Some Relations between Semantically Coupled Languages and Machines

Todd Davies
Artificial Intelligence Center
SRI International

ROUGH DRAFT: 29 April 1986

Introduction

The current century has been marked, in every discipline of human study, by a preoccupation with language: how we see the world through it, what its limits are, how it works and what it means. In artificial intelligence, we are concerned with making machines that seem to understand language in terms of the world that language describes, and which seem to understand the world in terms of language. In particular, the usual AI approach is to encode knowledge about the world in some sort of language, usually one with a serial structure that looks something like lisp or first order logic. This seems to work well for building machines that take as inputs expressions with a well understood syntax and semantics, such as commands and simple measurements have, but there is reason for some doubt about whether such languages can be be used to represent the knowledge required for intelligent processing of natural language and complex sensory input.

We may distinguish two basic categories of people who harbor this doubt. On the one hand, one can hold that our present vocabulary and programming language technology is theoretically adequate to describe all of the knowledge about the world that would be needed for intelligent perception, say, or natural language understanding, but that purely practical constraints on our ability to use this technology make it impossible for us to approach this theoretical limit. In this category would be included those who think that it would take too many programmers, or too much time, or too much processing power given our current capabilities, but that it may someday be possible, with advanced software tools and faster, more parallel machines, to program knowledgeable machines in something resembling a traditional language. The second, more extreme form of doubt one can have is the suspicion that *any* language which is constricted to the English vocabulary and which has the set of properties that characterize current programming languages (non-iconicity, serialness, context-freeness, closed-form semantics) must be fundamentally incapable of representing all the knowledge required for intelligent action. If this doubt were right, it would mean that such languages fail some criteria for adequacy in representing knowledge that, we see by our own example, is implicit in the time series of sense data coded in the brain. This paper is intended to give the second type of doubter a theoretical

framework for thinking about this problem. It is hoped that this may help to answer how the second type of doubter could, even possibly, be correct.

There is much work being done in AI under the assumption that neither of the above-mentioned doubts are correct, that on the contrary it is possible to build very intelligent-seeming and useful machines just by formalizing our knowledge in one of the standard languages. This is the view taken by Nilsson in @cite[Nilsson]. Opposed to this there is work ongoing in such areas as connectionism (@cite[CogSci]), visual languages (@cite[Raeder]), and perceptual coding (@cite[??]), much of it motivated by a belief in the inadequacy of the traditional programming methods for problems like perception and natural language processing. The odd thing is that the division between the two camps--the doubters and the non-doubters (@cite[Pentland])--which has at its center an important and substantive technical issue, seems mainly to be a matter of religious persuasion. Very little effort has gone into attacking the question head on from an unbiased, theoretical standpoint. It may be thought that the issue can only be resolved empirically, by having both camps work within their own perspectives until it becomes clear which approach is superior. This paper is an outgrowth of an attempt to discover whether the problem really is purely empirical, but there are no results to report at this point. There is only a taxonomy of possibilities, and some indications about how the questions which concern us may be answered by further study. The motivation for looking at this problem springs from a feeling that it ought to be possible to work out carefully the formal conditions that would have to be true if one side or the other were right in the debate, and that if we can discover these conditions, the methodological questions should become cleaner and easier to answer.

Let us briefly review some of the questions which traditionally crop up, in various forms, in the philosophy of language and of artificial intelligence methodologies, and at which this paper is aimed.

1. The actions of a deterministic machine may be viewed as functions on the machine's internal state. Therefore, if a programming language is to specify what outputs should result from which states, the language must be capable of expressing the conditions a state needs to satisfy in order to perform some action, and it must be capable of specifying each action the machine is designed to carry out. A problem would arise if the language chosen for programming were incapable of distinguishing these conditions and actions. For complex tasks in which the set of states which satisfy the condition for an action is large and difficult to define, one is driven to ask whether even ordinary English is adequate for the description task. But preliminary to answering this, we need to know more generally for a language what it would mean for it to be *in*adequate to this task. Let us call this problem that of *linguistic adequacy*.

