SRI International

FORMAL THEORIES OF KNOWLEDGE IN AI AND ROBOTICS

Technical Note 362

September 10, 1985

By: Stanley J. Rosenschein
Artificial Intelligence Center
Computer Science and Technology Division

APPROVED FOR PUBLIC RELEASE:
DISTRIBUTION UNLIMITED

This work was supported in part by a gift from the Systems Development Foundation and in part by FMC under Contract 147466 (SRI Project 7390).

SRI International®
333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

Abstract

Although the concept of *knowledge* plays a central role in artificial intelligence, the theoretical foundations of knowledge representation currently rest on a very limited conception of what it means for a machine to know a proposition. In the current view, the machine is regarded as knowing a fact if its state either explicitly encodes the fact as a sentence of an interpreted formal language or if such a sentence can be derived from other encoded sentences according to the rules of an appropriate logical system. We contrast this conception, the interpreted-symbolic-structure approach, with another, the situated-automata approach, which seeks to analyze knowledge in terms of relations between the state of a machine and the state of its environment over time using logic as a metalanguage in which the analysis is carried out.

1 Introduction

This paper deals with some conceptual problems underlying software design in robotics and artificial intelligence (AI). Its aim is to explore the theoretical foundations of knowledge representation with a view to developing a rigorous mathematical theory of objective information content that could be used to define the semantics of knowledge representation schemes employed in AI and robotics.

The paper is organized as follows. Section 2 explains the need for a rigorous computational theory of knowledge and information for robotics-related tasks. As essential background material, Section 3 presents some formal theories of knowledge and action developed in philosophy, computer science, and AI. Section 4 examines two contrasting approaches to the problem of tying these formal models to actual programs: (1) the standard AI approach to knowledge representation, based on symbolic structures that are interpreted semantically by the designer and manipulated formally by the program; (2) a somewhat different approach based on correlations over time between the state of a system and that of its environment. In Section 5, we discuss several issues related to the application of the correlational approach to software design for robotics.

2 Why Study Knowledge Formally?

The need for a formal theory of knowledge can be seen most clearly in the analysis of multiple communicating processes. In designing a system involving multiple robots, communication between a human and a robot, or even a single robot with distributed control, there is often a need to model the information one part of the system has about other parts. Similar issues arise in natural-language understanding, distributed artificial intelligence (the design of intelligent cooperating agents), the analysis of communication in distributed computer systems, and protocol design [5]. In each of these areas researchers have found it necessary to model what process A knows about process B's knowledge.

Upon further reflection it becomes clear that even the design of a single intelligent program usually involves modeling what one process knows about another. The reason for this is simple: AI systems (with the possible exception of mathematical theorem-provers) are intended to have knowledge *about something*, namely, the surrounding world in which they are embedded. For instance, an intelligent robot should have knowledge of the environment it senses and controls. In describing this situation formally, the surrounding environment is most naturally modeled as a second process, which, from a theoretical standpoint, moves the problem into the realm of distributed systems and raises all the usual questions about modeling the knowledge one process possesses about another.

Every major subarea of AI can be described in a way that highlights the importance of the concept of knowledge:

- **Perception** has to do with an agent's acquiring *knowledge* about its environment by interpreting sensory input.
- **Planning** has to do with an agent's acting on the basis of its *knowledge* of the consequences of its potential actions.
- **Reasoning** involves an agent's deriving conclusions from facts it already *knows*.
- **Learning** involves incrementing *knowledge* through experience.
- **Communication** (e.g. in natural language) involves the continual updating of mutual *knowledge* possessed by the speaker and hearer.

