In Proceedings ECAI, Brighton 1986

REFERENCE WITHOUT CAUSAL LINKS

Aaron Sloman
Cognitive Studies Programme
University of Sussex
(Now School of Computer Science, Univ. of Birmingham)

Abstract

This enlarges on earlier work attempting to show in a general way how it might be possible for a machine to use symbols with 'non-derivative' semantics. It elaborates on the author's earlier suggestion that computers understand symbols referring to their own internal 'virtual' worlds. A machine that grasps predicate calculus notation can use a set of axioms to give a partial, implicitly defined, semantics to non-logical symbols. Links to other symbols defined by direct causal connections within the machine reduce ambiguity. Axiom systems for which the machine's internal states do not form a model give a basis for reference to an external world without using external sensors and motors.

Keywords

Understanding, semantics, symbols, reference, reality, logic, inference rules, models, meaning postulates.

Introduction

Before knowledge there must be understanding. In order to know or wrongly believe that something is or is not the case you must understand some representation of the possibility. How? What conditions would enable a machine to understand symbols that it manipulated? Unlike most philosophers who discuss meaning and understanding (e.g. Putnam 1975) I am not concerned with the use of an *external* language for communication. The question is - how can a machine understand a language used internally for planning, reasoning, remembering, etc. This question has had little attention from AI theorists, probably because most regard it as obvious that machines understand programming languages, or at least machine codes. (But see Woods 1981, Cohen forthcoming.) Clearly computers can already manipulate symbols that *people* understand, but not everyone agrees that computers themselves could ever understand symbols, or have any other mental states or processes. There are three main attitudes. Behaviourists take our question to be: could a machine produce behaviour indicative of understanding (e.g. passing the 'Turing test')? Dualists ask: can apparently non-physical but introspectable mental processes like ours exist in machines? A third option is to ask: what sorts of internal mechanisms could enable a machine, or person, to understand?

The behaviourist often relies on the old argument that we can never know anything about the contents, or existence, of other people's minds except on the basis of their observable behaviour. We assume that our parents, children, friends and enemies all have minds and mental states, though we have nothing but their behaviour to go on, albeit very subtle kinds of behaviour. Tests that are adequate for people should be adequate for machines, never mind what processes produce the behaviour. This answer is implicit in much AI work.

Neither behaviourists nor dualists are concerned about explanatory power. Yet 'understands' is an explanatory concept. Neither behaviour nor a peculiar unanalysable kind of entity like a soul can in itself explain success in thought, action and communication, whereas understanding does. Moreover, the primary use of symbols is NOT to communicate with others but to provide a usable store of information, instructions, plans, to allow inferences to be made, and to formulate questions, problems, and goals: in short *representation is prior to communication.* This understanding of internal symbols cannot be tested directly by external behaviour. (Does behaviour show whether dogs use symbols, or what sort?)

Understanding processes may not even be internally accessible. The introspectible and the mental are not co-extensive.

Explanatory insight requires the third approach, adopting what Dennett (1978) calls the 'design stance', exploring mechanisms that might support meaningful uses of symbols. I'll start with a logic-based machine, then generalise.

Recapituation

In Sloman (1985c) I discussed general conditions for using symbols to refer, listing an (incomplete) collection of 'prototypical' conditions satisfied by human uses of symbols, and showing that many of the conditions are satisfied even by computers without AI programs.

Unprogrammed computers can interpret symbols in the machine language because of built-in causal links between on the one hand bit patterns and on the other hand memory locations or their contents, properties and relations or operations on the memory. Some symbols influence the selection of memory location, others influence the action to be performed there. Conversely, when a machine interrogates its memory, states of the memory can cause appropriate new symbols to be constructed and stored. So, the mapping from symbols to locations, properties and relations is significant for and can be used by the machine itself because of two-way causal links. (The word 'use' does not imply purpose or consciousness: plants use oxygen and supports.)

Built-in causal connections explain how machines can interpret symbols as instructions or questions about internal states and processes. Assertion and external reference, usually require additional programming. (See Sloman 1985c. Cohen [forthcoming] makes similar points.) Let's examine internal reference more closely, and then return to external reference.

All reference is to virtual worlds

Causal mechanisms give machine codes for addresses and instructions a semantics that attributes a formal structure to the designated world. This world is usually taken to be a linear configuration of bit-patterns. Yet it is not physically linear, and the bit-patterns are abstractions relevant to the way the machine works, rather than objective physical entities.