2. Wittgenstein (@cite[Wittgenstein]) and Jackendoff (@cite[Jackendoff]) cite the problems involved in trying to give necessary and sufficient lexical conditions for the truth of linguistic expressions. If a closed-form, intensional semantics cannot be given

for a linguistic expression then any set of rules giving linguistic conditions under which an expression is true must be incomplete. If there are expressions for which this is true then some part of the meaning for such expressions must be non-linguistic, i.e. even though human beings can say whether an expression holds in any given circumstance, the language would, on this view, be incapable of stating the rule for determining whether it holds. A Supreme Court justice once said that he could not define obscenity, but he knew it when he saw it. Could this be literally true? How can an expression have meaning that cannot be stated in language? We might call this the problem of *linguistic definability.*

3. A question which does not seem to come up often, perhaps because it is confused with the problem of linguistic definability, is one that might be termed the issue of ***semantic bidirectionality.*** The problem arises in assessing the shared aspects of natural language understanding and natural language generation. In the former, natural language is an input to the machine, and in the latter it is an output of the machine. Now it might be assumed that the set of states in a machine to which an expression could give rise on input should be identical to the set of states which would result in the expression's emerging as output, but this might be a very unrealistic model of the conditions for understanding and generation of the same expression in human beings. On the contrary, it seems that what comes to mind when we hear or read an expression is not the full set of conditions which it could be used to describe. Neuroscience (@cite[Thompson]) and cognitive psychology (@cite[Bower]) support the possibility of this asymmetry because connections between synapses and some apparent associations revealed in memory experiments are unidirectional: a connection associating stop-sign with red does not necessitate a corresponding connection in the other direction, though there might be one. If this view of semantics is taken seriously, then it makes no sense to ask for a biconditional definition (linguistic or non-linguistic) for an expression because the expression's meaning must be split into what is understood by it and what it can describe.

4. It is sometimes argued that in order for communication to occur, the meaning of a linguistic expression must be the same for both parties in the communication. A pairwise induction argument can extend this to the claim that a linguistic expression has the same meaning for all speakers of a language. Yet it seems clear that we do not agree on the meanings of many expressions: two apparently competent users of the language often take opposite views on whether an expression applies in a given circumstance. One may hold that the expression has an objective meaning if it refers to a condition naturally distinct from others to which it does not refer, but the existence of these natural boundaries, especially in the domain of the abstract, spurs philosophical debate. When the expression has been defined to refer to a natural kind, then one may say that one or both of the speakers in a disagreement over its application is or are mistaken in what constitutes the natural kind, and that the expression really does have one proper meaning. If we allow for the possibility that an expression may have nonsingular, subjective meanings, we must give an account of how two agents can interpret an expression in different ways, express a condition in different ways, and yet communicate efficiently. Let us call this the problem of *semantic subjectivity.*

One approach to these questions is to try to cast them in a formal framework, in which the possibilities may be more precisely articulated. The following section sets forth such a framework with an attempt to motivate the selection of the particular model used in the analysis.

Theory

The analysis is given semi-formally. A model for a language-using machine is presented, relations that conditions of the machine can bear to expressions in its language are defined in terms of the model, and the consequent stucture of these relations is given. A case analysis in which the presence and absence of these relations is demonstrated appears in the section titled "Example" which follows this one.

Model

A machine (which might be any physical object) whose states are semantically related to expressions in a language can be modeled for our purposes as a structure

$M=<S,L,e,d>$,

where

1. S is a nonempty, mutually exclusive and collectively exhaustive set of *machine states*,
2. L is a nonempty set of all the possible *linguistic expressions* in a language,
3. $P(S)$ (the power set of S) is the set of all possible *conditions* (sets of *states*) of the machine,
4. $e:P(S)\rightarrow L$ (e is a partial *encoding function* that takes *conditions* into *expressions*), and
5. $d:L\rightarrow P(S)$ (d is a partial *decoding function* that takes *expressions* into *conditions*).

The structure may be elucidated as follows.

Machine States:

One can always define a set of states such that a machine is in exactly one of them at any given time. This does not mean that a set of states that usefully describe configurations of the machine is easy to come up with. In binary machines a useful set is given by the power set of its nodes, with a state being the set of nodes that are turned on. So it seems appropriate to assume the existence of a unique state set when the model is being applied to digital computers, but these are not the only types of machines to which the model may be applied. The brain may be idealized as a finite state machine (@cite[Hopfield]) with all neurons either on or off, but the state of a neuron is not a Boolean function of its inputs, so the analogy with binary machines is only partial even in the idealized case.