Thus, knowledge lies at the heart of every important area of AI. Indeed, if the popular press is to be believed, there is even an emerging "knowledge industry." People are beginning to talk about knowledge in AI systems as if it were a commodity that can be bought, sold, and packaged. From a theoretical standpoint, this is somewhat startling, since current theories of knowledge do not provide satisfying answers to even the simplest questions: *What is knowledge? How can we describe it formally?* These questions are hardly new; philosophy has been asking the first for several millenia in a discipline called epistemology, while formal philosophy has been addressing itself to the second question for at least the past several decades. For AI researchers and engineers, these two questions immediately give rise to a third: *How can we systematically build computer systems that possess world knowledge and the ability to act upon it?*

3 Formal Approaches to Knowledge in Philosophy, Computer Science, and AI

The traditional starting point in modeling knowledge derives from the observation that knowledge involves a relation between an *agent* (the knower) and some *facts* (the objects of knowledge.) We express the relation between agents and the propositional objects of knowledge by using the notation $K_x p$, which means that *agent* x knows *fact* p. (We omit the subscript when the agent is understood from the context.)

3.1 Epistemic Logics

Many formal systems have been proposed for modeling what agents can know about the world—particularly the relationship between the *facts known* and the *facts about knowing* [8,11]. This branch of formal philosophy, known as epistemic logic, is too rich to review here, but, to impart a sense of the type of formalism proposed, we present a simple propositional modal logic of knowledge that we shall call EL (for "epistemic logic"). From the formal standpoint, EL is just the modal logic S5 [9], with the necessity operator interpreted as

the "knowledge" operator K. Our purpose here is not to argue for this particular model of knowledge, but simply to illustrate the application of logical techniques to the modeling of knowledge and to set the stage for a discussion of knowledge representation issues in AI.

EL is a logical system consisting, as usual, of symbols, syntactic formation rules, axioms, rules of inference, and a formal model theory. It differs from ordinary propositional logic in three ways: (1) it has the operator K among its symbols and its formulas are closed under this operator; (2) in addition to the axioms and rules of inference of ordinary propositional logic, there are several special axioms and rules involving the K operator; (3) the model-theoretic semantics of the logic is given in terms of a set of "possible worlds" and an epistemic "accessibility" relation (described below). EL is designed to describe a single knower, so the "agent" parameter is left implicit—expressed neither in the logical language nor its formal semantics.

We begin by defining the symbols of the formal language:

Symbols: $P = \{p_1, p_2, ...\}$ (atomic formulas),
$\wedge, \neg$ (Boolean connectives),
K (knowledge operator),
$(,)$ (punctuation).

Next, we give the formation rules.

1. If p is an atomic formula, then p is a formula.
2. If ϕ and ψ are formulas, then so are $(\phi \wedge \psi)$, $\neg\phi$, and $K\phi$.
3. Nothing else is a formula.

The connectives $\vee, \rightarrow, \leftrightarrow$ can be defined in the usual fashon: $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$, and so on. We also omit parentheses according to the usual conventions. Next we provide some axioms and rules of inference.

Axioms:

1. Axioms of propositional logic.
2. $K\phi \rightarrow \phi$ (truth).
3. $K(\phi \rightarrow \psi) \rightarrow (K\phi \rightarrow K\psi)$ (consequential closure).
4. $K\phi \rightarrow KK\phi$ (positive introspection).
5. $\neg K\phi \rightarrow K\neg K\phi$ (negative introspection).

Rules of Inference:

1. From ϕ and $\phi \rightarrow \psi$ derive ψ (modus ponens).
2. From ϕ derive $K\phi$ (epistemic necessitation).

Finally, we define the semantics of the language.

Models: $\mathcal{M} = (W, \pi, \kappa)$, where

1. W is a nonempty set of *possible worlds.*
2. $\pi : P \rightarrow 2^W$ interprets atomic formulas as sets of worlds.
3. $\kappa \subseteq W \times W$ is a relation on worlds (the *epistemic accesibility relation*).

