not believe α. We will see more of this kind of notion in the next section.

A final observation [Perlis 1986] is that a modest extension of such reasoning systems as we have explored above, namely one with a memory of its past conclusions, quickly leads to inconsistency even when given the *negated* Montague axiom of fallibility! However, the apparent lesson is that commonsense reasoning systems may have to deal with inconsistent data and that ideal reasoners must be dismissed as impossibly unrealistic even for theoretical studies.

III. The negative problem of reflection

Now we return to the promised theme of Thm and the extent to which it can capture *fully* the idea of provability. We call this and related issues to be given, negative problems since they have to do with the syntactic negations of the predicates considered above. In fact, we start with Weyhrauch’s observation above, that one would expect a reasoner to be able to tell not only what it *does* know but also what it *doesn’t.* Recall the ‘rules of positivity’ earlier:

P | Thm(‘P’), and Thm(‘P’) | P

These allow the theory in question to decide positively instances of actual theorems, but do not address the case of non-theorems. However, one might reasonably wish to know when a wff is *not* a theorem. This was what Weyhrauch suggested as a desideratum for an ideal reflective reasoner. That is, we wish to be able to prove suitable instances of wffs such as ¬Thm(‘α’) and ¬Bel(‘α’). Note that consistency tests are of this sort, for consistency amounts to the non-provability of certain wffs; thus reflectively concluding self-consistency (of a given theory) amounts to solving part of the problem of negative reflection.

In general, however, this problem is undecidable. That is, although true instances of Thm (relative to a standard model) may be semi-decidable (we may be able to determine in a general way whenever a wff is a theorem of some given theory), they are not fully decidable: we may not be able to determine whenever a wff is *not* a theorem. The distinction lies precisely in the fact that the length of time taken to determine the former is unbounded so that we may never know that all possibilities for a proof have been checked. Of course, for an ideal reasoner with access to an oracle, this need not be a problem. Nevertheless, it is of considerable interest to study the less-than-ideal case in which computable answers may be obtainable.

Familiar instances of negative reflection in the artificial intelligence folklore are:

$$\neg\text{Bel}(`\alpha') \rightarrow \text{Bel}`\neg\text{Bel}(`\alpha')' \text{ [negative introspection]}$$

|-/- α implies |-- $\neg\alpha$ [negation as failure]

$$\frac{\beta : \text{Consis } \alpha}{\alpha}\text{[default rule]}$$

$$\frac{\alpha \rightarrow \text{Bel}(`\alpha') : \text{Consis}(\neg\text{Bel}(`\alpha'))}{\neg\alpha}\text{[autoepistemic rule (roughly)]}$$

But how do we establish something like ¬Bel('α') (or its consistency) in the first place? One glimmer of hope here is McCarthy's [1980, 1984] technique of circumscription. It provides a short-cut to consistency tests, by means of inner (syntactic) models; being semi-undecidable (essentially first-order proof-techniques) this cannot be strong enough to produce all desired results of negative introspection (nothing can). But it does produce very many cases, and there is hope that it may be sufficient to capture a large chunk of the cases of interest to commonsense reasoning. In effect circumscription (partially) reduces the negative reflection problem to the positive reflection problem: it determines ''positively'' that certain things *are* proven, and on that basis concludes certain others are false. As such it ios not a sound inferential mechanism, but seems to be so for the case of Bel urged here, at least for agents that can indeed introspect. In particular, one might hope to be able to determine many cases of ¬Bel simply by circumscribing Bel with respect to all axioms. This would have the advantage of being, in many cases, defeasible in the sense given by Moore regarding autoepistemic reasoning: the conclusion that ¬Bel('α') when in fact Bel('α') is not provable is (usually) sound, and in particular is so for the ideal reasoners in much of the literature.

IV. The problem of situated logics

We begin with two truisms:

Logic does nothing.

Engines do all.