Machine Conditions:

A condition may be viewed as a predicate on the state of a machine, and the subset of S whose elements satisfy the predicate may be viewed as its extension. An n-bit machine, which therefore has 2^n states, can satisfy 2^{2^n}-1 conditions (it cannot fail to be in a state), which are the Boolean algebra of subsets of S, excluding the null set. In binary machines, the set of conditions is thus the power set of the power set of the set of nodes, excluding ϕ, assuming all node settings are possible. If one can enumerate a set of conditions that can occur in a machine, one need not define the set S since the functions d and e in the model map to and from the set of conditions rather than the set of states. For Moore and Mealy machines (@cite[Holcombe]), therefore, in which L is the input/output alphabet, the condition returned by d for an expression E is just the subset of the machine's states that are consistent with E as an input, and the expression returned by e for a condition C is the disjunction of the outputs for all states in C.

Linguistic Expressions:

The language is modeled as a set of "expressions". This is deliberately vague, for one may want to discuss the semantics of words, or propositions, or clauses, or something else. Normally, L will be a set of formulas, so that, in an object language defined under propositional logic or predicate logic in conjunctive normal form, L would be the power set of the power set of the set of atomic propositions. One need not necessarily view L as a set of inputs and outputs for the machine; it is just the range of expressions for *encoding* machine conditions, and the domain of expressions *decoded by* machine conditions. It is sometimes said that propositions can be "semantically attached" to states in a machine: here, we are just distinguishing between the two attachment relations of encoding and decoding.

The Encoding Function:

The partial function e assigns to every machine condition either an expression in L or nothing. It may assign the same expression to more than one condition. Since the power set grows very quickly as the number of possible machine states increases, for machines with more than a few bits there will usually be many conditions that are not named by any expression. This is not true in a first order language in which e is defined for every state in S, since then each condition can be expressed as a disjunction of the expressions which name its states. But we can convince ourselves that in a machine with $2^{256,000}$ states, it will not be the case that e is defined for each state, let alone for every condition, given any L yet devised for communication. In fact, it looks as if the conditions for which e is defined will, in most practical cases, be a very sparse subset of $P(S)$. If the conditions that need to be tested for intelligent action are much more numerous than the set for which e is defined then the expressions in L will not be adequate for programming intelligent machines.

The Decoding Function:

The partial function d assigns to every linguistic expression either a condition in $P(S)$ or nothing. If d is undefined for an expression E in L, this just says that E is not semantically decodable as a condition in the machine. For synonymous expressions d returns the same condition. Expressions with no synonyms cannot be defined linguistically. The condition returned for E by the decoding function might not have E as its encoding function assignment. This leaves open the possibility of semantic asymmetry as mentioned in the previous discussion (see "Introduction") of the semantic bidirectionality problem. Intuitively, the decoding of an expression in the brain may be thought of as the mental condition which it causes, as distinct from the mental condition or collection of them which brings to mind the expression, or which is encoded by the expression.

When the expressions and conditions are assumed to have this causal or input/output relationship, the model will have the property that the conditions (resp. expressions) assigned in the decoding (resp. encoding) for a given expression E (resp. condition C) will be subsets of (resp. will entail) the decodings (resp. encodings) for any expressions E' (resp. conditions C') that are entailed by (resp. supersets of) E (resp. C). The model does not require this in general, however, as the interpretation of the decoding (resp. encoding) functions is not restricted to that of processing input to (resp. generating output from) an automaton, but can also be thought of as just a description of the condition constructed from (resp. expression recognized as) the expression (resp. condition).

Definitions

The model thus described, it is possible to define a set of properties that expressions and conditions can have with respect to each other and with respect to different agents. If we allow each agent to be characterized by a different model with the structure of M, then the sets and functions of the model will need to be indexed by agent or machine. For this paper, it is assumed that the sets L and S are the same for all agents, but that the partial functions e and d depend on the agent. This amounts to the assumption that agents share a language and a set of possible experiences, but differ in how they interpret the language and express their experiences. Allowance can be made for differences in language and possible states if one wants to be more general, by amending the arguments made here in way that should be obvious.