Satisfaction is defined relative to a model $\mathcal{M}$ and a world w (we supress reference to the model):

1. $w \models p$ if $w \in \pi(p)$ for all $p \in P$.
2. $w \models (\phi \wedge \psi)$ if $w \models \phi$ and $w \models \psi$.
3. $w \models \neg\phi$ if $w \not\models \phi$.
4. $w \models K\phi$ iff for all w', $\kappa(w, w')$ implies $w' \models \phi$.

Space does not permit a full discussion of the the logic presented here; the interested reader is referred to any standard treatment on modal logic [9] or its application to AI [14]. For our purposes it will be sufficient to explain the intuition behind the possible-worlds approach to the semantics of knowledge. It is useful to think of the possible worlds as representing alternative ways the world could be. The epistemic accessibility relation, $\kappa(w, w')$, indicates that world w' is a possible alternative state of affairs from the standpoint of what the agent knows in world w. Since the agent in a given situation doesn't know all the details of the world, the world might be this way or that way, so far as he knows.

To each world there corresponds a set of formulas: the formulas which are true in that world. Some of these formulas do not contain the symbol K, and their truth-values at a world will depend only on the truth-values of the atomic formulas at that world. In the case of a formula of the form Kp, its truth-value at a world depends (recursively) on the truth-value of the embedded formula p at worlds accessible from (i.e., epistemically compatible with) the given world. That is, the agent in a situation w is viewed as knowing fact p if p is true, no matter which of the alternative states of affairs compatible with his knowledge (in w) turns out to actually be the case.

3.2 Combined Logics of Knowledge and Action

Work has been done in computer science and AI to apply modal logics of a very similar kind to the domain of actions as well. This is of significance to AI and robotics because computers and robots do not just passively sense the world but react to it as well. For instance, *dynamic logic* (DL) has been developed to investigate how one might reason about the actions of programs [16,6]. One can think of DL as a logic that is just like EL except that, instead of a K operator, there is a family of operators $[\alpha]$, one for each α in some set A of actions. (In full dynamic logic, the actions are closed under program-constructing operations; here we just concern ourselves with primitive actions.)

Proof-theoretically, DL is also similar to EL. In full DL there are axioms describing the interactions of the program-constructing operators with the other connectives, but for our limited language the only special DL axiom needed is the one corresponding to the consequential closure axiom in EL:

$$[\alpha](\phi \rightarrow \psi) \rightarrow ([\alpha]\phi \rightarrow [\alpha]\psi).$$

This axiom distributes $[\alpha]$ over $\rightarrow$ effectively allowing modus ponens to work within the scope of $[\alpha]$. The only additional rule of inference needed is ***dynamic necessitation:*** from p derive $[\alpha]p$.

The semantics of DL has an accessibility relation for each action α. These relations might be called *dynamic accessibility relations* and they model the state transformation that is induced on the world by the actions. Indeed, the possible worlds themselves should now be thought of as world states, that is, as instantaneous states of affairs. (Our original epistemic logic was neutral with regard to time.) The formula $[\alpha]p$ would be true in a world state w if p were true in every world state that could result from the agent's performing action α in w.

Moore [14] has used a logic that combines elements of epistemic and dynamic logic to characterize the interaction between actions performed by an agent on the world and knowledge the agent has of the world.[1] In Moore's logic, both the K operator and the dynamic operators $[\alpha]$ are present (albeit in a different syntactic form from that presented here), and one can freely iterate their application to express facts about the agent's knowledge of the effects of actions on the world, or of the effects of actions on knowledge. In that logic, one can write down the equivalent of formulas like $K[\alpha]p$ (the agent knows that, after it has done α, p will be true) or $[\beta]Kq$ (after doing β, the agent will know that q is true.)

[1] There is a historical connection between dynamic logic and the possible-worlds logics of knowledge; it was Moore who brought to Pratt's attention the possibility of basing a model of program actions on modal logic.

4 Linking the Formalisms to Reality

Having a model-theoretic semantics for a combined epistemic-dynamic logic (EDL), while theoretically valuable, is not in itself sufficient for understanding what it means for computers to know and act. We need to know how the actual world corresponds to the abstract model.

