SESSION 2

The Symbol Level and the Knowledge Level

Allen Newell's Response

Let me try and talk about Dan Dennett's remarks just a little bit.

Several people made remarks about the Julie Christie example. The criticisms seemed to me to be a little bit odd because to say that one has access to something does not mean that one has access to *all* of that thing; having some information about Julie Christie certainly doesn't give one complete access to Julie Christie. That is what polite society is all about.

I can't remember whether Dan Dennett raised the following issue or if it came up later. At each level in the descriptive hierarchy there are simple systems which are not knowledge level systems. That is, as you go up the hierarchy from the circuit level, to the logic level, to the register transfer level, to the symbol level, to the knowledge level—at each level there are systems at the current level which are not symbol systems. There are lots of systems that do not realize a system at the next level up. There are lots of systems at the current level which are not symbol systems. For example, there are lots of ways of putting together a circuit which would not allow you to make a symbol system out of it. There are lots of ways of putting together a symbol system which would not allow you to make a knowledge level system out of it. In one sense what the knowledge level is assuming is that the symbol system is organized to realize a problem of rational goal-oriented behavior with respect to the data structures and the processes. And if the system is not organized so that it does that, then in fact there isn't any way to characterize that behavior in any reasonable sense in terms of knowledge. The idea is that each of these levels is obviously in some sense an approximation. Although one thinks of each of these levels as providing an alternative description of the same physical system, there is a whole lot of behavior which is describable at one level but not at another. In particular it seems very clear that the knowledge level is, as I said in the paper, a radical approximation. When someone says the word "approximation" the first thing you think is, well it's missing in fit a little bit. But the right statement, I think, about the knowledge level relative to the symbol level is that it is an approximation in *scope* and not just in the sense that it's sort of close but always missing a little bit. That is, there are many kinds of behavior that people engage in for which it's simply wrong to describe them as working from knowledge and achieving goals.

There are two obvious purposes for which one wants a knowledge level description. One of them is just for prediction. That is, if you have a system and you're outside it and you don't know anything about what's inside it, then the knowledge level may be sort of a minimal characterization in which I can still get some predictions. And if in fact the system turns out to be internally organized that way, then I can get predictions by understanding the environment that it's in and by only having to posit knowledge and goals and no other internal mechanism. So the distal prediction problem—which shows up in folk psychology—arises when I don't know what's inside, and this is a kind of a model from the outside. The other purpose of knowledge level descriptions shows up in engineering very strongly, when I wish to specify what a system will do before I build the structure to do it. (I call this the design problem.) Consequently, I have to find some way to talk about the system before I go down one level and design that. The knowledge level gets used here not because I'm distal in the sense that I don't know what it is made of, but because I'm going to specify the knowledge level description, and then I'm going to try to build a system that realizes it. So the knowledge level shows up strongly in design as well as in prediction, and there may be other areas as well.

Now one last remark about Dan Dennett's comments. I get the feeling that one of the things that the philosophers want most strongly is a characterization of the intentional stance—a characterization that in my terms really makes a separation between the knowledge level and the rest of the system. I think that in AI the use of the knowledge level is exactly analogous to the rise of all these other levels, which is to say that it has no special status. All the problems about the relationship of the symbol level to the knowledge level apply to each of these pairs of levels, and there is no special transition at the knowledge level. It is true that the knowledge level is in some respects peculiarly abstract relative to the other levels. That rests in a superficial way on the fact that if I look at the material that is being processed, if I look at the data structures in the system, then in some sense there is a direct connection as we go up until we reach the knowledge level. The register transfer level is just a collection of bits at the next level, and the bits are just a voltage level, so that the material substance seems to have a kind of material form all the way up. When I get to the knowledge level, the knowledge really is to be ascribed to the system because of the entire processing of the system as it tries to solve problems and to do rational processing. And that is not to be identified with just a material structure, so that there is no material correspondence from what is knowledge at the top level and what are data structures down lower. This reflects a remark that showed up in Dan's comments as well when he wanted to talk about this urge to try to get something a little more concrete, a little more reducible for knowledge. That seems to be a part of our whole history of trying to deal with it. Here one is really going the whole way and making it quite abstract, but I don't think it's any different.

With respect to Brian, I'm distressed that he found so much transparently wrong with my approach because I think of what I'm doing as simply reporting what is obvious to people in AI—I think of myself as a reporter of that scene. Since Brian is a fellow traveller, I assume that he understands AI about the same way I do and as much as I do. Either it's the case that AI is totally irrelevant to the enterprise that Brian really has in mind or in some sense I'm a really bad reporter, because he must be seeing things radically different that I do. The other possibility is that what has been happening in AI over the last 25 years is irrelevant to the enterprise in cognitive science that we're now discussing. Let me try and respond to some of his criticisms.

I'll begin with the question, "Where does the aboutness come from?" Why is it that something is "about" something? My claim is that this happens in two stages. The first stage is that there are symbols which lead to internal structures. I don't think this is obscure, and it is important in understanding where the aboutness comes from.

These symbols lead to certain expressions so that the access relationships are actually embedded in the architecture that allows you to get to them. Then one builds the symbols for representing the external world by building further programs and data structures in which the data structures contain knowledge about things in the outside world. So you then build up further symbols which access things that you can think of as knowledge about something—knowledge about Julie Christie for instance. If you want to ask why a certain symbol says something about Julie Christie, you have to ask why the symbolic expression that contains the symbol says something about Julie Christie. And the answer may be that it says something about Julie Christie because of processes that put it together which may themselves have knowledge about Julie Christie. So you get referred, once again, to some other knowledge. Ultimately it may turn out to depend upon history, it may depend on some point in the history of the system when it came in contact with something in the world which provided it with that knowledge. The system gets knowledge of Julie Christie only because of that historic sequence. If a program has knowledge, that's usually how it gets to have that knowledge. Of course there is another way it could have knowledge, namely from whatever puts the system together. In the case of people, or so most of us believe, it can come through evolution which takes parts of the universe and puts them together in order to form an organism of a given type. So there is embedded knowledge that doesn't come in from outside, it comes in from the construction of the system. This view is neutral with respect to the question of what's innate and what comes in through the senses. Knowledge clearly comes both ways.

Now it is conceivable, as I think some of you will argue, that this is an inadequate kind of mechanism. The claim that is being made is that it will in fact turn out to be adequate. The hypothesis is that we will find that through mechanisms that have symbol systems in them we can put together systems which do in fact have arbitrarily large amounts of knowledge about the world, and which behave according to that knowledge in ways that are in some respects approximated by rational behavior, while in other respects they reveal the operation of their processing. That is in some sense a discovery of computer science, not just of artificial intelligence. In computer science we can develop this kind of organization into a working system.