Some relations that can be defined with respect to the model are the following.

1. A condition C in $P(S)$ is *directly encodable* in L iff there exists an expression E in L such that $e(\mathrm{C}) = \mathrm{E}$.

2. C is ***partially encodable*** in L iff there exists a condition C' such that (1) $\mathrm{C} \subseteq \mathrm{C}'$, and (2) C' is directly encodable in L.

3. C is *linguistically distinguishable* in L iff C is directly encodable in L and there exists no other condition C' such that $e(C) = e(C')$.

4. C is *linguistically invertible* in terms of L iff $d(e(C)) = C$.

5. C is *identically encoded* by two agents $<S,L,e_1,d_1>$ and $<S,L,e_2,d_2>$ iff $e_1(C) = e_2(C)$.

6. An expression E in L is *directly decodable* in S iff there exists a condition C in *P(S)* such that $d(E) = C$.

7. E is *partially decodable* in S iff there exists an expression E' such that (1) $E \Rightarrow E'$, and (2) E' is directly decodable in S.

8. E is *mechanically distinguishable* in S iff E is directly decodable in S and there exists no other expression E' such that $d(E) = d(E')$.

9. E is *mechanically invertible* in terms of S iff $e(d(E)) = E$.

10. E is *identically decoded* by two agents $<S,L,e_1,d_1>$ and $<S,L,e_2,d_2>$ iff $d_1(E) = d_2(E)$.

Consequences

It can now be shown how the above definitions relate to each other. The properties of invertibility, distinguishability, direct en/de-codability, and partial en/de-codability form partial inheritance hierarchies of conditions and expressions. In addition, other theorems fall out of the definitions which can help to determine the place of a condition or expression within the hierarchies.

Taxonomy for Conditions:

The possible categories for a condition C with respect to L may be summarized as follows.

1. C can fail to be partially encodable in L.

2. C can be partially encodable in L without being directly encodable in L, but the reverse is not possible.

3. C can be directly encodable in L without being linguistically distinguishable in L, but the reverse is not possible.

4. C can be directly encodable in L without being linguistically invertible in terms of L, but the reverse is not possible.

5. C can be linguistically distinguishable in L without being linguistically invertible in terms of L.

6. C can be linguistically invertible in terms of L without being linguistically distinguishable in L.

7. C can be both linguistically distinguishable in L and linguistically invertible in terms of L.

Figure 1 summarizes the taxonomy of properties for conditions.

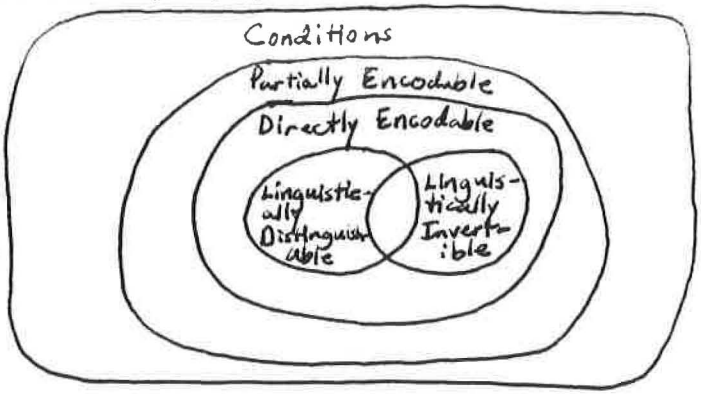

Figure 1: Categories of Machine Conditions

The possibilities may be elucidated as follows. Conditions in the first category above (not partially encodable) are those for which nothing in the language can be said: there is not even a more general condition which is directly encodable and is entailed by the first condition. This possibility is avoided as long as $e(S)$, the encoding of the most general condition, is defined, e.g. when "True" is in L and $e(S) =$ True. Category 2 consists of conditions that do not have their own encodings, but which imply more general conditions that can themselves be encoded. Conditions in the third category have their own encodings but, since this encoding is not unique among all the conditions, there is no linguistic means for encoding the difference between such a condition and another one which is encoded identically. In the fourth category, a condition has a representation in L, but the encoding of the condition is not identical to an expression which is decoded as that condition; hence it is not linguistically invertible. Conditions in category 5 can be encoded distinctly from all other conditions but are not the same as the conditions given rise to by their encodings. Category 6 is the opposite of category 5. Category 7 represents conditions that are distinctly encodable as well as being the decodings of their encoding expressions.

