American Journal of Computational Linguistics *Microfiche 80*

INFERENCE AND THEORY

DAVID L. WALTZ, EDITOR

Coordinated Science Laboratory
University of Illinois
Urbana 61801

Papers presented in two sessions of TINLAP-2, the 1978 Meeting of the Association for Computational Linguistics, held with joint sponsorship by the Association for Computing Machinery and its Special Interest Group in Artificial Intelligence.

TABLE OF CONTENTS

A NOTE ON PARTIAL MATCH OF DESCRIPTIONS: CAN ONE SIMULTANEOUSLY QUESTION (RETRIEVE) AND INFORM (UPDATE)?[1]

Aravind K. Joshi
Department of Computer and Information Science
The Moore School, University of Pennsylvania
Philadelphia, Pa. 19104

Summary: In data base query systems there is an implicit assumption that descriptions in queries must match exactly, i.e., queries are for retrieval only, and not for retrieval and updating simultaneously. A related assumption (or constraint) that in questions descriptions are used referentially only (i.e., a question cannot be used simultaneously for questioning and informing) seems to hold in ordinary conversations also, with some qualifications. Some issues related to the validity of such a constraint and its relation to partial matching of descriptions are briefly discussed in this note.

1. In a question-answer system each description in a query is used referentially i.e., for each description one expects to find an entity in the data base which serves as the unique referent for that description. For simplicity, hereafter we will consider only definite descriptions (in particular, definite noun phrases consisting of a definite article, an adjective, and a noun). Thus in (1)

(1) Is the red book on the table?

the description the red book will serve to identify an entity, say, e_1 in the data base[2] and the description the table, an entity, say, e_2. The question can be answered after verifying the appropriate relation between e_1 and e_2. For the purpose of making the definiteness transparent and also for simplifying the discussion in this note, let us assume that there is exactly one book and one table in the data base.

2. The match for the red book can succeed if e_1 has a color attribute with the value red. The match can fail either due to a mismatch or a partial match. A mismatch will occur if e_1 has a color value other than red, say green. A partial match will occur if e_1 has an unspecified value for the color attribute or if the possession of the color attribute itself has not been specified for e_1.

In the rest of the discussion, we will not be concerned with failure due to mismatch, although many of the issues raised below are quite relevant to this case also. We will be concerned with partial matches only. A partial match really is a partially successful match, where a part of the description has matched exactly, and the remainder has failed to match due to the lack of some information, and not due to a mismatch.

3. Let us consider the case of a partial match where the part of the description that matched is sufficient to identify the referent uniquely. In (2) this is trivially accomplished because of our assumption that there is exactly one book and one table in the data base.[3] Although we have a partial match (due to the lack of the color value or the color attribute itself for e_1), it will be possible to answer the question either by yes or no depending on whether e_1 is on e_2 or not, since the referents e_1 and e_2 have been uniquely identified. How should we proceed in this case?

1. If we insist that each description in the question must match exactly, then clearly, we have failed to establish a referent and the question cannot be answered.

2. On the other hand, we may assume that whenever we have a partial match and the referents are uniquely identified somehow, we should answer the question, and treat that part of the description which was not accounted for as new information. This new information can then be used to update the data base. Thus for the question (2), if the partial match is due to the fact that in the data base the value for the color attribute for e_1 is not specified, then we can now specify it to be red. If, on the other hand, the partial match was due to the fact that the possession of the color attribute itself is not specified for e_1, then the updating would involve adding a new attribute called color for e_1, and then specifying a value for it, which in is this case is red. The first type of update can be called content update and the second type, structure update; in the first case we have made a local modification of assigning a value to an attribute, while in the second case a new structural item has been added.[4]

4. There are a number of issues involved in adopting a strategy for updating upon a partial match when the matched part uniquely identifies the referent. We will state only two of these issues here and pursue the second in some detail.

a) The part of the description that was missing in the data base (and which led to a

partial match) is accepted as new information and used for updating. The strategy followed is that if an exact match fails due to the lack of some information then the missing information is treated as new and updating is done accordingly. This is a kind of default reasoning.[5] However, it is not clear whether we can allow such unconstrained updates. In data base query systems there is an implici assumption that the descriptions in queries must match exactly, i.e., queries are for retrieval only[6] and not for retrieval and updating simultaneously. Can we relax this requirement somewhat? We can get some ideas by looking at questions in ordinary conversations, which is what we will do briefly in b) below.

b) The hypothesis (or constraint) that in a question construct[7] definite descriptions are used referentially only (i.e., a question cannot be used simultaneously for asking a question and conveying some additional information) seems to hold in ordinary conversations also, with some qualifications. The three examples below briefly describe some of the problems involved.