For the *dynamic* part of the logic, i.e., that part concerned with actions as state transitions, it is straightforward to attach real-world interpretations to entities in the formal model. At any point in time, the computer-world pair is in a certain state. Events occnr (input events, output events, and internal operations in the computer), and the state of the world (including the computer) is transformed. That is the conventional meaning of the dynamic accessibility relations associated with the actions.

In the case of the epistemic accessibility relation, the situation is more problematic. How can we attach a real-world computational interpretation to the K operator? What does it *really* mean for a machine to know p? Under what real-world conditions would we say that a machine satisfies the axioms of EL? There are two approaches to this question that will be described. The first (the classical AI approach) is based on interpreted symbolic structures where the second (the situated-automata approach) takes the view that knowledge has to do with objective correlations of state between the machine and the world.

4.1 The Interpreted-Symbolic-Structures Approach

The key idea of the interpreted-symbolic-structures (ISS) approach is to view the state of the machine as encoding symbolic data objects—symbolic in the sense that the designer has in mind some particular interpretation that maps pieces of the internal state into pieces of the world. This interpretation can be used to assign truth conditions to the state, that is, to stipulate conditions under which the data structures express true assertions about the world.

The actual choice of data objects may vary, but knowledge representation research in AI has tended to focus on certain classes of structures. One of the most common is *labeled graphs* (e.g. ·semantic nets, so-called *is-a* hierarchies, and other taxonomic hierarchies), which can be thought of as collections of nodes and arcs labeled with contentful symbolic tokens, the content deriving from the intended interpretation the designer associates with these tokens. Another example is *frames*, which can be thought of as symbolic record structures. The data structures used to encode formulas of predicate logic are perhaps the canonical example of AI representational structures.

These data structures by themselves could hardly be called knowledge-representation structures, were it not for the interpretation function that the designer has in mind. It is this interpretation function that maps the symbolic elements into their truth conditional interpretations in the world and allows content to be assigned to the data structures. This is easiest to see in the case of the logical-formula structures, since the assignment of content

ordinarily follows the methods for recursive specification of truth conditions that has been so thoroughly explored in mathematical logic. However, other types of symbolic structures can be similarly interpreted, either by first translating them into more standard logical languages or by assigning truth-conditional semantics directly [7].

Under the ISS approach, it is only when one has an interpretation function in mind that it makes sense to ask the question, "What information about the world is encoded in the state of the machine?"—and even then there are at least two distinct answers, neither of which is entirely satisfactory. These answers are usually labelled the *semantic* and the *syntactic* (or *implicit* and *explicit*) [10,12], but both postulate a knowledge base of explicit interpreted symbolic structures, so in a sense they are both "syntactic." Let us call the knowledge base associated with the machine in state w, $kbase(w)$; for concreteness, the knowledge base can be thought of as the conjunction of a finite set of formulas.

The semantic approach stipulates that the agent knows proposition ϕ if ϕ is semantically entailed by the contents of its knowledge base. Formally,

$$w \models K\phi \Leftrightarrow (\forall w')\,(\,w' \models kbase(w) \Rightarrow w' \models \phi).$$

This amounts to defining the epistemic accessibility relation as follows:

$$\kappa(w, w') \Leftrightarrow w' \models kbase(w).$$

This interpretation of K has the property of satisfying the consequential closure axiom, as the reader may verify. However, it leaves us in a puzzling situation: If we assume that only the explicit contents of the knowledge base can affect action, how can the machine's actions be made contingent on *implicit* consequences of its knowledge? If they cannot, then even under the semantic conception of knowledge, every known proposition ***that actually controls behavior*** must be made syntactically explicit—that is, it must be derived inferentially. (This explains the traditional emphasis on mechanical inference techniques in AI.)