Let's talk about designation for a moment. The general reaction I've gotten to my definition of designation is that people notice that I am talking about a causal mechanism and so they invent some physical counterexample like a stomach or a car or something like this, and say, "That's obviously not a symbol—so there. That takes care of that definition." I have a complaint about that. What the critics ignore is that in my description there's a proposal for how designation *works*. And sure that may be wrong, but if it is, then what's really going on is that we just haven't got the right description of what's really going on down there in a computer.

Instead of just complaining about my definition, what Brian should have done is go back to what he knows about artificial intelligence and try and produce a correct version. In other words, he should have formulated the right description of what exactly is providing the action in all the AI programs. Now of course it may be that he feels that the description of what is going on is alright, and that it's just the character of the definition that's wrong. However, it didn't come over that way; it came over that he just wanted to say that my characterization was inadequate to describe what was going on.

Let me try and get at the problem Brian raises by going back and looking at the phenomenon that we have to deal with. What's true in all AI systems? What's true is that there is something called a "process." And what we see is that in parts of that process a collection of bits show up. Now that process almost always works as follows. It uses that collection of bits to open up a path, so it goes outside what was actually given to it to another word or another representation. The definition of the process is distal in the sense that what the process does depends on this other representation. To describe it, you don't describe it in terms of what it does with bits. Although there is such a description, the description involves going to the memory address register and putting those bits in the memory address register and then waiting until the system has automatically brought in the contents of this word. Then you can do something with these words; you can append a description of what that access path is like. The way you describe this thing varies: You can talk about it adding, you can talk about it as loading, you can talk about it as carrying out the LISP functions CAR, CDR, CONS, and so on. They all have that character. The other thing that you see happen is occasionally you fire up some circuit that may turn on the TV camera or close down the machine or do something externally. In all these cases you cannot define the process in terms of what the machine does with the bits, but only with respect to the arithmetic function of the circuits, or with respect to the distal action of the machine. That's where the action comes from in all these programs. So one ends up saying that what one has is a symbol which refers or designates (I don't much care about the word; I think all the words are inappropriate—maybe we should just use the word "access"). That is the relationship that is going to be critical for building symbol systems. Now these things—these bit patterns—are symbols only with respect to this process. If they were to appear in some other process, it might not behave this way. So consequently, the way I gave the definition—and I've never seen anybody actually respond to this—was to emphasize rather strongly that we talk about designation of a symbol with respect to a given process.

Now I'll raise a flag about "degeneracy." One is ill-advised in thinking about these systems to immediately go to the most degenerate form of system that you can think of (say, digestion) and assert that because the degenerate system satisfies the particular conditions of the definition, the definition is useless. There are many times when you may want a definition for your own analytical purposes, when the degenerate cases don't look very much like what you are talking about. It happens all the time, and yet such definitions are still scientifically useful most of the time. The important point about defining designation as a relationship, as I did, is not that it should by itself pick out exactly the class of symbol systems and exclude all the degenerate cases. The character of designation arises when one embeds the symbols in a whole set of processes. It is this whole set that finally gives you a symbol system, not just one symbol in relation to some arbitrary process.

If you take a particular set of processes, what we call a symbol system will have all of the following properties: It is something which can be programmed, can acquire new processes, can interpret its own behavior, and so forth. It's a system which has the property that you can build a version of it that will realize a knowledge level in the sense that it will carry out approximations of rational behavior which require it to do a lot of different kinds of things. And if I just pick any arbitrary mechanism, it's not at all clear that I'll be able to do that. The collection of processes which allows you to do this is quite simple. They are, in fact, spelled out in the paper; they are just a variant of the kind of processes one would spell out for any universal computing machine. The whole set of properties is necessary in order to have a symbol system. This solves the problem that Brian raised—not perfectly, but well enough.

There's another thing going on here. In my paper, things are not very formal. Yes, the general characterization of what a symbol system is is reasonably good for English prose (at least I hope it is), but it certainly is not a formalization of it. In fact, the particular description of what the designation relationship is is not formal either. Why not? One may suppose that maybe I'm simply not a formalist, but there's something more important going on. But the fact is that there exists no real structural characterization of universal systems. The situation seems to me to be the following: We can define classes of machines; we can define Turing machines; we can define random access machines; we can define an infinite number of different language machines (LISP machines and so forth); I can define particular classes of machines. Within each class of machines—assuming they have certain appropriate properties (this is true for all these programming languages and computers and architectures and so on)—there is a subclass which are universal machines. The definition of a universal machine is that, under a suitable mapping, it will mimic any of the machines within that class. We know a little about such universal machines. We know that you can build mappings back and forth between each of these classes. That's why, in fact, there turn out not to be an indefinite number of notions because if you look carefully each such machine provides you with essentially maximum flexibility in the kinds of functions it can realize. Such universality really reflects the capability that you want in a system that has the capacity of producing rational behavior, viz., the ability to produce arbitrary input–output functions for it's own use with respect to its detectors. As I said in my paper, universal machines and symbol systems are the same thing, even though they are not defined from the same point of view. And so what would be useful to have would be a formalization of this class and of its structural features. I don't know of any, and I have no idea whether there are any structural characterizations that really hold. The one thing you can do is identify all the components that show up in the definition of a symbol system. Because the mappings must exist, if I have symbols defined in any one of these, I can go through the inverse mapping and find out what are playing the roles of symbols and symbol expressions, of the memory, of the processes, and so on, in the universal machine. I can find that out for any of these machines; so in one sense, these have the structure of a symbol system. But since each of these systems is described in its own structural terms, that still doesn't give you a nice structural characterization. So when it comes right down to characterizing these systems, one tends to fall back on just functional description, which is what showed up in my paper.

Finally let's talk about the attribution of knowledge. One of Brian's points was what he called the "great gulf" or the "great separation." The notion is that it is only for me, the observer, that all these problems of how a system can have knowledge and how it can have aboutness and so forth arise. I am merely attributing such properties to the machine. The reason I can do this is because I can lay any interpretation on that system that I want. But now it seems to me that this is a form of solipsism, in the sense that no matter what we do it will always be we who, in fact, are interpreting what is going on. It seems to me that if one takes that route, which is to always ascribe this to the interpretation of the observer, then one is really going to have to face what might be called the problem of coincidence. As the systems that you are interpreting get more and more intelligent and do more and more things, it gets harder and harder for you to say that it is just because I choose to interpret it this way that they appear to understand what they're doing. In the extreme, if such machines were capable of doing good science, and of writing papers, you would have even more trouble insisting that it was just your interpretation. And if you did insist on this, these machines might grab you by the lapels and *insist* that they understand and that they refer. And if you persisted, they might even sue you for depriving them of their personhood! I'm just trying to illustrate the point about coincidence. It seems to me that that's where the attribution path could lead.