Implementations of equality of bit-patterns can vary enormously, and usually do not require physical identity: measurements may be thresholded so that different patterns of voltages are treated as identical bit-patterns. Other differences in the fine structure of physical components may be ignored completely. New technologies can use new physical representations for a string of bits. They may even differ from one part of a computer to another: bits in a fast memory cache may be very different from both bits other parts of the memory. A portion of the memory may be replaced with a component using a new, but functionally equivalent device. The class of potentially usable physical mechanisms is quite open-ended.

So two implementations of logically the same structure may differ radically in how symbolic patterns are represented and how they are translated into physical processes representing the same action, such as comparing the contents of two memory locations. The internal world of a computer is therefore an abstraction. Symbols referring to memory locations, their contents or actions thereon, refer to 'virtual' entities, relations and processes, that may be implemented in the physical world in different ways. The common use of the phrase 'virtual memory' is a partial acknowledgement of this.

The referents of symbols in a machine language therefore do not form 'natural kinds' from the point of view of a physicist, for there is no simple correspondence between the truth of any assertion about the computer's memory, and the state of the world as viewed by a physicist. I am not denying that the machine uses symbols to refer to physical objects. Rather, the reference is indirect and 'wholistic', in the sense that individual symbols make sense because the whole system can be attached to the world. (This is explained further below. See also Quine 1953.)

Related arguments show that people do not refer directly to objectively individuated physical entities or their intrinsic properties, relations, states, etc., but rather to a reality conceived of in terms of a system of concepts we find useful. This must be true of concepts used by any thinker or perceiver, even a physicist during office hours. The environment, as conceived of by a symbol user, is inevitably a ‘virtual’ world with properties defined by the system of concepts used. Usually these will have been selected because of their relevance to the needs or functions of the user (or a class of users). Other people, other animals, or other machines may conceive of the same reality in a totally or partially different way. We all inhabit, think about, and act in virtual worlds, in this sense. So would a robot.

Virtual worlds are not mere abstractions: they are *implemented* or *embodied* in one physical world, much as a particular design of computer can be implemented in actual hardware, whose failings may show up in bizarre behaviour of the virtual machine. In philosophical jargon, one world can be ‘supervenient’ on another, as temperature is supervenient on motion. Two supervenient worlds may share a substrate. So the fact that two organisms inhabit different virtual worlds doesn’t mean they live in totally unrelated realities, as relativists claim. An event in the common underlying world may affect them both, albeit differently. Living in a different virtual world doesn’t help a mouse escape being eaten by an owl or one culture being destroyed by another. However, it would be incoherent to try to describe the common underlying reality in neutral terms.

Theory-based reference

An ordinary computer’s ability to refer to its simple internal virtual world is severely restricted. For example, it can check the contents of a location at a certain time, but cannot answer a question about what the contents were at some past time. It can’t obey instructions like:

If you’ve changed the value of X more than three times then perform action A.

Some information about the past could be stored, but no machine could keep complete and explicit historical records of its internal states and events unless the memory was constantly growing to include the new information -- an explosive requirement. A practical system would keep only partial records. A theory about its constraints could then allow inferences about unrecorded facts.

The need to use a theory is even more obvious if the machine has to answer questions about the future. Like a software designer, it could answer questions about the future contents of its memory if it examined its programs to infer their effects, or non-effects. If no instructions refer to a certain location and that location is not connected to external transducers, the machine could infer that the location will not be changed. Of course, inferences about a virtual machine can founder if the underlying reality misbehaves.

The important point is that the ability to use symbols to refer to past and future states of a world capable of change, even a ‘totally accessible’ world like that of the machine’s own memory, requires the use of an explicit or implicit theory about the constraints governing the world and the sorts of changes that can occur in it. The same applies to reference to the present, in a partly inaccessible world. Put another way: no interesting world can be totally accessible: theory-based inference will always be required for reference to some portions of the world. In this sense knowledge (or at least belief) is prior to some forms of understanding. So the dependence between knowledge and understanding is mutually recursive.

The theory used may be explicit in a representation manipulated by the machine, or implicit -- ‘compiled’ into mechanisms, as in most vision systems. The best understood technique is to use a logical language to express the theory and logical principles for making inferences. Before examining more general capabilities, let’s examine what is involved in giving a machine a grasp of logic (extending my 1985c).