1) Suppose that 1) there is only one individual in the context, 2) the speaker believes that he is a plumber, 3) the hearer is unaware of his being a plumber, and 4) the speaker believes that the hearer is unaware of his being a plumber. Under such circumstances it would be inappropriate to use (3) to aske the question (4), and simultaneously inform the hearer that (5).

(3) When did the plumber leave?
(4) When did the person leave?
(5) He is a plumber.

If (3) is used by the speaker (possibly due to a mistaken belief that the hearer is aware that the person is a plumber), it is unlikely that the hearer will update his model without some clarification or some response such as Oh! I didn't know that he was a plumber, i.e., the hearer will not update without any interrupting responses. This example illustrates that the question construct cannot be used for questioning and informing simultaneously, and if it appears to have been so used (due to the speaker's ignorance of the hearer's lack of some information), the updating by hearer is not without an interrupting response, thus indirectly confirming the hypothesis.

2) Again suppose that 1) there is only one individual in the context, 2) the speaker regards him as a grouch, 3) the hearer has no such specific evaluation of him, and 4) the speaker believes that the hearer has no such evaluation. In this case, it seems not completely inappropriate for the speaker to use (6), in order to ask the question (7), and simultaneously inform the user that the speaker regards (8) to be the case.

(6) When did the grouch leave?
(7) When did the person leave?
(8) He is a grouch.

With evaluative information; simultaneously questioning and informing appears to be a bit more convenient. If (6) is used by the speaker, it appears that the hearer can update his model, without any interrupting responses, with the attribute grouchy attached to the entity, as speaker's evaluation (and the hearer's too if he agrees with the speaker). Even if the hearer asks for clarification, it is likely to be of the form Oh! I didn't know that you thought he was a grouch rather than Oh! I didn't know that he was a grouch (compare this to the response in the previous example).

3) Finally, there is an apparent violation of the hypothesis in examples such as (9).

(9) Who is sitting to the right of your lovely wife?

(9) can be used by the speaker to ask the question and pay a compliment (a side effect) rather than to convey new information. Thus the hypothesis does not appear to be violated in these cases.

5. Some of the issues which merit further investigation are as follows. 1) To what extent the hypothesis can be violated and what are the side effects. If the constraint is mutually understood by the speaker and the hearer, then any apparent violation of it will be recognized and may be accompanied by a side effect (implicature?) in addition to the updating. 2) To what extent updating without interrupting responses depends on the shape of the description, the syntactic construct in which it appears (e.g., questions, it-clefts, declaratives, etc.)[8], the role it plays in the construct (e.g., subject, topic, etc.), the discourse model (for the speaker and for the hearer) created so far, [9] etc. 3) To what extent the 'new' information used for updating has to be somehow relevant to the 'old' information, either by being inferrable from it or by being able to fit it into the discourse structure created so far, etc.[10]

Notes:

1. This work is partially supported by NSF Grant MCS76-19466. I wish to thank Jerry Kaplan, Lorrie Levin, Stan Rosenschein, Ivan Sag, and Bonnie Webber for valuable discussions.

Some of the issues raised here will be discussed in detail in a forthcoming paper by Joshi and Rosenschein (Strategies for reference and ascription in object centered representations).

2. We will assume a rather simple-minded structure for the data base. It will consist of entities and attributes, and relations among entities.

3. However, in general, unique reference may be established due to the context, and the structure and content of the data base.

4. In the data base context, updates are usually content updates. Structure updates are not

permitted. In a conversational context and discourse understanding, clearly, both types of updates are possible. In these contexts it is not clear whether we can always tell which type of update has taken place. Structure updates should be harder than context updates, cognitively speaking, but this is only a conjecture at this time.

5. See "On reasoning by default" by Raymond Reiter (this volume). The closed world assumption discussed in this paper is also relevant to our discussion. See also "Fragments of a theory of human plausible reasoning" by Allan Collins (this volume), and "Inferencing on partial information" by Aravind K. Joshi, in Pattern Directed Inference (ed. F. Hays-Roth and D. Waterman), Academic Press. 1978.