The point of these bold but essentially tautologous statements is twofold:

> It is often claimed that logic (of a certain stripe) can (or cannot) *do* this and that, whereas doing something is the arena of an engine rather than a language or a specification of an inferential relation. An engine may *use* a logic.
>
> When an engine uses a logic, it always does so in a context, both of control strategies and of external events that constitute the actual real-time behavior of the engine (electric power and so on). The use of a logic always depends on the *situatedness* of the engine involved, i.e., an engine provides the situation in which a logic is put in touch with the real world.

In a familiar expression [Kowalski 1979], algorithm = logic + control. Doing things is the domain of a control mechanism; the logic may consist of fodder which that mechanism utilizes. The control design is a matter of choice not necessarily tied to anything strictly related to the logic (syntax). However, typically one is interested in cases in which there is some significant relationship between the two, even perhaps in which there is some syntactic representation (declarative knowledge) of the control to facilitate reflective reasoning by the 'system' *within* its logical structure *about* both that structure *and* its own control. For instance, the system, R, might draw the conclusion that some of its own axioms or rules are fallacious, or that they are all valid, or that it will take it many hours to determine whether it contains a contradiction, or that it may never so determine (e.g., due to its control which may keep it from spending too much time on any one problem, or due to its consistency which however may not be provable by R). We see then that reflection idea arises naturally in the context of control.

The theme of logic + control is an old one, and not surprisingly, since logics and their formal theorems become elements of a reasoning system only when provided with an embedding into that system, i.e., only when situated in some particular manner in that system. This idea is at least implicit in the situation calculus of [McCarthy&Hayes 1969] and [Green 1969], in production systems [Newell and Simon 1972], in semantical studies [Barwise and Perry 1983], in planning research [Suchman 1985], and in work on limited reasoning [Fagin and Halpern 1985, Levesque 1984, and Drapkin and Perlis 1986]. It is also implicit in default reasoning, in which there is the spector of reality contradicting assumptions made on the basis of limited knowledge. What seems to be gaining more attention lately is that meta-knowledge *of this sort*, that is, pertaining to the situatedness of the engine that 'pumps' the logic, is crucial to many kinds of intelligent behavior. An example from [McDermott 1982] as discussed in [Drapkin and Perlis 1986] is that of Dudley and Nell: Nell is tied to the railway tracks and in danger of being crushed by an onrushing locomotive; Dudley must save her. Here it is essential that Dudley's planning be seen *by him* to occur in the course of time in which Nell's plight becomes more and more precarious as the train approaches. Hence Dudley does not have the luxury of looking ahead to the most thought-out conclusions. The approach suggested in [Drapkin and Perlis 1986] is to picture the inference engine according to the maxim that its deductive steps take place in the same flow of time as everything else. An implementation of this kind of reasoning is in progress employing a memory-based model as in [Perlis 1984].

Situated logics will presumably play an increasingly central role in artificial intelligence in the future.

V. The problem of aboutness

One connotation of the word 'reflection' is that of the presence of a semantic content to the symbolic processes in question, that is, to there being an *object* of reflection. One reflects *on* something. Just how words or symbols get these 'meanings' is the famous problem of aboutness or reference. In short, how is it that a program or other computational entity can assign meanings (of its own) to its symbols? This is often discussed in terms related to the mind-body problem, and the problem of consciousness: how can a material/mechanistic entity have mental states? If such a program can be built then it may be fair to say that it would be reflecting at a high level. This then seems an appropriate theme to address as we end our survey of reflection.

Much has been written on this problem, largely by philosophers. There seem to be at least three main camps. Some (dualists, e.g., [Popper 1965]) argue that meaning is not a mechanical notion at all, but that it is a dual phenomenon to material aspects of behavior, and therefore not to be understood in ordinary scientific modes of discourse. Others (functionalists, e.g., [Dennett 1978]) claim that meaning is merely a useful terminological category one agent uses in reasoning about (the functioning of) another (the ''intentional stance'' of outsiders). Still others (empiricists, e.g., [Thagard 1986]) argue that there may be internal phenomena that clearly produce genuine 'attribution of meaning' within an agent, but that the specific character is yet to be discovered. This latter contention is addressed (at least to some degree) by recent work in artificial intelligence, sketched below.