Taxonomy for Expressions:

The possible categories for an expression E with respect to S may be summarized as follows.

1. E can fail to be partially decodable in S.

2. E can be partially decodable in S without being directly decodable in S, but the reverse is not possible.

3. E can be directly decodable in S without being mechanically distinguishable in S, but the reverse is not possible.

4. E can be directly decodable in S without being mechanically invertible in terms of S, but the reverse is not possible.

5. E can be mechanically distinguishable in S without being mechanically invertible in terms of S.

6. E can be mechanically invertible in terms of S without being mechanically distinguishable in S.

7. E can be both mechanically distinguishable in S and mechanically invertible in terms of S.

Figure 2 summarizes the taxonomy of types of expressions.

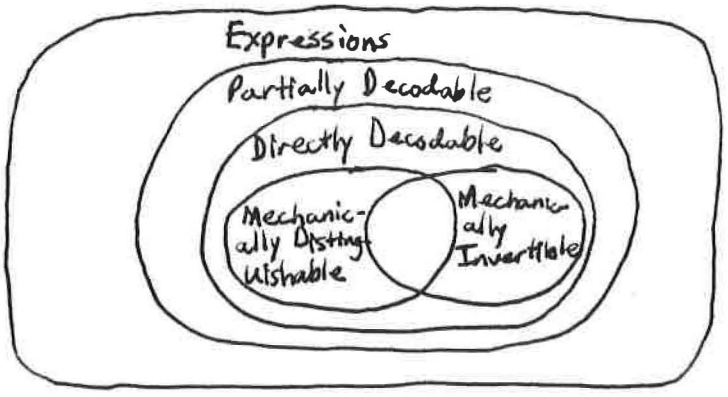

Figure 2: Categories of Linguistic Expressions

The possibilities may be elucidated as follows. Expressions in the first category are those which do not constrain the set of possible states in any way. If "True" is in L and is decoded as S then all expressions are at least partially decodable assuming E $\Rightarrow$ True for all E in L. The second category picks out expressions that do not themselves correspond to a particular condition but which entail expressions that do correspond to one. Category 3 represents expressions that have a particular condition assigned to them but which are not unique in the condition they pick out and are therefore synonymous for decoding purposes with at least one other expression in L. In the fourth category are expressions that pick out a particular condition, but are not themselves the encoding for that condition. In category 5 are expressions that pick out a condition that cannot be decoded as any other expression, but which are not identical to the expression given rise to by their decoding. Category 6 is the opposite of category 5, and category 7 represents expressions whose decoding is unique and has, itself, an encoding equal to the expression.

Meanings for Different Agents:

The theory of the possibilities for the relation between one machine's semantics and another's is not as well developed as the theory for single models. Essentially, we have distinguished two possibilities each for conditions and expressions when either is shared by more than one agent. In the case of conditions, the agents may either encode them identically or not, and in the case of expressions, the agents may or may not decode them identically. It seems implausible that

communication between agents would require identical encoding and decoding functions for all of them, but it is not clear what dissimilarities between the functions would stifle communication. Agents can have theories about what other agents mean when they use an expression, but these must be built up by observation of the states in which the other agent uses the expression (@cite[Davidson]).

If an expression, for one agent, encodes or is decoded by a condition that holds in states which do not give rise to (or are not given rise to by) that expression in another agent, then the agents disagree on the application of the expression, and there may or may not be enough common ground between them to communicate in general. This area of the theory is clearly in need of further work, but we have at least established, it seems, a way to state the formal conditions on semantic disagreements.

Other Consequences:

A set of theorems provable from the definitions follows. The list is not intended to be complete in any sense.

1. If C is linguistically invertible in terms of L then e(C) is mechanically invertible in terms of S. (??)

2. If E is mechanically invertible in terms of S then d(E) is linguistically invertible in terms of L. (??)

3. More ... (??)

The properties defined and elucidated in the theory can now be put to work in the analysis of an example.