The advocates of the syntactic interpretation of K take this need for syntactic explicitness one step further—but at the expense of consequential closure. They argue that consequential closure is merely an idealization in the first place; how can a resource-limited agent possibly *know* all the consequences of its knowledge, *know* all the tautologies, and so forth [10]? They propose a syntactic definition of knowledge in place of the semantic approach. According to the syntactic definition, an agent is said to know ϕ in state w if (part of) $kbase(w)$ explicitly encodes the formula ϕ. Here at least there is a match between the proper conditioning of actions—an explicit test is performed on the structure in question—and the definition that stipulates which facts are known.

The main point to be noted regarding the ISS approach, whether under the semantic or the syntactic conception, is the following: the ascription of knowledge requires viewing the machine's state as structured in a certain way; knowledge is not an objective property of the way the machine is embedded in the world. Stated another way, the ascription of knowledge depends upon the attitude of a designer; the same machine in the same state in

the same environment will possess different knowledge under a different interpretation of its structures. We take this to be an undesirable state of affairs—one that we attempt to remedy by applying the theory of knowledge advanced in the next section.

4.2 The Situated-Automata Approach

The alternative we suggest is to ground the notion of knowledge in objective correlations between machine states and world states. An example of this approach in a limited domain is the vision work done by Horn, Marr, and others [2,13], where objective physical constraints are used to establish logical relationships between predicates on retinal images and predicates on object surfaces (the symbol $\rightleftharpoons$ is a schematic symbol taking as substitution instances the connectives *if, only if,* and *iff*):

$$\phi(\text{RETINA}) \rightleftharpoons \psi(\text{ENVIRONMENT})$$

It will be shown that the correlational approach can be extended beyond low-level-stimulus presentations and can be used to give a complete information assignment for arbitrary machine states (viewed as equivalence classes of stimulus sequences). This assignment will depend solely on the constraints that govern the coupling of the machine to its environment.

We begin with some elementary definitions from automata theory. Let $M = (S, \Sigma, A, \delta, \lambda, s_0)$ be a (deterministic) automaton, where

- S is a (finite or infinite) set of *states.*
- Σ is a set of *inputs (stimuli).*
- A is a set of *outputs (actions).*
- $\delta : S \times \Sigma \rightarrow S$ is a *next-state function.*
- $\lambda : S \rightarrow A$ is an *output function.*
- $s_0 \in S$ is an *initial state.*

We assume the machine M is connected to an environment that can be in one of some (probably very large) number of states; this environment is generating the inputs for M and responding to M's outputs. (The machine-environment configuration is pictured in Figure 1.)

Let Φ be the complete lattice of instantaneous *world conditions* ordered by $\sqsubseteq$ and closed under $\wedge$ and $\vee$ (the usual operations of lattice meet and join). If ϕ and ϕ' are elements of Φ, then $\phi \sqsubseteq \phi'$ is intuitively interpreted to mean that condition ϕ is less general than (i.e. entails) condition ϕ'. Thinking of the elements $\phi, \phi', \ldots$ as sets of possible world states, the structure can be viewed as a special kind of lattice, namely the Boolean algebra of subsets of possible worlds, with $\wedge$ and $\vee$ corresponding to set intersection and union.

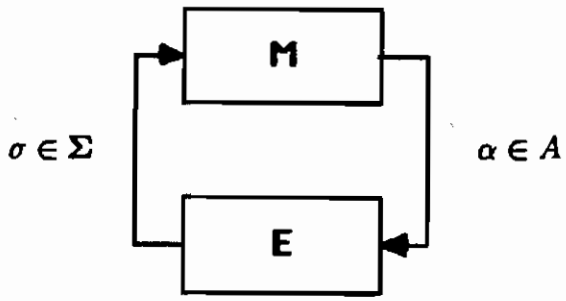

Figure 1: An automaton coupled to its environment

We sometimes write $\phi(w)$ to indicate that condition ϕ holds of the world-state w. Also, we assume a distinguished world-condition $\phi_0 \in \Phi$, which intuitively corresponds to the strongest condition the world must satisfy when the machine is started in condition s_0.