Giving a machine a grasp of logic

Computers easily manipulate boolean values and use logical symbols analogous to ‘and’, ‘or’, ‘not’ with semantics defined by truth-tables. Circuits can be built with such operations wired in. This is a primitive inference-making capability. If truth-values for **p** and for **q** are checked directly, then the machine can infer the truth-value for **p & q.** Similarly for ‘or’ and ‘not’. Computers can also be programmed to store and derive consequences from collections of axioms expressed in predicate calculus. But can a machine really understand first-order logic’s unrestricted quantification? What does it mean to talk about all the real numbers, all the possible people that might exist, all the legal programs in a certain language.

Quantifiers ‘For all x’, and ‘For some x’, (first analysed by Frege) are not definable by truth-tables, though when predicates with finite extensions are used, it is possible to use truth-tables. If the extension of P is a finite set (e.g. a set of locations in the computer’s memory), then

(1) ‘Ax(Px -> Qx)’ (‘For all x, if Px then Qx’),

can be shown to be true or false by treating it as the conjunction of all instantiations of ‘Qx’ with members of the set. So if a and b are the only elements of P’s extension, then (1) is equivalent to ‘Qa & Qb’, and a truth-table can be used to evaluate it. Similarly, if P has a finite extension, then

(2) ‘Ex(Px & Qx)’ (‘For some x, x is P and Q)

can be treated as a disjunction of instantiations of Q.

Despite the factual equivalence, there is no definitional equivalence, since quantifiers would be needed to assert that there are no other individuals to consider in addition to those listed in the conjunction or disjunction. The inference rules for universal and existential quantifiers are not restricted to finite sets. Some of the inferences are unproblematic, even for infinite sets, for example:

Universal Instantiation (UI):
AxFx => Fa

Existential Generalisation (EG):
Fa => ExFx

More problematic are the reverse inferences: universal generalisation (UG) and existential instantiation (EI), especially if the domain is infinite. In fact, it is not at all clear in what sense people understand unrestricted quantifiers (as shown in part by the existence of non-standard models for any consistent set of axioms for arithmetic).

For dealing with finite sets, we can avoid problematic quantifiers by introducing a form of assertion that certain individuals a1, a2, .. ak comprise the total extension of a predicate P, e.g.

Extension(P,a1,a2,a3,..ak)

If we allow sets, then this could be expressed as a binary relation:

(3) Extension(P, {a1,a2,a3,..ak})

Using the Extension predicate it is possible easily to express a fact of the form ‘a, b and c are the only M’s present’, making it unnecessary to use non-monotonic reasoning to infer that there are no other instances of M. Extension facts could be expressed in ordinary first order logic, but the notation would be very clumsy, including a component something like:

Ax {Mx -> (x = a V x = b V x = c)}

('everything that is M is identical with a or with b or with c'). The search space for proofs involving expressions like this is very awkward. For assertions about relations, the use of ordinary logic is even more clumsy, whereas a slight generalisation of our notation copes with the extensions of n-ary relations, e.g.

(4) Extension(Loves, {{John,Mary},{Fred,Mary},{Mary,Tom}})

Lambda-notation could be used to represent complex predicates occurring as the first argument, but in some contexts a notation using a new form of quantifier might be more convenient, e.g. expressing (3) and (4) as

(3') AOx{a1,a2,a3,..ak}P(x)

and

(4') AOxy{{John,Mary},{Fred,Mary},{Mary,Tom}}Loves(x,y)

Where 'AO' can be read as 'All and only', and the quantified variables range over the specified set.

Mechanisable procedures could directly check the truth of such 'Extension' assertions in the machine's accessible world. Perceptual mechanisms, for example, need the ability to check bounded quantified assertions: X is a square only if the extension of the predicate 'side of X' has exactly four members. Such inferences may be disguised by being compiled into special-purpose modules, such as structure matchers. By making them explicit we may explore their power and limitations. For instance, checking an Extension assertion may require knowledge analogous to 'naive physics' (Hayes 1969, 1985) for interpreting perceptual evidence. When I look into a small box I can easily discern the extension of 'apple in this box', but not necessarily 'flea in this box'; whereas visual data doesn't suffice to determine the extension of 'apple in this room' unless the room contains no objects large enough to hide an apple. Learning such meta facts may be an important form of child development. (These notions need further exploration and elaboration.) Quantifiers ranging beyond known individuals would still be required for a world where extensions of sets are not known, even if they are knowable. For example, this would be needed in questions, like 'Extension(P,{a,b,x..})' ('is there an x other than a and b which is P').