Example

For demonstrating the possibilities we choose a model consisting of a machine with only four states (a two-bit machine), and a language that is propositional logic parameterized by a set of atoms. In particular, we will view the states of the machine as representing the four suits of cards: Club, Diamond, Heart, Spade. We first define the machine, the class of languages, and some example encoding and decoding functions, and then use these in the presentation of each example case.

Machine

The machine (which is the same in each case presented below) consists of a set of nodes $\{n_1,n_2\}$. The set S of states is thus the power set $\{\phi,\{n_1\},\{n_2\},\{n_1,n_2\}\}$, which we may rewrite as $\{1,2,3,4\}$. The set of conditions is the set of all subsets of S, excluding $\{\}$.

Language

Each example case will have defined for it a set A of atomic propositions, from which the set L may be constructed as follows:

1. If E is in A then E is in L.

2. If E and E' are each in L then so are (E), (E'), $\neg$E, $\neg$E', E $\wedge$ E', E $\vee$ E', and E $\Rightarrow$ E'.

3. Nothing else is in L.

For instance, the following set A might be defined for the machine described above:

{Club,Diamond,Heart,Spade,Red,Black,True,False}.

Functions

The encoding and decoding functions e and d are defined separately for each case. The encoding (resp. decoding) function in each case will assign at most one expression (resp. condition) to each condition (resp. expression). For instance, if the set A is as given above under "Language", and the machine is the one defined for the example under "Machine", the encoding function might be defined as follows:

$e(\{1\})$ = Club $\wedge$ $\neg$Diamond $\wedge$ $\neg$Heart $\wedge$ $\neg$Spade $\wedge$ $\neg$Red $\wedge$ Black $\wedge$ True $\wedge$ $\neg$False
$e(\{2\})$ = $\neg$Club $\wedge$ Diamond $\wedge$ $\neg$Heart $\wedge$ $\neg$Spade $\wedge$ Red $\wedge$ $\neg$Black $\wedge$ True $\wedge$ $\neg$False

.
.
.

$e(\{1,2\})$ = (Club $\vee$ Diamond) $\wedge$ $\neg$Heart $\wedge$ $\neg$Spade $\wedge$ (Red $\vee$ Black) $\wedge$ True $\wedge$ $\neg$False

.
.
.

$e(\{1,2,3,4\})$ = (Club $\vee$ Diamond $\vee$ Heart $\vee$ Spade) $\wedge$ (Red $\vee$ Black) $\wedge$ True $\wedge$ $\neg$False.

The decoding function might be defined as follows:

d(Club) = {1}
d(Diamond) = {2}
d(Heart) = {3}
d(Spade) = {4}
d(Red) = {2,3}
d(Black) = {1,4}
d(True) = {1,2,3,4}
d(Club $\vee$ Diamond) = {1,2}

.
.
. etc.

with d(False) undefined. Note that L is an infinite set, so d cannot be finitely enumerated.

Cases

An instance of M for each of the categories defined in the taxonomies under "Consequences" can now be given in terms of this example.

Conditions:

1. Suppose A = {Club,Diamond,Heart}, and e is undefined for {4}, {1,4}, {2,4}, {3,4}, {1,2,4}, {1,3,4}, {2,3,4}, and {1,2,3,4}. Then {4} and all of its supersets are *not partially encodable.*

2. Suppose A = {Red,Black}, and e is undefined for {1}, but $e(\{1,4\})$ = Black. Then {1} is *partially encodable but not directly encodable.*

3. Suppose A = {Red,Black}, $e(\{1\})$ = Black, and $e(\{4\})$ = Black. Then {1} and {4} are both *directly encodable but not linguistically distinguishable.*

4. Suppose A = {Red,Black}, $e(\{1\})$ = Black, and d(Black) = {1,4}. Then {1} is ***directly encodable but not linguistically invertible.***

5. Suppose A = {Red,Black}, $e(\{1,4\})$ = Black, e^{-1}(Black) = {{1,4}}, and d(Black) = {1}. Then {1,4} is ***linguistically distinguishable but not linguistically invertible.***