In addition to expessing relations (e.g. of entailment) among instantaneous world conditions, we must also express how elementary input events of the machine are related to changes in world conditions. For this purpose we associate with every $\sigma \in \Sigma$ a function from Φ into itself: the *strongest postcondition* function. Intuitively, ϕ/σ, read ϕ *over* σ (cf. [17]), is the most specific world condition that can be guaranteed to hold at time t' given that ϕ holds at t and that the elementary input stimulus σ is sensed over the time interval $[t, t']$.

The operator / is easily extended to *sequences* of input stimuli. Let Σ^* be the set of finite sequences over Σ, Λ the null sequence, $\bar{\sigma}$ an element of Σ^*, and ; concatenation. Define

$$\begin{aligned} \phi/\Lambda &= \phi \\ \phi/(\sigma;\bar{\sigma}) &= (\phi/\sigma)/\bar{\sigma} \end{aligned} \tag{1}$$

The condition $\phi_0/\bar{\sigma}$ corresponds to what the state of the world must be, given that the automaton has been started in its initial state (with the world satisfying ϕ_0) and has experienced the sequence of events $\bar{\sigma}$.

Since the automaton's knowledge of the current state of the environment is mediated only through inputs, it is entirely possible (even likely) that certain conditions $\phi \in \Phi$ will remain entirely inaccessible to it in the sense of there being no sequence of inputs that can discriminate situations in which ϕ holds from those in which it does not. We call the conditions which the automaton could, in principle, discriminate the *knowable conditions* and define these to be

$$\hat{\Phi} = \{\phi_0/\bar{\sigma} \mid \bar{\sigma} \in \Sigma^*\}$$

A given world condition ϕ is discriminated by arranging for the machine to recognize sets of input sequences $\bar{\sigma}$ such that $\phi_0/\bar{\sigma}$ entails ϕ. The states of automata can be viewed as grouping input histories into equivalence classes and the coarseness or fineness of the equivalence relation determines the degree of blurring of information about the world. Even an automaton with enough internal states to encode each distinct input history in perfect detail would only be able to discriminate elements of $\hat{\Phi}$, which, in general, is a strict subset of Φ. An automaton with less than perfect memory will induce coarser equivalence classes on input sequences, and, in general, these will correspond to "collapsings" (disjunctions) of elements of $\hat{\Phi}$. These collapsings are determined by the state-transition rules of the automaton.

This situation is described formally as follows. For each state s of the automaton there is an associated language (set of input sequences) L_s, the equivalence class of input-stimulus sequences that leave the machine in state s when it is started from state s_0. The formal definition of L_s is gotten by first extending δ to sequences in the usual way:

$$\begin{aligned} \bar{\delta}(s, \Lambda) &= s \\ \bar{\delta}(s, \bar{\sigma}; \sigma) &= \delta(\bar{\delta}(s, \bar{\sigma}), \sigma) \end{aligned} \tag{2}$$

and defining

$$L_s = \{\bar{\sigma} \in \Sigma^* \mid \bar{\delta}(s_0, \bar{\sigma}) = s\}.$$

We can also define a pair of related concepts: $L(\phi)$, the language of condition ϕ, and $\phi(L)$, the condition of language L.

$$\begin{aligned} L(\phi) &= \{\bar{\sigma} \in \Sigma^* \mid \phi_0/\bar{\sigma} \sqsubseteq \phi\} \\ \phi(L) &= \bigvee_{\bar{\sigma} \in L} (\phi_0/\bar{\sigma}) \end{aligned} \tag{3}$$