New inference rules like the following would allow Extension facts to play a role in reasoning with quantifiers:

```
Given:
   Extension(P,{a1,a2}) & Qa1 and Qa2
infer:
   Ax(Px -> Qx)

Given:
   Extension(P,{a1,a2}) & Ex(Px & Qx)
infer:
   Qa1 V Qa2
```

The use of 'Extension' or the 'AO' quantifier would not add expressive power to ordinary logic, since every occurrence can be replaced by a translation including a very clumsy disjunction of equalities, as indicated above. Similarly, the above rules would not validate any new conclusions. However, they would permit in a single step useful inferences that would otherwise require several awkward steps and additional searching.

The main point is that there appears to be a useful subset of logic involving a concept of bounded quantification that could be clearly understood by a finite mechanism, since quantification over infinite sets would not be allowed. Even if the actual universe is infinite, the portion an intelligent agent will need to think about for most practical purposes will be finite. So, giving a machine the ability to handle formulas of this restricted predicate calculus may enable it to have an adequate understanding of logical operators, though not yet an understanding suitable for advanced mathematics. (A similar restriction may apply to many animals, young children and perhaps even most adults.)

Once quantifiers have been introduced in this way, it may be that the restriction to finite sets can be dropped, using a syntactic manoeuvre to extend semantic power. Unfortunately, the history of mathematical logic shows that there are deep problems of interpretation of logical symbols in infinite domains.

Anyhow, the claim is that by extending existing AI techniques we could enable a machine to employ a store of symbols using a logical notation to formulate instructions, questions or assertions concerning its ‘accessible’ world, and using inference rules that define the semantics of logical operators.

Implicitly defined non-logical symbols

Non-logical symbols can be interpreted as names or predicates via direct causal links, but this won’t do for reference to an inaccessible reality. Machines will need to refer to external objects, events, locations, etc. How can they use symbols to describe objects, properties, and relationships to which they have no direct access? Direct causal links via sensors and motors are often practically impossible, and may even be logically impossible, for instance when referring to hypothetical objects in hypothetical situations that never arise, as when I talk about the children my still-born elder sister might have had, or when a robot contemplates disasters it then manages to avoid. Though we cannot have causal links to events that never occur, or to non-existent individuals we can refer to them. How? The answer is implicit in our preceding discussion: using a logical formalism allows a set of ‘axioms’ implicitly to define semantics for non-logical symbols referring beyond what is immediately accessible. The ideas are familiar to philosophers studying theoretical concepts in science.

The basic idea is an old one: a collection of axioms for Euclidean geometry can partially and implicitly define predicates like ‘line’, ‘point’, or ‘intersects’, so that their possible interpretations are restricted. Tarski (1951) showed formally, in his recursive definition of what makes an assertion true, how some portion of reality can be or fail to be a model for a set of axioms (sentences in a logical formalism). Carnap (1956) suggested that, in a logical notation new symbols can be defined implicitly by ‘meaning postulates’. This was used to account for theoretical concepts of physics and dispositional concepts like ‘brittle’. Our machine could do something similar.

A model for an axiom set is not, as sometimes suggested, another symbolic structure denoting the world. Models cam be portions of the real world. (Hayes 1985 explains this very clearly.) More precisely, they are virtual systems *implemented* in the physical world just as a computer’s memory was shown above to be a virtual system implemented in the world. Put another way, a model may be seen as a bit of the world ‘carved up’ in a certain way, just as people using the concepts ‘finger’, ‘thumb’, ‘knuckle’ and ‘palm’ conceptually carve up human hands in one way, whereas an anatomist, or a dog licking its master’s hand, might use a different ontology.

Axioms alone do not pin down meanings unambiguously, since there are always different possible models. Any model for axioms of projective geometry has a dual in which points and lines are interchanged, for example. Axioms merely define the non-logical symbols to refer to aspects of the structure common to all the acceptable models. So meanings defined purely axiomatically have structure without content. They cannot be used to talk about specific properties of our world, or particular individuals, but only about possible types of worlds or world-fragments, though their features might happen to exist in our world. Adding new axioms can narrow the range of possible models, but will never pin the interpretation down to any actual portion of reality.