6. Suppose A = {Red,Black}, $e(\{1\})$ = Black, $e(\{1,4\})$ = Black, and d(Black) = {1,4}. Then {1,4} is *linguistically invertible but not linguistically distinguishable.*

7. Suppose A = {Red,Black}, $e(\{1,4\})$ = Black, e^{-1}(Black) = {{1,4}}, and d(Black) = {1,4}. Then {1,4} is ***linguistically distinguishable and linguistically invertible.***

Expressions:

1. Suppose A = {Raining}, and for E equivalent to Raining $\vee$ $\neg$Raining, d(E) is undefined. Then Raining is *not partially decodable.*

2. Suppose A = {Raining,True}, d(True) = {1,2,3,4}, d(Raining) is undefined, and Raining entails True. Then Raining is ***partially decodable but not directly decodable.***

3. Suppose A = {Club,Spade,Black}, d(Club $\vee$ Spade) = {1,4}, d(Black) = {1,4}. Then Black and Club $\vee$ Spade are both *directly decodable but not mechanically distinguishable.*

4. Suppose A = {Club,Black}, d(Black) = {1}, and $e(\{1\})$ = Club. Then Black is ***directly decodable but not mechanically invertible.***

5. Suppose A = {Club,Black}, d(Club) = {1}, $d^{-1}(\{1\})$ = {Club}, and $e(\{1\})$ = Black. Then Club is *mechanically distinguishable but not mechanically invertible.*

6. Suppose A = {Club,Spade,Black}, d(Black $\wedge$ $\neg$Club) = {4}, d(Spade) = {4}, and $e(\{4\})$ = Spade. Then Black $\wedge$ $\neg$Club is *mechanically invertible but not mechanically distinguishable.*

7. Suppose A = {Club,Black}, d(Club) = {1}, d^{-1}({1}) = {Club}, and e({1}) = Club. Then Club is *mechanically invertible and mechanically distinguishable.*

As the above example cases make clear, to determine whether a condition (resp. expression) is partially encodable (resp. decodable) or not, one must know the lattice structure of the conditions (resp. expressions). All of the other properties can be determined without knowing this, however.

Conclusions

What does this analysis tell us? It doesn't seem to tell us very much about what must be the case, but it gives us a theoretical framework for talking about what may be the case, and makes it clearer, perhaps, why the issues mentioned earlier have not been resolved. Let us return to the four problems enumerated in the Introduction, and try to look at them in terms of the theory just presented.

Adequacy

The first problem discussed was that of linguistic adequacy. It concerned the question of whether a particular class of languages (e.g., applicative languages @cite[Sloman]) are capable of encoding enough of the machine's conditions to give the appearance of intelligence on the part of machines programmed with those languages. We may now cast this problem in terms of the theory. Linguistic adequacy, it seems, may be characterized at more than one level. A language may be adequate for directly encoding enough conditions without being adequate to distinguish among enough of them, or to encode them in a way that is linguistically invertible. This question can be asked with respect to practical, human limits as well as theoretical limits for a particular language.

The theory suggests that there is a companion problem to that of linguistic adequacy, namely the problem of the *mechanical* adequacy of particular machines with respect to a given language. Symmetric with the questions about languages, we may ask of a *machine*, for a particular set of expressions we are interested in tracking, whether the machine's states are capable of decoding enough of those expressions, distinguishing their meanings, and decoding them in a way that is mechanically invertible.

Definability

We can, it seems, now answer how an expression can have meaning that cannot be stated in the language from which the expression came. If the expression is mechanically distinguishable by a machine then it is undefinable in terms of the language. Yet, such an expression has a meaning, namely the condition that is its decoding. Even if an expression is directly decodable and *not* mechanically distinguishable, this does not guarantee that its full meaning may be defined linguistically, for it may still be that the definition is not the encoding for the same condition as that for which the expression is an encoding. Linguistic Definability must therefore be given in

terms both of decoding and encoding. An expression may be linguistically definable for purposes of one only, or both.

It seems plausible to conjecture that expressions in English can at least sometimes fail to be linguistically definable. The alternative is that English is a closed system, in which everything that is genuinely meaningful can be defined in terms of everything else. This work has not conclusively established that this is not the case, but there does not seem to be any reason to believe that it is the case, and the ease with which languages which do not have this property can be constructed makes the possibility that English is not a closed system easy to imagine.