One last comment about folk psychology and the knowledge level. We need to separate different levels: knowledge level, symbol level, architecture level, and so forth. If you look at what folk psychology is, it is a great mixture of all of these things—it is not just the knowledge itself. People use all sorts of ideas about other people in folk psychology, including such things as that people have a limited short-term memory. Folk psychology is a great mish-mash that wanders all over these levels and even considers things from lower levels, such as the effects of disease or of overheating on the person, and so forth. So one does not want to make the assertion that folk psychology simply comes from the knowledge level.

General Discussion of Newell's Paper

Friedman: I was struck yesterday by how everything was going along very smoothly and scientifically until the word "designation" was used. In the morning session there seemed to be something everyone agreed about, namely something called the computational model of the mind. There was a universal overarching hypothesis and there were also evidential considerations for this as everybody seemed to agree. Fodor laid out a very general scientific research program and then the two respondents brought to bear actual empirical facts, one about real people, one about machines. But as soon as the topic of designation came up all this nice agreement and all this seeming contact with empirical data went away. So I got the feeling that there was an underlying philosophical tension about the relationship between syntax (which is, of course, intimately connected with the notion of computation) on the one hand and semantics on the other. And I thought it might be helpful to lay out a little history of how philosophers viewed the relationship between these two, and that this might explain some of the tension, as I saw it, in the discussion.

Now this idea, that something called logical syntax or a computational model is important, arose from the development of modern logic, and then in the '20s and early '30s, from the work of various philosophers, especially Wittgenstein in the *Tractatus* (1922) and Carnap in *The Logical Structure of the World* and in a book called *The Logical Syntax of Language* (1934). This last work is probably the most explicit version of the view. They said modern logic has given us this powerful new tool we can call logical syntax. And then they took the revolutionary step of saying that using that and that alone we can give a complete picture of the nature of language, of the mind, and of what thought is all about—just using the mathematical and syntactic structure that modern logic has discovered. I think one thing that inspired them is what we can call the infinite complexity of modern logic as compared to traditional logic. Here I don't just have in mind the idea that people always have in mind when they say that in logic we can build up an infinite number of different sentences from a finite number of parts—that there's this compositional character. That's a trivial property which is shared by propositional calculus and by monadic or Aristotelian logic with one-place predicates, where you don't get the characteristic kind of complexity that I think people saw in modern logic. That kind of complexity arises when you go beyond monadic logic—which is something Russell used to emphasize all the time—to what he calls relations or polyadic predicates. Because, on the one hand, using relations and not just one-place predicates or monadic or syllogistic techniques, you can actually assert that there are an infinite number of things. With monadic or syllogistic logic, any consistent sentence or set of sentences you write down has finite models, while with a two place relation (say "bigger than") you can say "For all x there is something bigger than x," and that guarantees an infinity. So, as Russell said, we can grasp an infinity by means of signs.

But even more interesting and important than this was the idea that you could construct an infinite number of distinct logical forms of concepts or logical forms of notions. If you think of it, what logical properties do *sets* have? Well, basically cardinality. If you talk about the logical form of something as simple as a set, the only difference is in terms of cardinality. But as soon as you get a relation, you have an infinite number of forms that you can pick out by purely formal or mathematical properties, like that something is symmetric and transitive. You can build orderings, linear orderings, then you can have orderings of various numbers of dimensions, and then, seemingly using just logical notion, you can, as it were, individuate an infinite number of different concepts purely by their logical features. This is the key idea in Carnap's *Aufbau* where one of the interesting technical moves in it is to assert that we have this general idea of a quality space, like sound space or color space, where concepts are supposed to be arrayed in these quality spaces and where we have to individuate them. How do we do it? Well, there's one and only one quality space which has five dimensions—and that's the visual field: three of color, two of space. And then we can, as it were, build up all the other qualities in terms of their formal relations to this one five-dimensional quality space.

There's a kind of revolutionary idea here about meaning: namely that just using these logical forms—and they are defined purely syntactically or formally—one can, as it were, represent anything you like. You don't need any such notion as reference to the external world or anything like meaning which correlates signs to other objects. If you just understand the logical relations and logical complexities of the signs and how they relate to each other, you can already represent anything you like. You have, as Wittgenstein put it, the right mathematical multiplicity to pick out any notion. By "represents" they don't mean "corresponds to something outside," they mean give you the logical consequences that you want from such a notion. I can tell you everything you want to know about the visual field, Carnap thinks, by its formal properties, starting with the ideas about dimensionality and so on. And so they basically want to destroy the idea of a word–world relation as anything important for thought. They replace it with this purely formal thing that reduces to logical relations between expressions. Furthermore, by looking at things that way you can make this notion, these concepts, thoughts, and so on, suitable for refined mathematical study—like in recursion theory, automata theory, and so on. It seems to me that a lot of the strength of the so-called computational view exploits this fact. You have this mathematically rich structure and the modern idea of a formal system, a polyadic formal system, where you have a kind of infinite complexity that you didn't have in previous views of logic. And then in these philosophers, and especially clearly in Carnap, it became the view that logical syntax is all there is: Nothing has to be added. You don't have to go find semantics. You don't have to correlate anything with anything else. Once we understand the logical syntax of our language, we have enough complexity to do anything we might want semantics to do. I think you see this tension in the opposition produced when Professor Newell used the word "designation." He gave a talk, which seemed to him to be reflecting just the common practice of the field, and then another practitioner of the field said well this just goes completely against our commonsense idea of reference. True it does go against the commonsense idea of reference, and it stems from this radical philosophical view that reference in the commonsense conception is not what you want, because we now have a better way to talk about the mind without any such notion. So I think, on the one hand, people gripped by this computational model are driven philosophically toward the view that really reference and semantics aren't part of the issue. But on the other hand, they sometimes want it, because they're talking about actual human beings and not just abstract formal systems. They have to somehow relate it to reality, and the only notions around seem to be ones like reference and meaning. But perhaps they are not the notions one wants to use if one is going to take seriously a computational view.