Sloman [1986] and Steels [1986] have suggested that there is an important contextual phenomenon regarding *internal* symbolism in computational behavior. That is, there is an aspect to the execution of programs that bears on environmental events in a very direct way, namely the internal states of the executing hardware and software itself. This is a physical phenomenon and hence an environment even though it is not external to the computer. Still, when a value is read (copied) from a storage register or other (virtual) memory location, the resulting copy bears a physical relation (identity or equivalence) to the original and can be said to be *about* the original in ways that are significant for the ongoing functioning of the program execution. Thus what is a tenuous link in general between symbol and symboled, seems a firmer sort of thing for these internal cases.

Note that the observation of Sloman and Steels can also be stated in terms of a reasoning entity that contains both the symbol *and* the symboled (reference) within itself. This suggests that explicit incorporation of the representing phenomenon may be what actually creates representing behavior in the first place. Thus uttering ''the cow'' calls attention to a presumed cow *and* to the uttering of ''the cow''. I.e., 'cow' is used *both* as animal and as word. What is the advantage of such double-duty of symbols? Well, it may account for the ability to reconsider, while not losing sight of the original attitude. One may consider whether the supposed thing out there is really a cow, while remaining very aware that one had just described it with the expression ''the cow''. Thus reflection may confer the ability to acknowledge and deal with the possibility of error or conjecture. The upshot is that the word ''cow'' should (normally) mean cow *to the program* and not just to the programmer. For this, the *use* of ''cow'' should be under the control of the program and hence changeable by the program as it sees fit, e.g., as it considers whether what is (situated!) in front of it is really a cow. Or, it may decide to use '4-legged ruminant' instead of 'cow', if it thinks there is confusion regarding the latter, and thereby claims ''that thing in front of me is a 4-legged ruminant.'' Two expressions (''that thing'' and ''4-legged ruminant'') are being matched. This *presumption of an entity of discourse* as reflected in the more general expression ''that thing'', seems to capture part of the ideas present in the contributions of Sloman and Steels. The syntactic expressions are internally tied to this presumption, as well as the fact that this tie itself can be reflected (quoted) into another expression, possibly lending itself to flexibility and error-correcting.

VI. Conclusions

Reflection of a meta-language into the object language plays many roles in formal logic, and in particular seems destined to occupy a principal place in the arena of intelligent mechanical systems. Our brief survey indicates strong ties with many topics, from modal logic to proof theory, from paradox to meaning. It is suggestive to this author that we have only touched the tip of the iceberg, and that far richer aspects of reflection of the meta-level in logic will turn out to be involved in intelligent behavior.

In the interests of greater completeness than this brief survey can manage, the bibliography contains various entries not discussed in the text; here we simply point out a few additional items. The idea of reification was emphasized by McCarthy [1979] and pursued by Creary [1979], Attardi&Simi [1981], and Haas [1981,82]. A separate effort related to reification has been pursued vigorously in mathematical logic, by Church and Curry and others in the domain of lambda calculus, which also has applications to the semantics of programs (see Stoy [1977] and Barendregt [1984]).

Acknowledgements

I would like to thank the organizers of the Workshop on Meta-Level Architectures and Reflection, held in Sardinia in October 1986, for their invitation to prepare this paper, based on my survey lecture, for inclusion in this book. I also wish to thank Carl Hewitt, Richard Weyhrauch, Luc Steels, Brian Smith, Maria Simi, Hector Levesque, Jim de Rivieres, Pattie Maes, Rosalie Hall, and others, for interesting and useful comments.

This work has been supported in part by the following institutions:
The U.S. Army Research Office (DAAG29-85-K-0177)
The Martin Marietta Corporation.

Bibliography

Asher, N. and Kamp, H. [1986] The Knower’s paradox and representational theories of attitudes, *Proceedings, Theoretical Aspects of Reasoning About Knowledge,* March 12-22, 1986, Monterey, 131-147.