6. See "Cooperative responses from a natural language data base query system: Preliminary report", by S. Jerrold Kaplan, Technical Report, Department of Computer and Information Science, University of Pennsylvania, November 1977.

7. We will limit outselves only to wh questions and yes/no questions.

8. Lorrie Levin has made a perliminary investigation of the update potential of some of these constructs (unpublished).

9. Entity-oriented discourse models have been considered for problems of reference (see "A formal approach to discourse anaphora" by Bonnie Webber, Ph.D. Dissertation, Harvard University, 1978).

10. A detailed discussion of some of these issues will be included in a forthcoming paper by Joshi and Rosenschein (see note 1).

WITH A SPOON IN HAND THIS MUST BE THE EATING FRAME

Eugene Charniak
Department of Computer Science
Yale University

ABSTRACT

A language comprehension program using "frames" "scripts", etc. must be able to decide which frames are appropriate to the text. Often there will be explicit indication ("Fred was playing tennis" suggests the TENNIS frame) but it is not always so easy.("The woman waved while the man on the stage sawed her in half" suggests MAGICIAN but how?) This paper will examine how a program might go about determining the appropriate frame in such cases. At a sufficiently vague level the model presented here will resemble that of Minsky (1975) in it's assumption that one usually has available one or more context frames. Hence one only needs worry if information comes in which does not fit them. As opposed to Minsky however the suggestions for new context frames will not come from the old ones, but rather from the conflicting information. The problem them becomes how potential frames are indexed under the information which "suggests" them.

1 INTRODUCTION

Understanding every day discourse requires making inferences from a very large base of common sense knowledge. To avoid death by combinatorial explosion our computer must be able to access the knowledge it needs without irrelevant knowledge getting in its way. A plausible constraint on the knowledge we might use at a given point in a story or conversation (I shall henceforth simply assume we are dealing with a story) is to restrict consideration to that portion of our knowledge which is "about" things which have been mentioned in the discourse. So if we have a story which mentions trains and train stations, we will not use our knowledge of, say, circuses. This requires, of course, that given a topic, such as trains, or eating, we must be able to access its knowledge without going through everything we know. Hence we are lead in a natural way to something approaching a notion of "frame" (Minsky 1975): a collection of knowledge about a single stereotyped situation.

In the above discussion however I have made a rather important slight of hand. Given a story we only want to consider those frames "about" things in the story. How is it that we decide which frames qualify? I was able to gloss over this because in most situations the problem, at least at a surface level, does not appear all that difficult. If the story is about trains, it will surely mention trains. So we see the word "train", and we assume that trains are relevant. What could be easier.

Unfortunately, this ease is deceptive for the story may mention many topics of which only a few are truely important to the story. For example.

> The lawyer took a cab to the restaurant near the university.

Here we have "lawyer", "cab", "restaurant" and "university" all of which are calling for our attention. Somehow on the basis of later lines we must weed out those which our only incidental.

But a more immediate difficulty are those situations where a story deals with a well defined topic, yet never explicitly mentions it. So consider:

> The woman waved as the man on the stage sawed her in half.

Here we have no difficulty in guessing that this is a magic trick, although nothing of the sort has been mentioned. We are able to take "low level" facts concerning sawing, stages, etc and put them together in a higher level "magician" hypothesis. As such, the phenomena illustrated here is essentialy bottom up.

Of course, any time we try to infer relatively global properties from more local evidence we may make mistakes. That this creates problems in frame determination is shown by the nice example of Collins et. al. (forthcomming). (To get the full import of the example, try pausing briefly after each sentence.)

> He plunked down $5 at the window. She tried to give him $2.50 but he refused to take it. So when they got inside she bought him a large bag of popcorn.

The first line is uniformly interpreted as a buying act (most even going further and assuming something like a bet at a racetrack). The second line is then seen as a return of change, but the refusal is problematic. The third line resolves all of this by suggesting a date at the movies - a considerable revision of the initial hypothesis.

To sumatize the last few paragraphs, the problem of frame determination in language comprehension involves three sub-problems.