If, however, some of the symbols already have meanings determined independently of the axioms, then this will restrict the set of possible models, and thereby the possible interpretations of the new symbols. It may even attach the new symbols to a particular bit of the world. (See Nagel 1961, Pap 1963 on theoretical terms in science).

Semantic links between binary codes and objects or events in the machine's memory are based on causation not axioms. This semantic attachment to the actual world can be inherited by axiomatically defined terms, if the axioms link new and old terms. A theory that links unobservables, like 'neutrino', to observables, like 'flash' may limit possible models of the theory to things in this world. A blind person may attach meanings to colour words not too different from those of a sighted person, because much of the meaning resides in rich interconnections with concepts shared by both, such as 'surface', 'edge', 'pattern', 'stripe', etc. Likewise, a machine with logical powers could interpret symbols as formulating assertions or questions about things beyond its immediately accessible world, as follows.

Using logic to think about inaccessible objects

In addition to symbols referring to its memory locations and their contents and relationships, a machine could use a logical formalism to express axioms for predicates analogous to 'before', 'after', 'inside', 'outside', 'further', and and perhaps even 'cause'. (How this comes about does not matter for now: it could arise from random processes, external input or some kind of creative learning. I am trying only to explain how the machine could interpret such symbols.) The predicates could then be combined with names of locations to make assertions about the contents of or events in the machine's memory, which would form a Tarskian model for the database of axioms. The model could be a unique minimal model because some of the symbols had causal links with this portion of the world. Adding assertions using existential quantifiers would allow the formation of a new database for which the accessible world would not be a complete model. For example, an assertion might state that there exists a location 'beyond' the last known memory location. Such assertions would then be about inaccessible entities.

More formally, consider an existentially quantified assertion of the form:

(5) Ex(Rxa)

('There is an x such that x stands in the relation R to a', where a is a known object). If (5) is consistent with the rest of the database, but the machine can establish that no *known* object stands in the relation R to a, then it must express a hypothesis, or question, about some hitherto unknown entity. There could be a sub-set of the larger world containing the machine, that forms a Tarskian model for the extended database including (5).

Thus the machine uses symbols to formulate a proposition about something beyond its known world, without relying on any external causal links. It requires only some relations (e.g. spatial and temporal relations) linking internal entities to which it can already refer (using causal links) with external objects and events. 'Causes' is merely one of several internal/external relations able to support external reference.

The machine might have symbols for several such relationships defined partly axiomatically and partly by mechanisms for creating or checking internal instances of the relations. These symbols could be used to formulate collections of assertions for which the accessible world was not large enough to provide a model. The relations need not, but might, include a notion something like our concept 'cause', implicitly defined by a set of axioms, including axioms for the practical uses of causal relations. How exactly 'cause' should be defined is an old and unsolved problem. (A sophisticated machine might use the meta-level notion of a type of relationship whose detailed definition is not yet completely known, and build descriptions of relationships -- of unknown types -- between accessible objects and others -- also of unknown types. Cf Woods 1981)

Notice that I am only talking about the machine using symbols with a certain semantics. I am not discussing conditions for *successful* reference. I.e. the machine may use symbols that purport to refer, but don't actually, just as a deluded person can use a phrase like 'the burglar upstairs' even though it actually refers to nothing. The conditions for meaning, or understanding, are weaker than the conditions for knowledge (though more fundamental). Nothing is implied about the machine being *aware* of using symbols with a meaning. It seems that very young children and many animals use internal symbols without being aware of the fact. They lack the required self-monitoring.

The machine might think of its own internal states as embedded in a larger structure with a web of named relationships, and speculate about the properties of that structure, which it could refer to as: 'this world'. (It could do this even though some of its speculations were false, and not all important relationships are already known about.) External objects would then be referred to in terms of their supposed relationships to known internal objects. (This is an old idea in philosophy. cf. Strawson, 1959, ch.3). Such reference need not depend on sensors or motors providing causal links with the remote particulars. However, I am not, like philosophical phenomenalists, proposing that external objects be *defined* in terms of concepts relating to the internal world. This is not a reductionist theory.

It is sometimes argued that the semantics of empirical predicates must always be partly probabilistic, since often only probabilistic assertions are justified by available evidence. For instance, when only part of the extension of a predicate is accessible, statistical rules can be used to order, and perhaps assign probabilities to hypotheses compatible with available evidence. These rules, however, are simply heuristics for dealing with incomplete information and do not affect the semantics of the language used.