I just want to make another couple of remarks about what happened to the philosophical view after 1934—to both Carnap and Wittgenstein after they came out with a view like this. I think Carnap was seduced by Tarski into thinking that semantics was a mathematically respectable subject which could somehow generate other kinds of formal structures that would be just as interesting as the syntactic ones. In the course of this he wrote what might have been the most trivial book ever written, *Studies in Semantics*. I don't remember when that was written, somewhere around 1940. He just gives trivial disquotational definitions of everything. One has to see it to believe it. But he didn't just like semantics because he thought Tarski had made it mathematically respectable. He also thought that we need something to define the analytic–synthetic distinction. That is, within this whole formal system you could distinguish between some sentences that were true by virtue of their meaning and some that were true by virtue of your beliefs; so that if your beliefs changed, meanings could still stay the same. It was philosophically important that you be able to somehow make that kind of distinction. He recognized under Quine's influence that you couldn't do it syntactually and said let's try the next thing, let's try semantics. One thing I wonder, and I ask people in AI and cognitive psychology about, is whether they want that analytic–synthetic distinction. In some of the discussion we had this morning, Jerry Fodor was saying that you're committed to something that Quine has shown to be false, and you have this anti-holistic view. Hilary Putnam, in his talk, I think was saying that the problem arises because you're trying to make a distinction between knowledge and belief.

In pursuit of the analytic–synthetic distinction, and again goaded by Quine's criticism, Carnap turned to some half-hearted behaviorist psychology and wrote *Meaning and Necessity*. In an appendix to that book called "Belief and Synonymy in Natural Languages," he says that we can set up some simple behavioral tests for sameness of meaning that will be sufficient to give us the analytic–synthetic distinction. Then Quine in *Word and Object* gives a counterargument in his radical translation story. He shows that if you really take seriously this quasi-behavioristic account of meaning, what you get is nothing like what we understand as meaning. In particular, it is holistic, and it doesn't have an analytic–synthetic distinction in it. Again, I wonder if that criticism is relevant to people in psychology. It seems rather to be addressed to a particular philosophical program that the analytic–synthetic distinction is embedded in. I think that cognitive psychologists want to use the word "meaning" for another function.

I just want to make one more remark: What happened to Wittgenstein? Well of course he wrote a book called *Philosophical Investigations* which is about logic and psychology. One of the questions he was addressing was the third question which John Haugeland asked: What gives a formal system life? If you just write things down on a piece of paper, they just sit there. They don't actually perform any inferences or do anything. There has to be a causal mechanism or medium that this abstract formal system is operating in. Professor Haugeland asked about it, but in the *Investigations* Wittgenstein emphasizes an obvious answer: namely spoken language. People talk to each other, and you find a medium of inference and discussion and discourse, an intersubjective one, and it takes place over time. And that's where we'll find life. And of course this last idea is, I think, dramatically opposed to what people in AI would think, or it seems to be, when you say, "No, there's a medium in the head somewhere that's really driving this mechanism." Wittgenstein was suggesting, I think, that there's another medium, the medium of conversation and spoken language. And furthermore, I think there are various considerations and arguments which suggest that we need something like this to even have rule-governed behavior in the sense that even a formal system—let alone that semantics—requires.

Newell: I found that useful. Let me pick up on little pieces of it and go on to express what's probably a fairly crude view, but one that seems to me to characterize the problem in another way. At one stage of your discussion you said there are these formal systems and then there are people. And that is clearly one of the things that cognitive science wants to deal with. It wants to talk not just about formal systems, but also about people and about behavior and so forth. But there's also the computer, and the computer really behaves too. Whatever else it does, it behaves. It's not just a formal system—it's a hunk of stuff: It may have a formal system embedded in it in some way, but it lives in real time and real space, does things to the world, and reacts as the world does things to it. Now the ideas that are coming from AI are concerned with what it takes to make computers do interesting things. In building such systems we find ourselves faced with the same problems that are posed when we ask how people do these things, and thus such work can provide some answers. In the case of the compiler, I can check the internal structure. We have systems which can ultimately carry out intelligent behavior and which have all the appearance of making reference to the outside world. Yet we know that they are nothing but gears animated by a formal system, or rather a realization of a formal system. Do I then consider that that must be sufficient for reference? That's rather a crude way of saying it, but I think if you ask what is interesting about AI, one of the answers is found in the observation that that's what's happening.

Friedman: It seems to me that this continuing argument poses a challenge to that. If there's anything to the idea that what's essential to reasoning is language, then it doesn't make sense to talk of reasoning and referring unless we have a community of people talking to each other or unless we have more than one of these agents and they're talking to each other. There's a sense in which computers are not going to reason unless you design them so they can talk to each other. How can we design a computer which can, all by itself, think?

Haugeland: Yes, but that all started with an "if."

Friedman: Yes, that's why I say it's a challenge.

Jerry Fodor: It's a challenge if its antecedent is true; if its antecedent is false, it's sort of irrelevant.

Haugeland: I take it that it's just as likely or possible that when these fancy machines that Al Newell and his friends build, and will continue to build, get really good, that's going to be a challenge to Wittgenstein.

Newell: Yes, and one of the things one is always asking is whether one can convert some of these problems into real issues about where the practical programs of AI are going to run into trouble—not trouble in your interpretation of it, but into real trouble. If there really were some things which it was having real troubles with and it really did derive from some real difficulties with the kinds of structures that were being used, that would be great—we could then really have something we could work on.

Fodor: Wittgenstein owes us an explanation of how the natural language works. That can't be just assumed, and if you look at the explanation that he gives, it looks like a form of behaviorism.

Friedman: I don't think it's good strategy to just dismiss Wittgenstein here. He was one of the first people to recognize the power and complexity of the computational view and, indeed, to adopt it as the basis of a metaphysical system. The greatest sytem of 20th century metaphysics was based on this idea of a polyadic logical calculus. He was one of the first to recognize it and to see its fundamental importance. Then he meditates on the idea some more and decides that there's something wrong with it, not because we have to add semantics to it, but because when you even think about the idea of rules in syntax, you need more than one set of them. You need some of them talking to each other.

Newell: That's a trivial condition to satisfy in the sense that there are lots of machines around autonomously talking to each other.

Hayes: Sure, they talk to one another a lot faster than you and I talk to one another.

Friedman: I'm not criticizing machines. Machines can talk to each other—fine. I'm saying that to try to understand thinking by first designing a machine that thinks and solves problems, and then goes out and has conversations, is backwards. We first need to design machines that can talk to each other, and then they will be able to solve problems.