Attardi, G. and Simi, M. [1981] Consistency and completeness of OMEGA, a logic for knowledge representation. *IJCAI-81*, 504-510.

Barendregt, H. [1984] *The Lambda Calculus*. North-Holland.

Barwise, J. and Perry, J. [1983] *Situations and Attitudes*, MIT Press.

Boolos, G. [1979] *The Unprovability of Consistency*. Cambridge University Press.

Boolos, G. and Jeffrey, R. [1980] *Computability and Logic*, 2nd edition. Cambridge University Press.

Burge, T. [1984] Epistemic paradox, *J. Phil., 81*, 5-29.

Burge, T. [1979] Semantical paradox, *J. Phil., 76*, 169-198.

Chellas, B.[1980] *Modal Logic*. Cambridge University Press.

Creary, L. [1979] Propositional attitudes: Fregean representation and simulative reasoning. *IJCAI-79*, 176-181.

Dennett, D. [1978] *Brainstorms*. Montgomery, VT: Bradford Books.

des Rivieres, J. and Levesque, H. [1986] The consistency of syntactical treatments of knowledge. *Proceedings of the Conference on Theoretical Aspects of Reasoning About Knowledge.* Monterey, October 1986.

Drapkin, J. and Perlis, D. [1986] A preliminary excursion into step logics, *Proceedings, International Symposium on Methodologies for Intelligent Systems*, Knoxville, Tennessee, October 1986.

Eberle, R. [1974] A logic of believing, knowing and inferring. *Synthese 26*, 356-382.

Elschlager, B. [1979] Consistency of theories of ideas, *IJCAI-79*, 241-243.

Fagin, R. and Halpern, J. [1985] Belief, awareness, and limited reasoning: preliminary report. *IJCAI 85*, 491-501.

Fagin, R., Halpern, J., and Vardi, M. [1984] A model-theoretic analysis of knowledge. *Proc. 25th IEEE Symp. on Foundations of Computer Science*, 268-278.

Feferman, S. [1984] Toward useful type-free theories, I. *J. Symbolic Logic, 49*, 75-111.

Gettier, E. [1963] Is justified true belief knowledge? *Analysis 23*, 121-123.

Gilmore, P. [1974] The consistency of partial set theory without extensionality, in T. Jech, (ed.) *Axiomatic Set Theory*, Amer. Math. Soc., 147-153.

Godel, K. [1931] Uber formal unentscheidbare Satze der Principia Mathematica und verwandter Systeme I, *Monatsh. Math. Phys.*, 38, pp. 173-198.

Green, C. [1969] Application of theorem proving to problem solving, *IJCAI-1*, pp. 219-239.

Gupta, A. [1982] Truth and paradox, *Journal of Philosophical Logic 11*, 1-60.

Haas, A. [1982] Planning mental actions. Ph.D. thesis, University of Rochester.

Haas, A. [1981] Reasoning about deduction with unknown constants. *Proc. 7th IJCAI*, pp. 382-384.

Halpern, J. and Moses, Y. [1985]A guide to the modal logics of knowledge and belief: preliminary draft. *IJCAI 85*, pp.480-490.

Halpern, J. and Moses, Y. [1984] Towards a theory of knowledge and ignorance. *AAAI workshop on Nonmonotonic Reasoning*, New Paltz.

Herzberger, H. [1982] Naive semantics and the Liar paradox, *Journal of Philosophy 79* 479-497.

Hintikka, J. [1962] *Knowledge and belief*. Cornell University Press.

Hughes G., and Cresswell, M. [1968] *An introduction to modal logic*. Methuen.

Israel, D. [1980] What's wrong with nonmonotonic logic? *Proc. First Annual National Conference on Artificial Intelligence.*

Konolige, K. [1982] A first-order formalization of knowledge and action for a multiagent planning system, *Machine Intelligence 10*, 41-72.

Konolige, K. [1984] Belief and incompleteness. SRI Tech. Note 319.

Konolige, K. [1985] A computational theory of belief introspection. *IJCAI 85*, pp.503-508.