The indeterminacy of model-based semantics

The kind of semantics described here will always be indeterminate in that alternative models can exist. Like other relations, causal linkage via sensors, motors, or computer terminals used for purely verbal communication, can reduce, but never totally eliminate, semantic indeterminacy. It can narrow down reference to particulars, such as a particular place, or object or other agent, but such reference is never totally unambiguous in the way philosophers often dream of. There's always the possibility that some hidden complexity in the world prevents uniqueness of reference -- as when mischievous identical twins fool a teacher into thinking he knows who he is dealing with when he doesn't. Human and machine uses of symbols must be subject to exactly the same indeterminacy. Anyone designing machines to interact with people will need to take account of this fact.

New axioms can extend the ontology

This indeterminacy is an important aspect of the growth of knowledge, since it is always possible (except in very simple cases) to add new independent axioms that constrain the possible models and add precision to the implicitly defined terms. It is also generally possible to add axioms postulating both additional entities and new relations between those entities and the previous ones, just as science advances partly by postulating new sorts of entities: like atoms, neutrinos, genes, gravitational forces, and new relationships between them and familiar objects. This often adds coherence to disparate observations.

A similar process might occur in our machine -- and perhaps in a child. Some parts of the memory (e.g. a retina connected to external transducers) might be changed by external events. A machine that cannot think about an external world will be forced to treat these events simply as inexplicable occurrences. If the 'axioms' are (somehow) extended to refer to a suitably structured external environment including a process whereby structures are 'projected' into its memory (usually with considerable loss of information) it may be able to make sense of the phenomena, e.g. explaining 2-D retinal changes as resulting from different views of the same external 3-D scene.

The machine may also discover that certain changes that it can produce in parts of its memory (connected perhaps to motors) are followed by changes in its sensory registers. The relationship between the two sorts of changes may at first seem arbitrary and inexplicable, but by adding axioms describing suitable external structures and causal connections, the whole thing may be made to fit into a coherent framework. (A more complicated story is required if the system is to allow that its senses can sometimes malfunction and deceive it. Similarly scientific theories accommodate faulty instruments.)

A machine using an explicit set of axioms describing external (and internal) structures may be contrasted with one that merely uses perceptual and planning mechanisms that happen to be consistent with such a set of axioms (a 'compiled' version of the axioms, or a compiled theory about the world). This may be relevant to understanding differences between animal species. The latter system could not support some explicit learning processes, for example.

Loop-closing semantics for non-propositional symbols

So far the discussion has assumed that the machine uses a logical language to formulate axioms and record beliefs. Though little is known about the high level representations used by brains, it seems unlikely that birds, baboons or babies use explicit Carnapian meaning postulates or logic with Tarskian semantics to enable them to perceive and act on things in the world. Yet many animals appear to have rich mental lives including awareness of external objects. Might something other than logical and propositional representations explain this?

Perhaps a generalisation of Tarskian semantics is applicable to a wider range of intelligent systems. Not all internal representations have to be propositional, any more than our external representations are. There are good reasons for using a variety of forms of representations, including analogical representations such as diagrams, maps, ordered lists, networks, etc. Visual systems use some representations related to image structures. Many of these non-logical symbolisms can be implemented in computers. They can be used for a variety of purposes, including representing goals, percepts, beliefs, instructions, plans, and so on. (See Sloman 1985a).

If we think of such representations as having a semantics partly defined by their use in perception, planning, acting, etc., then the notion of a model might be defined as 'an environment which can coherently close the feedback loops'. Roughly, this requires the environment to be rich enough for external objects and events, to project (via perceptual mechanisms and action mechanisms) into internal representations of beliefs, goals and behaviour. The projection need not preserve structure (as neither geometric image-forming projections, nor Tarskian mappings do), but must support some notion of valid inference. That is, certain transformations of correct representations must be useful for making predictions, or drawing inferences about the environment. An environment that allows successful predictions to be made and goals to be achieved, and checked using perception, provides a model in this sense. This notion of semantics requires further investigation. Tarskian semantic theory takes a God-like perspective, contemplating mappings between symbols and things independently of how anything uses those mappings. This may not be possible in general, for instance, if semantic relations are highly context sensitive. We'd then need to adopt a design stance and think about the mechanisms.