Biderectionality

The theory splits the problem of semantic biderectionality into two types. There is invertibility for conditions and for expressions. Both are with respect to a given agent. By building in separate functions for encoding and decoding, we have allowed for the possibility that asymmetry in these two forms of meaning is a superior model of human understanding to one in which all conditions and expressions are semantically invertible. If it is agreed that one need not (even *should* not) discover all the possible conditions that could generate (be encoded as) a particular expression in order to understand (decode) uses of that expression then the burden for natural language understanding may be lessened. This may be a point that has been made often in the past, but it seems to get forgotten on occasion. Asking what is meant by (i.e., how one can linguistically encode the decoding of) an expression need not entail asking what is the set of all possible states which that expression encodes.

Subjectivity

The last problem mentioned in the Introduction, that of semantic subjectivity, is the one about which the analysis thus far has had the least to say. It has been included only because the model developed for analysis seems like a reasonable framework within which to look at this problem, as well as at the ones about meaning for single agents. The model makes clear how the meanings of a given expression or condition for two agents can be arbitrarily similar without being identical. The decodings of an expression, for instance, may be two conditions (one for each agent) whose intersection is smaller than the set of states in each's decoding, and if those states in which the interpretations differ arise seldom in relation to those in the intersection, then it seems plausible that communication can occur very smoothly despite the disagreement. If, on the other hand, the expression is often used in discourse situations in which one or more agents is not in a state in the intersection then misunderstandings will take place with corresponding frequency.

Possibilities

In the absence of sound arguments that eliminate the problems we have been considering, it looks like we cannot discount some possibilities that may be troublesome for artificial intelligence. Firstly, there is the possibility, for any given language (resp. machine) with which we work, that it will fail to be adequate for programming intelligent behavior because of its own limitations for encoding (resp. decoding) the conditions (resp. expressions) in which we are interested. Secondly, it is possible that text understanding systems written in a given language may be intrinsically limited in terms of the intelligence they can exhibit. Thirdly, it may be impossible to discover objective, interpersonal truth conditions for linguistic expressions. Fourthly, it may be impossible to discover necessary and sufficient truth conditions (even for a single agent) for linguisitic expressions due to semantic asymmetry. And finally, theories of intelligent behavior expressed in a given language may fail to capture human knowledge about even commonsense pscyhology because it might be impossible to express this knowledge in that language, or because the meaning of the language for mentalistic terms is too subjective.

Challenges

So what can we do about these problems? The following recommendations seem reasonable.

1. Before beginning any task to build an intelligent system, we should analyze the language and the machine we intend to use in a rough way in terms of the model presented in this paper, or a better one, asking ourselves which relations of representability seem likely to be satisfied by the language and the machine. It may be possible to work on small portions of a domain in order to get a feel for the adequacy of a language or machine, by counting states or some method of analysis prior to the full scale project. We should be convinced that the project has a good chance of working even after seriously considering the possible limitations of the language or machine. Such a strategy seems preferable to that of failing to consider the possible limitations at all.

2. We should begin with an openness to many language types and architectures, especially those for which the motivation seems to be to overcome the limitations taxonomized in this paper. Work on connectionism, on visual languages, and on learning seems to be motivated by the tedious nature of defining knowledge in traditional programming languages, while work on parallel architectures and denser and faster computers seems motivated by limitations of current machines. Efforts should be made to secure a technology powerful enough to accomplish the desired task.

3. We should avoid pausing so long to consider the options that empirical investigation fails to take place when it is the only good way to discover the limits of a given approach. The benefit of work like that in this paper, if there is any, is just to remind us that the goal may not be even theoretically possible to achieve, depending on what that goal is. Using tools that one seriously suspects will be limited in what they can achieve is perfectly okay if what one expects is within those limits.

These last two points are rather obvious, and are listed here only to give a more complete view of a strategy that might emerge if this analysis is taken to be worthwhile. The ruminations reported in this paper have not been without benefit to the me, as they have been a chance to think about some foundational issues on the way to something more concrete.