Intuitively, $L(\phi)$ is the set of input sequences that pick out condition ϕ in the world. $\phi(L)$, on the other hand, is the strongest condition guaranteed to hold after the occurence of any input sequence in L. This latter notion can be used to define the *information content* of a state directly:

$$\mathit{info}(s) = \phi(L_s).$$

This model of information is related in a straightforward way to the possible-world models of epistemic logic presented in section 3.1. All that is needed is to specify what the epistemic accessibility relation κ amounts to in the current setting. For any world-state w let $\mathit{state}(w)$ denote the state of the automaton in world w. Defining

$$\kappa(w, w') \Leftrightarrow \mathit{state}(w) = \mathit{state}(w'),$$

it follows that

$$\kappa(w, w') \Leftrightarrow \mathit{info}(\mathit{state}(w))(w').$$

In words, $\kappa(w, w')$ will hold whenever the automaton is in the same state in w as it is in w', or equivalently w' satisfies the condition corresponding to the information content of

the machine's state in world w. Since under this definition κ is an equivalence relation, the axioms of modal system S5 are satisfied. In particular, as in the "semantic" construal of the ISS approach, deductive closure is guaranteed automatically. It is trivial to verify the validity of

$$K(\phi \rightarrow \psi) \rightarrow (K\phi \rightarrow K\psi).$$

Unlike the ISS approach, however, no assumption is made regarding any data structure encodings in the state of the automaton. Furthermore, there is no presumption that actions may be made contingent on knowledge of arbitrary propositions. In fact, actions are contingent on actual machine states s (this contingency being expressed by the output function $\lambda(s)$.) The information content of s is epiphenomenal (though, as observed below, it serves a useful function for the designer.) Furthermore, not all logical conditions definable in the designer's metalanguage have equal status as potential knowledge states—only those conditions ϕ such that $L(\phi)$ are recognizable languages fall into this category. (In some ways, this distinction is similar to the one that exists between functions and *computable* functions.)

5 Discussion

Logical assertions have an important role to play in the correlational approach, but this role differs markedly from that played by the logical-formula data structures of the classical knowledge-representation approach. In the old approach, the machine is viewed as manipulating data structures that encode logical assertions as linguistic objects. In the new approach, logical assertions are not part of the machine's knowledge base, nor are they formally manipulated by the machine in any way. Instead, these assertions are framed in the metalanguage of the designer, who uses them only to express (to himself!) various background assumptions he is making and to characterize (again for himself) the information content of the states of the machine he is designing. His purpose is to comprehend the emerging design and verify that the machine will behave as desired. In particular, there is no need to encode the background constraints themselves explicitly in the machine's state, since they comprise a permanent description of how the automaton is coupled to its environment and are themselves invariant under all state changes. Indeed, there is no need to regard the states as structured into "symbolic" data at all.

Although this paper contains only the beginning of a fully developed theory of situated automata, it appears that such a theory could make a useful technical contribution to the ongoing debate in AI and philosophy of mind over the role of interpreted representations ("language of thought") in the semanticity or intensionality of mental state [4,3,1]. The present account indicates how propositional content can be assigned systematically to arbitrary computational states that are not prestructured as interpretable linguistic entities, and thus it serves as at least prima facie evidence against the need for a language of thought in order to achieve full semanticity.

Interesting properties emerge when one applies the informational definitions above to complex machines constructed of interconnected parts. For instance, one can analyze "inference" within a machine as internal information flow. Consider a machine with two internal subparts, each possessing a local state. If at time t part A carries information p (in the sense described above) and part B carries $p \rightarrow q$, then certainly the entire machine is in a state it could be in only if the world satisifed both p and q. Thus, according to the definitions, the entire machine knows $p \wedge q$—although neither of its subparts may. Now let us assume that parts A and B sense each other's state and change their own states accordingly; information will have crossed the boundary so that not only will the whole machine be in a state where both p and q are known, but each of the subparts will know these facts as well. The detailed analysis of the dynamic loci information is important in applications such as robotics, since it is not sufficient that the robot as a whole know what to do; the end effector must also know. The information must flow to subparts of the machine.