The meaning attributed to a symbolic structure will be relative to the system's ability to have precise and detailed goals and beliefs. How specific the semantic mapping is will depend on how rich and varied is the range of percepts, goals and action strategies the system can cope with. An image representing a desired view of a scene may be constantly checked against current percepts as the machine moves. If the matching process requires very detailed correspondence between image and percept, the semantics will allow more different situations to be represented distinctly, than if matching is very tolerant. If different degrees of tolerance are required for different purposes, the semantics will be context sensitive.

Like Tarskian semantics, 'loop-closing' semantics leaves meanings indeterminate. For any level of specification at which a model can be found, there will be many consistent extensions to lower-levels of structure (in the way that modern physics extends the environment known to our ancestors). In both cases, as John McCarthy has pointed out in conversation, if the total system is rich enough, and works in enough practical situations, the chance that we've got it wildly wrong may be small enough to be negligible, except for sceptical philosophers.

Methodological note: do we need the design stance?

Why isn't it enough to describe a Turing test that a machine might pass? Because how behaviour is produced is important. Two systems producing the same behaviour over all the tests that can be dreamed up in a lifetime, might differ in how they would perform in some test not yet thought up. For example, there is a difference between a program with a generative ability to solve problems in arithmetic, and one that uses an enormous table of problems and solutions that happens to include answers to all the arithmetic questions that any human being will ever formulate. (Compare the distinction (Cohen [forthcoming]) between 'simulated understanding' and 'simulated parroting'.) Mere use of a look-up table would not constitute competence at arithmetic. Success in passing tests would be partly a matter of luck: the missing problems are never posed. Correct answers are produced simply because the entries happen to be in the table. Such a program can be described as successful but unreliable. It would not work in all the cases that could possibly arise. A table could not be checked except by examining every entry, whereas meta-level reasoning can be used to check a generative mechanism. So there are good engineering reasons for rejecting the Turing test as adequate.

A follower of Ryle (1949) might attempt to deal with all possible tests by postulating an infinite set of behavioural dispositions. We'd still need an explanation of the infinite capability in terms of a finite mechanism that can reliably generate the required behaviour. So the behaviourist analysis of mental states is unsatisfactory from an engineering point of view. From a naive philosophical point of view it is also unsatisfactory because it seems to leave something important out. Many people are convinced, on the basis of their own experience, that mental states like understanding have a kind of existence which is plain enough to those who have them, but which is quite unlike and independent of the existence of physical bodies or their behaviour. No amount of behaviour by a machine, however similar to human behaviour, could demonstrate the existence of this non-physical sort of state or entity. Dualists often admit that they lack conclusive evidence that other people have the same mental states as they do. But because of similarities of constitution and origin they are willing to give other people, or even other animals, the benefit of the doubt -- but not so computers.

Dualist objections to behaviourism assume that mental objects and events are non-physical entities directly perceivable only by introspection. So, unlike the postulated entities of theoretical physics, each person has direct and infallible access to a different subset. There is an element of mystery, since we have no explanation of how these entities can generate and control behaviour in physical systems. Nevertheless, it is a compelling view and is at least part of the motivation of many who object to the claim that computational mechanisms can explain the existence of mind. No amount or type of symbol manipulation could bring into existence new entities with the required properties.

There are reasons for the persistence of the dualist view. Like the theory that the sun and stars revolve around the earth, it is supported by common observations and also satisfies a powerful need to think of oneself as special. A theory thus motivated cannot be undermined simply by evidence and argument. Something analogous to therapy, or moral persuasion, is required, but there is no guarantee that the same techniques are relevant to everyone. Homo sapiens may not be unique in the space of possible intelligent machine-types, but each individual human being has a uniquely tangled web of reasons spawning motives and beliefs.

Since therapy cannot be conducted in a public essay, I have concentrated only on scientific and engineering considerations relevant to the design stance. A key issue is reliability: unless there is an underlying generative mechanism there is no reason to believe that intelligent behaviour will continue, no matter how many tests have already been passed. This is the key reason for rejecting the Turing test. Even common sense concepts, I believe, work on the assumption that explanatory mechanisms underly observed generalisations of all kinds, even when nobody knows what the mechanisms are, and even when there's a wide-spread confusion over what would be an adequate explanation.