Kowalski, R. [1979] Algorithm=Logic+Control, *CACM*, August 1979.

Kripke, S. [1975] Outline of a theory of truth, *J. Phil., 72*, 690-716.

Kripke, S. [1963] Semantical analysis of modal logic. *Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik, 9*, 67-96.

Levesque, H. [1984] A logic of implicit and explicit belief. *Proc 3rd National Conf. on Artificial Intelligence*, 198-202.

Levesque, H. [1986] Making believers out of computers, *Artificial Intelligence, 30*, 81-108.

Lob, M. [1955] Solution of a problem of Leon Henkin, *Journal of Symbolic Logic*, 20, pp. 115-118.

McCarthy, J. [1980] Circumscription--a form of non-monotonic reasoning, *Artificial Intelligence 13*, 27-39.

McCarthy, J. [1979] First order theories of individual concepts and propositions, *Machine Intelligence 9*, 129-147.

McCarthy, J. and Hayes, P. [1969] Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*, Meltzer, B. and Michie, D. (eds.), Edinburgh University Press.

McDermott, D. [1982] A temporal logic for reasoning about processes and plans, *Cognitive Science*, 6, pp. 101-155.

McDermott, D. and Doyle, J. [1980] Non-monotonic logic I. *Artificial Intelligence*, 13 (1,2), pp. 41-72.

Montague, R. [1963] Syntactical treatments of modality, with corollaries on reflexion principles and finite axiomatizability, *Acta Philosophica fennica, 16*, 153-167.

Moore, R. and Hendrix, G. [1979] Computational models of beliefs and the semantics of belief-sentences, SRI Tech. Note 187.

Moore, R. [1977] Reasoning about knowledge and action. *IJCAI 77*, pp.223-227.

Newell, A. and Simon, H. [1972] *Human Problem Solving*, Prentice Hall.

Perlis, D. [1984] Nonmonotonicity and real-time reasoning, *AAAI Workshop on Nonmonotonic Reasoning,* New Paltz.

Perlis, D. [1985] Languages with self-reference I. *Artificial Intelligence*, v. 25, 301-322.

Perlis, D. [1987] On the consistency of commonsense reasoning, *Computational Intelligence*, 2, 180-190.

Perlis, D. [1987] Languages with self-reference II, submitted to *Artificial Intelligence*.

Popper, K. [1965] *Conjectures and Refutations*. Basic Books.

Quine, W. [1946] Concatenation as a basis for arithmetic. *J. Symb. Logic, 11*.

Reiter, R. [1980] A logic for default reasoning, *Artificial Intelligence, 13.*

Rieger, C. [1974] Conceptual memory... Ph.D. thesis. Stanford University.

Sloman, A. [1986] Reference without causal links, *Proceedings, 7th ECAI*, July 21-25, 1986, Brighton, UK. 369-381.

Smorynski, C. [1985] *Self-Reference and Modal Logic*. Springer-Verlag, New York.

Steels, L. [1986] The explicit representation of meaning. *Proceedings, Workshop on Meta-Level Architectures and Reflection*, Sardinia, October 1986.

Stoy, J. [1977] *Denotational Semantics*. MIT Press.

Suchman, L. [1985] Plans and situated actions, Tech Report ISL-6, Xerox Palo Alto Research Center, February 1985.

Tarski, A. [1936] Der Wahrheitsbegriff in den formalisierten Sprachen, *Studia Philos., 1*, 261-405.

Thagard, P. [1986] Parallel computation and the mind-body problem. *Cognitive Science*, 10, 301-318.

Thomason, R. [1980] A note on syntactical treatments of modality, *Synthese* 44, pp 391-395.

Vardi, M. [1985] A model-theoretic analysis of monotonic knowledge. *IJCAI 85*, 509-512.

Weyhrauch, R. [1980] Prolegomena to a mechanized theory of formal reasoning, *Artificial Intelligence, 13*, 133-170.

Winograd, T. [1980] Extended inference modes in reasoning by computer systems, *Artificial Intelligence, 13*, 5-26.