Some of the ideas presented here are already being applied to robot research currently under way in the Artificial Intelligence Center at SRI. In the spirit of the original Shakey project of the late sixties and early seventies [15], we are constructing an intelligent mobile robot on which several experiments in perception, reasoning, and planning will be carried out. Some of these experiments will be based on interpreted symbolic structures, others on the emerging theory of situated automata. The latter appears to offer interesting possibilities for exploring intelligent real-time operation, since it integrates knowledge representation, perception, and planning in a single theoretical framework that does not rely crucially on expensive runtime deductive inference. We are currently extending the formal techniques described here and implementing software development tools that facilitate the construction of complex, hierarchically structured programs with knowledge properties that can be derived compositionally from those of its parts.

6 Acknowledgments

I have profited immensely from discussions with Fernando Pereira, Leslie Pack Kaelbling, Nils Nilsson, Robert Moore, Brian Smith, Jon Barwise and John Perry and from comments on an earlier draft by Jerry Hobbs, David Israel, and Mike Georgeff. Any opinions and errors are, of course, ultimately my own.

7 REFERENCES

[1] Barwise, Jon and John Perry. *Situations and Attitudes.* MIT Press. Cambridge, Massachusetts, 1983.

[2] Brady, J. Michael (ed.). *Computer Vision.* North Holland Publishing Company. Amsterdam, The Netherlands, 1981.

[3] Dennett, Daniel C. *Brainstorms.* Bradford Books. Cambridge, Massachusetts, 1978.

[4] Fodor, Jerry A. *The Language of Thought.* Thomas Y. Crowell Company. New York, 1975.

[5] Halpern, Joseph and Y.O. Moses. "Knowledge and common knowledge in a distributed environment." Proceedings of the 3rd ACM Conference on Principles of Distributed Computing, 1984, pp. 50-61; a revised version appears as IBM RJ 4421, 1984.

[6] Harel, David. *First Order Dynamic Logic.* Lecture Notes in Computer Science, Vol. 68. Springer-Verlag, 1978.

[7] Patrick Hayes. "In Defence of Logic." Proceedings of the Seventh International Joint Conference on Artficial Intelligence, Vancouver, B.C., 24-28 August, 1981, pp. 559-565.

[8] Hintikka, J. *Knowledge and Belief.* Cornell University Press, Ithaca, 1962.

[9] Hughes, G. E. and M. J. Cresswell. *An Introduction to Modal Logic.* Methuen and Co. Ltd., London, 1968.

[10] Konolige, Kurt. *A Deduction Model of Belief and its Logics.* Technical Note No. 326, Artificial Intelligence Center, SRI International, Menlo Park, CA, August, 1984.

[11] Kripke, Saul. "Semantical Analysis of Modal Logic." *Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik* **9**, 1963, pp. 67-96.

[12] Levesque, Hector J. "A Logic of Implicit and Explicit Belief." Proceedings of the National Conference on Artificial Intelligence, 1984, pp. 198-202.

[13] Marr, David. *Vision.* W. H. Freeman and Company. San Francisco, California, 1982.

[14] Moore, Robert C. "A Formal Theory of Knowledge and Action." In *Formal Theories of the Commonsense World*, Jerry R. Hobbs and Robert C. Moore (eds.), Ablex Publishing Company, Norwood, New Jersey, 1985.

[15] Nilsson, Nils J., "Shakey the Robot," Technical Note No. 323, Artificial Intelligence Center, SRI International, Menlo Park, CA, April 1984.

[16] Pratt, Vaughan R. "Semantical Considerations on Floyd-Hoare Logic." Proceedings of the 17th IEEE SYmposium on Foundations of Computer Science, October 1976, pp. 109-121.

[17] Rosenschein, Stanley J. "Plan Synthesis: A Logical Perspective." Proceedings of the Seventh International Joint Conference on Artficial Intelligence, Vancouver, B.C., 24-28 August, 1981, pp. 331-337.