Conclusion

This paper extends earlier suggestions about how it is possible in principle for a machine to use symbols *it* understands. Here I've concentrated on what I had previously called 'structural conditions' (Sloman 1985c). Analysis of 'functional conditions' for meaningful use of symbols would require a description of mechanisms for symbols representing the machine's own goals, desires, plans, preferences, policies, likes, dislikes, etc. (For an initial sketch see Sloman and Croucher 1981.) A surprising conclusion is that external causal connections are not needed to support reference to an external world, provided that there are internal causal links between symbols and the machine's innards. Of course, external links are needed for reducing indeterminacy and checking truth or falsity: a requirement for knowledge. But understanding meanings is a more fundamental ability, with fewer requirements.

Theoretical AI investigates what can (or might) occur, what should occur and what does occur. I have been concerned only to explore some possible designs for cognitive systems. What the best designs for practical purposes might be, and how biological systems actually work, remain open questions.

Acknowledgements

This work is supported by a Fellowship from the GEC Research Laboratories, and a grant from the Rennaisance Trust. I have profited from conversations with many colleagues, especially Bill Woods. Woods (1981) expresses, I believe, very similar views, using very different terminology.

BIBLIOGRAPHY

Carnap, R., *Meaning and Necessity* Phoenix Books 1956.

Cohen, L.J., 'Semantics and the computer metaphor' in R. Barcan Marcus, G.Dorn, P. Weingartner (eds) *Logic Methodology and Philosophy of Science VII,* Amsterdam: North-Holland, forthcoming. (Initially circulated in 1983)

Dennett, D.C., *Brainstorms,* Harvester Press 1978.

Evans, Gareth, *The Varieties of Reference,* Oxford University Press, 1982.

Fodor, J.A., *The Language of Thought* Harvester Press 1976.

Frege, G., *Translations from the philosophical writings,* ed. P. Geach and M. Black. Blackwell, 1960.

Hayes, P.J., 'The naive physics manifesto' in D. Michie (ed) *Expert Systems in the Microelectronic Age,* Edinburgh University Press, 1979.

Hayes, P.J. 'The secnd naive physics manifesto' in R.J.Brachman and H.J.Levesque (eds), *Readings in Knowledge Representation,* Morgan Kaufmann, 1985.

Hempel, C.G, 'The Empiricist Criterion of Meaning' in A.J. Ayer (Ed.) Logical Positivism, The Free Press, 1959. Originally in *Revue Int. de Philosophie,_Vol.4.* 1950.

Nagel, E. *The Structure of Science,* London, Routledge and Kegan Paul, 1961

Pap, A., *An Introduction to the Philosophy of Science* Eyre and Spottiswoode (Chapters 2-3). 1963.

Putnam, A., *Mind Language and Reality: Philosophical Papers Vol 2,* (chapters 11, 12, 13), Cambridge University Press, 1975.

Quine, W.V.O., 'Two Dogmas of Empiricism' in *From a Logical point of view* 1953.

Ryle, G. *The Concept of Mind,* Hutchinson, 1949.

Searle, J.R., 'Minds, Brains, and Programs', with commentaries by other authors and Searle's reply, in *The Behavioural and Brain Sciences* Vol 3 no 3, 417-457, 1980.

Searle, J.R., *Minds Brains and Science,* Reith Lectures, BBC publications, 1984

Sloman, A. and M. Croucher, 'Why robots will have emotions' in *Proc. IJCAI* Vancouver 1981.

Sloman, A., D. McDermott, W.A. Woods 'Panel Discussion: Under What conditions can a machine attribute meaning to symbols' *Proc 8th International Joint Conference on AI,* Karlsruhe, 1983.

Sloman, A., 'Why we need many knowledge representation formalisms', in *Research and Development in Expert Systems,* ed M. Bramer, Cambridge University Press, 1985. [1985a]

Sloman, A., 'Strong strong and weak strong AI', *AISB Quarterly,* 1985. [1985b]

Sloman, A., 'What enables a machine to understand', in *Proc 9th International Joint Conference on AI,* UCLA, 1985. [1985c]

Sloman, A., 'Did Searle attack strong strong or weak strong AI?', in A. Cohn and R. Thomas (eds) *Proceedings 1985 AISB Conference,* Warwick University, Forthcoming.

Strawson, P. F., *Individuals: An Essay in Descriptive Metaphysics,* Methuen. 1959.

Tarski, A., 'The concept of truth in formalized languages' in his *Logic Semantics Metamathematics,* New York, 1951.

Woods, W.A., 'Procedural semantics as a theory of meaning', in *Elements of discourse understanding* Ed. A. Joshi, B. Webber, I. Sag, Cambridge University Press, 1981.