Newell: That's not the way evolution did it.

Friedman: We thought and talked before we communicated?

Newell: A number of animals did, yes. Before there was language there was a long history of the evolution of the organism.

Friedman: They didn't solve problems like we do with anything like the logical calculus, which is precisely what got this whole ballgame going.

Hayes: It absolutely isn't clear whether they were using something like a logical calculus or not.

Haugeland: I have a comment about the status of the knowledge level and what's going on there. I think it's really neat to draw a distinction of this character, and I think it's really neat that it's coming out of AI too. But I'm not altogether happy with the details of the way you've done it, and in particular with seeing a knowledge level as a level just above the symbol level in the same stack of levels that goes down to devices or whatever. I think there are really two ideas run together in what you've called the knowledge level. I can get at one of them by going back and talking about the symbol level a bit. It's clear in complex interesting systems that the symbol level is not unique in an important sense. In some sense, in fact, none of the levels are quite unique; you could mess around with them some. But there's a more robust sense in which the symbol level is not unique, that is, there are layers of symbol levels. There's assembly code, then you go up and you've got some higher-level language and so on. And so you can go up and presumably there will be some top one of those for any given system that's running at that time. If it's an AI system and if it works, the symbols in the top-level system (at least many of them) are going to have reference outside the system to Julie Christie and whatnot. Some of the symbols are to be understood as referring to inner parts of the machine. Then there are higher symbols that are composed out of these in some sense, and, in virtue of how they're composed out of them and how they all interact and various things like that, they refer to things in the outside world. The top symbol level in your stack does seem to me to be on the same stack, that is just more layers up from the devices, with one addendum: They are interpreted and have meanings and references. While the lower-level symbols have references *into* the machine, the higher-level ones have references *out of* the machines—that makes a difference.

With regard to an interpretation, there are always some constraints or conditions of adequacy on the interpretation. You can't just interpret any old object as being about any old topic, unless you appeal to the Skolem–Loweheim Theorem, and that's a sign of some kind of deep mistake that I've never understood. But in decent company you can't do that. Nobody has really articulated what the conditions of adequacy are on the interpretation of the symbols in the various levels in this symbol system hierarchy. I don't think anybody in your business or anybody in my business has, and we should all work together on that and see what comes of it. But roughly, something like the operations on the symbols, the changes that they make, the transformations that they go through and so on, must in some way be appropriate with regard to whatever it is they are supposed to be about. And that, I think, is where your notion of rationality fits. I take the top level of the symbol levels, and its interpretation in the world, together with its associated notion of rationality (or whatever sort of appropriateness is the governing condition on the interpretation) to be one of your lines—one of the two ideas that I thought were blended together in what you were thinking of as the knowledge level.

Now the other point, which it seems to me important to keep conceptually distinct, is not another stage in this same stack of levels. It represents a different kind of enterprise. There you don't look at the system in terms of how it's composed (what the components are, and the laws of interactions among those components in terms of which its history progresses), as you do in the case of all the system levels you talked about. Instead, you look at it from the outside as a black box and you see how it interacts with the rest of the world, or you think of it in terms of its I/O functions. You pay no attention to how it's composed, what it's composed of, or how its components (if any) interact; rather you look only to external interactions. Those interactions may be chaotic: You may be able to make nothing of them. On the other hand, they might not be chaotic. They may be a simple alternation pattern—black ball, white ball, black ball, white ball—or they can have much more subtle and intricate patterns that are extremely difficult to characterize. And I take it that what Dennett calls the "intentional stance" is all about finding a particular vocabulary and apparatus for describing this intricacy, i.e., making sense of finding order in the elaborate behavior pattern of a black box. You do that by characterizing the box. It is a box of the "has-goal-of-getting-to-the-grocery-store" type. I avoid saying that there is in it a goal of getting to the grocery store because that confuses it with the other sense of the knowledge level. In this second sense, I'm not talking about what's *in* the black box, but rather *characterizing* it as a box of a certain kind. The intent of calling it that kind is to pick out a thread in the fabric of its biography which I've noticed in its complicated behavior pattern.

The same goes for beliefs as well as goals. It turns out that you can come up with a description of the object in terms of which you can predict what it will do in novel situations, explain why it did what it did in other situations, and so on. It is a matter of fitting the individual events in its biography into an overall pattern of a consistent biography. The type of consistency is rationality again, but now we're talking about the fabric of its behavior pattern, not about its constitution. I think both aspects of the knowledge level are important. It's also important to keep them separate for various reasons. One reason is that if I can think of any way of separating two things, then it bothers me if they are not separated. Another reason is that the intelligible behavior pattern is the explanandum of psychology. That's the observed phenomenon which is remarkable and amazing and calls for explanation: These boxes behave in this remarkable way which we've been able to characterize. Proposing a hierarchy of the type that Newell has described, which ends with a high-level symbol system, is one way of explaining how that can be.

Newell: There is a kind of a one–one mapping about what you would get out of a user booklet.

Haugeland: Well that's what I would have gone on to say if I hadn't used up my time. I think I just will. It seems to me if you have a knowledge level in my second sense, the black box behavior pattern sense, then there are four distinguishable possibilities as to how that might relate to a knowledge level in the symbol system sense. One possibility is that there is a one-to-one correlation. It would turn out that for every goal attributed from the outside—"Dennettwise"—there is a goal locatable inside as a high-level symbol structure. Another possibility is that there could be a partial correlation, which is, I gather, what you were suggesting. Some of the things line up pretty well; others don't line up so well, but you still get something that's useful. The third possibility is that though there's an internal symbol system which has the appropriate input–output function, its internal structure doesn't line up at all illuminatingly with the, as it were, surface structure of the external interpretation. They might predict the same I/O, but how they get at it is different. I take it that that's one of the points Dennett makes in the Introduction to *Brainstorms*. The fourth possibility is that there isn't any internal symbol system. There's some other thing—a soul or jelly or whatever subserving the interpretation. All four of these possibilities are compatible with there being a knowledge level in the externalist sense. So that's another reason for keeping them separate.

Newell: For any relatively wide scope of behavior, it's not the case that you're going to find a symbol system which can be replaced with just the assumption that it has the knowledge and will do whatever is required, given that knowledge, to achieve the goal. Consequently, the knowledge level is always an idealization over the symbol level. In a sense, if you think of building better and better symbol systems, it's the limit point. I don't have difficulty with saying that those are truly separate things, as in one sense they are for the engineer. When the engineer wants to design a widget that does some intelligent task, he starts off by specifying knowledge and goals. He doesn't bother to distinguish which symbol level he's talking about. Do you believe it's an empirical matter whether there exists any other way in the universe to achieve this? I.e., do you think it's an empirical matter that to have a system which behaves over a sufficiently wide variety of types of knowledge and goals requires a symbol system? Do you think it only represents your ignorance that you can't do it with jello?

Haugeland: I believe that it's an empirical matter whether or not you can do it with a symbol system and also whether you can do it with anything else. I think it's empirically false that you can do it with jello. It's the empirical falsity that matters. But should it turn out—you were asking a conditional—should it turn out that it can be done with a physical symbol system and in no other way, would I then be prepared to relinquish the conceptual distinction? No, why should I? It was believed for some time, and for all I know it still is, that gravitational and inertial mass were exactly coincident. But it was a conceptual mistake to confuse them.

Putnam: It seems to me that the answer to the question of whether there is an exact line-up between things at the knowledge level and things at the syntactic level has to be "no" on any view. There are very simple examples, you don't need cases of confused speakers. Let's imagine, as in folk anthropology, that the Eskimos have many, many words for kinds of snow—let's say 34 words for kinds of snow. Let's also imagine—I don't know whether this is true or false, but there are analogous cases in anthropology—that they don't have any one word for "snow," that the idea of having only one word for these 34 kinds of different things has not occurred to them. Now I might explain a piece of Eskimo behavior incorrectly by saying he knows that snow is white. The Eskimo would have to take a long time even to subvocalize the thought. Well, you can see the point, the point of indirect discourse which is, of course, essential. You don't really have full-fledged *believes that's snow*.

Fodor: That doesn't make your point, Hilary, because nobody in his right mind thinks what Dan thinks I understand. And nobody in his right mind would think that: namely, that the kind of syntactic theory that you have is that for every propositional attitude there's a token in your head. What you have is closure relations, and computational relations and functions over them. And the way you do the fudging in this area is figure out which functions correspond to propositional attitudes and which don't. Nobody thinks there is a token-to-token correspondence. But what I really wanted to ask John Haugeland (and I'm really asking this to Dan Dennett as well) is why do you take it for granted that there is a behaviorally specifiable—or an I/O specifiable—notion of a knowledge level? Maybe, and this seems overwhelmingly plausible to me, our notion of knowledge and rationality and stuff like that is shot through with psychological speculation. You say we simply can't do it if there's room for that sort of relation.

Haugeland: Well it may be that it's expressed in terms of hypothesized mechanisms, but it seems to me that it is, in fact, an externalist peripheralist because that's all anybody has ever had to go on.

Fodor: I don't see any reason to believe that.

Haugeland: The question is, what is it a way of characterizing? It's a way of characterizing the non-randomness, such as it is, in our biographies.

Fodor: But look, John, that's the data for it; that doesn't show that that's the burden of the concepts. For all I know—and everything we know about every other case of concept formation makes this plausible—for all I know, what goes on in folk psychology (which is, I take it, what we are talking about) is that you adopt, or are probably born with, a picture of the mind which among other things has very high predictive success of making sense of the I/O relations. Now I see absolutely no reason to suppose that you could abstract out the theory with which the thing is riddled. Homer tells you not just what Achilles did but what he was thinking about and what he wanted and stuff like that. You could abstract all that out and get a notion of belief/desire psychology that's I/O interpretable.

Dennett: But the suggestion is that if you say of Achilles what he was thinking, that's true; we say that Homer said it and various other people did. The question is: And what is it that we're saying when we say that? Are we, in particular, saying of Achilles that there occurred within him an event, a particular of a certain sort?

Fodor: I think that's what everybody takes himself to be thinking. Why should we turn into behaviorists at this late stage of the game? Why on earth should we have done that? Why should we have accepted these behaviorist scruples?

Dennett: Because it's innate in us to do that.

Block: The point that I thought John was trying to make—which seems to me got utterly lost in the discussion (though I think Jerry Fodor was trying to get at it)—was that there are two totally separate questions here corresponding to this ambiguity in the notion of the knowledge level. One is, is there a correspondence between people's actual propositional attitudes and symbols, and what kind of correspondence is there? The other totally different question is, what kind of correspondence is there between input–output stuff and either one of these other two, let's say the symbol level? Now I take it, the point that Jerry was getting at about behaviorism is that it is well known that there is no such correspondence, between the input–output level and either of these two levels.

It's a familiar point. I think Hilary Putnam actually made this point best with his perfect pretender example. It's perfectly clear and this ought to be something that all philosophers learn in graduate school. It's easy to think of cases where there's a many-to-one relation between propositional attitudes and input–output relations. There's an infinite set of input–output relations. One kind of example is a case where you've got one set of beliefs and desires producing a set of input–output relations, and then you have the same set of input–output relations produced by a pathological set of beliefs and desires which include say, just to take a weird kind of example, a desire to give everybody the impression that you have certain beliefs that you don't actually have. Hilary's example was a perfect pain pretender who acts like he's in pain. But you could also have someone who assigns infinite weight to giving the impression (that is, his preference function assigns infinite weight to giving the impression) that he has beliefs B_1 through B_n when really he has beliefs B_1' through B_n'. The point is that the constellation of beliefs B_1 through B_n plus normal preferences produces the same input–output behavior as this pathological preference B_1' through B_n'.

Hayes: Ned Block, as a non-philosopher who didn't go to philosophy graduate school, I would like to ask whether anybody has actually come up with an actual example of that in any detail at all?

Fodor: Pat, that's the wrong question. The right question is, "Has anybody given an input–output analysis of any propositional attitude?" And the answer to that is "No."

Dennett: First I should make a confession. I started as sort of barefoot boy here. I cannot somehow bring myself to believe that the issues that bring philosophy and artificial intelligence and cognitive psychology together are really the sort of metaphysical issues that you get to via things like the Löwenheim–Skolem Theorem. I don't myself yet understand—maybe I never will—how those arguments are supposed to run, although Hilary Putnam, during the coffee break, has been trying to explain it to me. I propose just to sort of step back from that abyss and adopt, as a sort of crutch, naive naturalism, which no doubt assumes all sorts of metaphysical naivetés, and then stick to my guns about the status of the knowledge level or the intentional stance.

Now having said that, just to try to sort of close off that avenue of discussion which I can neither contribute to nor follow further, I'd like to draw people's attention (I do this somewhat reluctantly) to a dog that hasn't barked here yet. I take this as a sign of real progress in the development of cognitive science. There have been many occasions in the last two days for people to engage in what we might call the "Dreyfus–Minsky Three Step," and it hasn't happened. For instance, here was one occasion: Allen Newell said of some system, of some very crude system, that it knew that *higher than* is transitive, and nobody started doing the Dreyfus–Minsky Three Step. The Dreyfus–Minsky Three Step: First Dreyfus' move is, "You just said, and I can hardly believe my ears, that that system knew that *higher than* is transitive. Now if it really knew that—in order really to know that—it would have to know this and that and this; and you'd have to understand this and you'd have to be able to do this; and you'd have to be embedded in this culture and so forth. And your system doesn't do any of this; therefore, it doesn't know that *higher than* is transitive." To which the reply (actually it has two parts) is, "Tell me exactly what knowing that *higher than* is transitive involves, and then it will be child's play for me to add that to my system." Or there's another routine: "Okay, I'll put scare quotes around the word 'know' and get on with my business."

Now, what I want to suggest about that move is that, although to some eyes that looks like a sort of a lapse into sloppiness and crudity, there's another way of looking at it, which is progress. After all, if we're actually going to discharge our homunculi in these systems, we don't want them to be too smart. And so it's not clear that we want the subsystem that has to do with how the information that *higher than* is transitive *really* to know that, because then maybe Dreyfus is right: It would have to be too darn smart to be some little subsystem in some larger system. We only want it to sort of quasi-know it because if it only has to have some of the flavor of knowing that *higher than* is transitive, then it will be all that much easier to: (a) design, and (b) in virtue of designing it, discharge it.

Now the reason I draw attention to this is not to suggest that we start dancing that dance (I think we all learned whatever we were going to learn from that), but to point out how the dance is actually made possible by exactly the separation between the knowledge level and the physical symbol level that we've been discussing. The reason that the first step in the dance is always possible is that there is an indefinitely large normative demand on believing that P, for any P, for desiring that Q, for any Q; that's where the rationality assumption generates ad lib and ad infinitum and ad nauseum demands on what really knowing that would involve. So there's always plenty of room for the first Dreyfus move, no matter how smart the system one is criticizing. The point to make, though, is to notice that we can also do it with people. You say that your child knows that you're a college professor, and you say, well, if my child really understood that, he'd have to understand the difference between universities and junior colleges and so on. And so it will go for everybody, for virtually any belief. It's that idealizing feature of the knowledge level, which comes from the rationality assumption, which permits one always to play that game. And that is a way of seeing the radical approximation relation between the knowledge level and the physical symbol level—at least that's one way.

There was an attempt by John Haugeland to present my view as strongly opposed to one feature of Newell's view, and that was with regard to the isolation of the knowledge level from any system level at all. Newell's response to that was that he saw interpenetration of these levels all through the discussion. And I think that that is in fact right. I don't think there's a conflict here at all. I think both that one must maintain that a knowledge level or an intentional stance characterization is a black box, entirely design-neutral characterization, and that it nevertheless often is utilized in discussion of design problems that are deeply interpenetrating into the system. And in fact we've been seeing a perfect example of that this very morning when, for instance, it was claimed that in the visual processing model there is information about physics. Now to say that there's that information about physics in it is not to say how it is in it—it is not to say whether it's implicit in the algorithm. As long as you're just talking about whether the information is there or not, you're talking about the knowledge level, even though you're not talking about the whole person—you're talking about just a little part of the visual system. And you're arguing as follows: A certain task is performed by that subsystem which could not be performed if it did not have a certain amount of information. That information is simply a logical prerequisite of getting the answer. Since it gets the answer, the information must be there. The next question is, "How do we put the information in there?" And then one has set a rather nice design problem to which there might be rival solutions. So, in effect, although the intentional system stance is holistic and black box, there is what you might call local holism. You can take subparts of the whole system you're describing and, pending further elaboration, treat each little subpart separately. It seemed to me, in fact, that as John was describing Newell's position, or trying to contrast mine with Newell's, he was really just contrasting two parts of Newell's position. That is, he was contrasting what Newell was calling the knowledge level with what Newell was calling the symbol level.

Smith: I guess there're two things I want to say. The first point I guess is methodological. I think it is true that Newell and I are both trying to report on the same field, and I take it our understanding of it is the same. And ultimately we ought to agree. Presumably as well as reporting, there's a certain amount of rational reconstruction going on, and we're trying to present a coherent account. And one of the things that I think that AI as a field has been guilty of is that its practice and its theoretical report or its public report haven't been as closely connected as they ought to have been in various ways. One place where I think the practice often outstrips the report is in subtlety. That's maybe to be expected. Another thing is that the use of terminology in the report hasn't necessarily been the same terminology that other fields which are represented here today might use.

The other point is just a minor, technical comment on the whole notion of implementation. John Haugeland pointed out that often in the AI systems we build there are many symbol levels. The relationship between one and the other is called implementation. And we agree when we have one implementation of a system in another one, but we don't actually have a handle on exactly what it is that's implementing. But there are some striking things about that implementation relationship, I think. One of them is that the properties of an implementing machine correspond only sort of configurationally, and incorporate bodies to the ingredients down one level. You cannot find one-to-one correspondences between the internal symbols of one level and the internal symbols of another. What's funny concerning this discussion as to whether there is going to be a certain kind of correspondence, is that there are just as many technical instances of that lack of correspondence. I think a lot of what computer science has done is to try to collect theoretical notions that enable one to construct these extraordinarily complex systems that relate in these configurational ways. There are these systems of hundreds of thousands of lines of code where the important properties—the properties you make your million dollars for when you sell them on the market—are external configurational properties, only wildly related to the account of symbols even at the top levels. It's because of that, I guess, that I don't know what kind of top symbol level the computational metaphor is going to be successful in engendering. It just seems to me an empirical matter whether the correspondence that John talks about can be drawn. To be honest I don't think it can be, I have to confess that I'm a pessimist. One of the things that is striking about a lot of the modern compiler architectures is that they have very much the same vocabulary describing them. I just think it's also striking that a lot of the characteristics that have been described today as holding at the knowledge level are in fact becoming evident at the description of these formal symbol levels.

Newell: Let me pick up on that. Part of the reason for shoving the knowledge level over into the same stack turns out to be that all of these issues between the knowledge level and the symbol level show up at all these other levels as well. There is a common problem that at every level there are I/O descriptions and then alternatives. The one thing I don't quite understand is that once I arrive at the symbol level, I find it extremely easy to build versions of other symbol level architectures within this implementation; there are levels after levels in which I can even go and recurse. After I realize FORTRAN in LISP, I can then go realize LISP in FORTRAN, and FORTRAN in LISP, and I can just keep that going as long as I want. There's no problem in that. There's a little bit of an issue about the extent to which you can do that at lower levels. It's not quite clear to me whether you can or cannot, in this sense of "realize," realize one register transfer system in terms of another. We never do it very much, but we do occasionally realize register transfer modules in terms of combinatorial logic, so we do it both ways, but not very often.

Smith: It is true we simulate architectures all the time.

Newell: The issue is whether, once I arrive at, say, the register transfer level in one register transfer system, I can then realize a different register transfer system in that register transfer system. Now you can certainly go up to the programming level and then simulate another register transfer system, but that's not what I mean. So there may be some things which are not homogeneous. One of the reasons for wanting to put all these together is that every problem one finds in one, one finds in another. One finds the analogues up and down, so it really looks like we're just talking about how levels of systems relate to each other; certainly at the level of programming, it has led to this great profusion of level after level of systems and virtual machines and so forth, which stack up as high as you want.

Haugeland: I just wanted to reply. A sequence of remarks on a larger scale of patterns in the discussion is ending up aligning me with the thesis that there could be a one-to-one correspondence between the knowledge level and the top symbol level. And I want, in effect, to beg off of that. I think that the point that Brian Smith made about the looseness of the relation, even in different levels of explicit implementation, is extremely important. And that has to do with the configurational properties of the unit at the higher level, of the pieces at the lower level constituting the unit at the higher level, and so on.

The other difficulties brought out have also perhaps somewhat watered down analogues—for instance, the difference between what is explicitly represented or encoded versus what is implicitly represented or encoded, alternative interpretations you can interpret this way or you can interpret that way subject to whatever constraints you can think of expressing, and so on. So there is a lot of slack available between what these symbols mean and what these symbols at the level that they're implemented in mean. And I was allowing as the strongest possible relation between the knowledge level and the top symbol level that they could be at most as closely tied as are the adjacent symbol levels.

Fodor: There's a danger of making a technical mistake here. The condition that's wanted is not that for each symbol at one level there should be a corresponding symbol at the other, but simply that a necessary and sufficient condition for states described at one level is that they should be specifiable in the vocabulary of the other level. It's a much weaker condition. So the question is, "Can I give necessary and sufficient conditions for believing that P in terms of some computational relation to the symbol?" It's not, "Is there a symbol that I have in my head if and only if I believe that P?"

Haugeland: I think I just drew that distinction, didn't I?

Fodor: I thought you didn't. I thought what you said was, given the looseness of reduction between other levels, it's unreasonable to suppose that we're going to get anything better or that we're going to get reduction between the knowledge base and the symbolic level. I was trying to suggest that you were describing the constraints for reduction much too rigorously. You're certainly not going to find a specific token or top symbolic token such that if you have that in your head you believe that P. You'll just get what you get at the other levels. But no argument has been given here against the thesis that states defined at one level can be given necessary and sufficient conditions in the vocabulary defined at another level.

Haugeland: Subject to various kinds of indeterminacy, yes, I've put that as an upper bound on the strongest relation you could have between the knowledge level and the top symbol level.

I just wanted to reply to your final line, that the thrust of my remarks was to remake Newell's point in the first place: the distinction between the knowledge and the symbol level. In a way I think that's right. That is, I don't take myself to have been lunging at you with savagery in my eyes. What I wanted to bring out was your locating of the knowledge level at the top of a stack which had a sort of coherence in its successive layers. I think that the knowledge level—I'd rather call it the knowledge stance and not another level—is different in kind conceptually from the top of that stack. And that has to do with whether it's externalist or internalist and various sorts of things like that.

Newell: But in fact, what is at the top of that stack that you were trying to distinguish is not any of those symbol levels. It is, in fact, the closure of what produces something which I was calling at the knowledge level. So in fact the kind of thing you were talking about, the top of that stack, that top symbol system if one goes up from that is such that you can then have a description of a system there which is the closure of the way that particular symbol system will behave. Then that has a description at the knowledge level. But if I now isolate what I took to be your, let's say, I/O clean statement, if I simply isolate that, I can find other systems that will realize it.

Putnam: It's getting late, so I'm going to have to ask people to meet me half way. It might be that there is one and only one assignment of the knowledge level which would be reasonable by various informal criteria we actually use, without there being a finite set of axioms in terms of which we could pick out what that reasonableness consists in. Now that would not show, by the way, that the knowledge level is nothing more than the symbol level. It would not show that it's reducible to the symbol level even in the sense that Jerry Fodor was alluding to, viz., the possible definition of terms in the knowledge level in terms of the terms in the symbol level. That would mean that you could believe that the knowledge level is irreducible but supervenient, as philosophers say, on the behavior level.

Now I think that maybe that, or something close to that, may be Dan Dennett's position; that is, he believes that the intentional stance from the knowledge level is not reducible to the behavior level but that the knowledge level is supervenient on idealized behavior. I don't believe it's even supervenient, but I wanted to make the distinction between believing that the knowledge level is reducible (in the most liberal sense that the logician can understand) to the symbolic level and believing that it's supervenient on the symbolic level.

The other distinction, of course, is that of believing in reducibility—not just supervenience—system by system, or species by species, and believing in supervenience at the level of the most abstract theory of cognitive systems. One could find possible definitions of "believes" and so forth which work for the whole species.

THEORETICAL ISSUES IN
COGNITIVE SCIENCE

ZENON W. PYLYSHYN

University of Western Ontario
Series Editor

Language Learning and Concept Acquisition: Foundational Issues, William Demopoulos & Ausonio Marras, Editors (1986)

From Models to Modules: Studies in Cognitive Sciences from the McGill Workshops, Irwin Gopnik & Myrna Gopnik, Editors (1986)

Meaning and Cognitive Structure: Issues in the Computational Theory of Mind, Edited by Zenon W. Pylyshyn & William Demopoulos (1986)

The Robot's Dilemma: The Frame Problem in Artificial Intelligence, edited by Zenon W. Pylyshyn (1986)

MEANING AND COGNITIVE STRUCTURE:

Issues in the Computational Theory of Mind

edited by
Zenon W. Pylyshyn and William Demopoulos
University of Western Ontario

1986

ABLEX PUBLISHING CORPORATION
NORWOOD, NEW JERSEY