1) Stories will typically elude to many higher frames, any of which might serve as the *context for the incoming lines.* How do we choose between them?
2) The words used in a story may not directly indicate the proper higher frame. How do we do the bottom up processing to find it?
3) If we are lead astray in the course of (2), how do we correct ourselves on the basis of further evidence.

In the paper which follows I will be primarily concentrate on (2) with (3) being mentioned occasionally. In essence my position on (1) is that it will not be too much of a problem, provided that the cost of setting up a context like "restaurant" is small. If it is never used then as the story goes on it will receeded into the background. How this "receeding" takes place I shall not say, since for one thing it is a problem in many areas, and for another, I don't know.

Concerning (2) and (3), we will be lead to a position similar to that of Minsky (1975) and Collins et. al (forthcomming) in that a frame will be selected on the basis of local evidence, and corrections will be made if it proves necessary. We will see however, that there are still a lot of problems with this, position which do not at first glance meet the eye.

2 THE CLUE INTERSECTION METHOD

Rather than immediately presenting my scheme, let me start by showing the problems with an alternative possibility, which I will call the "clue intersection" method. This alternative is by no means a straw man as one researcher has in fact explicitly suggested it (Fahlman 1977) and I for one find it a very natural way of thinking about the problem.

The idea behind this method is that we are given certain clues in the story about the nature of the correct frame, and to find the frame we simply intersect the possible frames associated with each clue. To see how this might work let us take a close look at the following example

> As Jack walked down the aisle he put a can of tunafish in his basket.

The clues here are things like "aisle", "tunafish' etc. Of course, I do not mean to say that it is the English words which are the clues, but rather the concepts which underlie the words. I will assume that we go from one to the other via an independent parsing algorithm. (However this assumes that there is no vicious interaction between frame determination and disambiguation. Given that disambiguation depends on prior frame determination (see (Hayes 1977) for numerous examples) this may be incorrect.) So the input to the frame determiner will be something like

```
ST-1 (WALK JACK-1 AISLE-1)
ST-2 (PERSON JACK-1)
ST-3 (EQUAL (NAME JACK-1) "JACK")
ST-4 (EQUAL (SEX JACK-1) MALE)
ST-5 (AISLE AISLE-1)
ST-6 (PUT JACK-1 TUNA-FISH-CAN-1 BASKET-1)
ST-7 (BASKET BASKET-1)
  . . .
```

The details of the representation do not figure in the paper, and those which do are fairly uncontroversial. An exception here is the use of specific predicates like BASKET or AISLE. We will return to this point in the conclusion.

Given this representation we can imagine one method of finding the appropriate frame. Our clues are the various predicates in the input, such as as AISLE, BASKET, etc. Index under each of them will be pointers to those places where it comes up. Under AISLE we might find CHURCH, THEATER, and SUPERMARKET, while BASKET will have LITTLE-RED-RIDING-HOOD, and SUPERMARKET. The point is that none of these clues will be unambiguous, but when we take the intersection the only thing which will be left is SUPERMARKET.

There are, however, problems with this view of things. For one thing it ignores what I will call the "clue selection" problem. Put in the plainest fashion the difficultiy here is deciding exactly what clues we will hand over to the clue resolution component, and in what order. So in the last example I selected some of the content of the sentence to hand over to the clue resolver, in particular AISLE, and BASKET. This seemed reasonable given that they do tend to suggest "supermarket", as desired. But there is more information in the sentence. It was Jack who did all of this. Why not intersect what we know about Jack with all of the rest, or WALK? Or again suppose something ever so slightly odd happens, such as the basket hitting a screwdriver which is on the floor. SCREWDRIVER will have various things indexed under it, but more likely than not the intersection with the rest of the items mentioned above will give us the null set. For that matter, is there any reason to only intersect things in the same sentence? The answer here is clearly no, since there are many examples which require just the opposite.

> Jack was walking down an aisle. He was pushing his basket.

But if we do not stop a sentence boundries where do we stop? It is ridiculous to go through the entire story collecting clues and then do a grand intersection at the end.

A reasonably natural solution to the clue selection problem would start with the observation that usually we already have a general frame. When new clues come in we see if they are compatible with what we already believe. If so, fine. If not, we see if the clue suggests a different context frame. If not (as with, say, WALK which occures so often as to be unsuggestive) then nothing more need be done. If there are newly suggested context frames they should be investigated. This will be done for every predicate. Now the clue intersection method is compatible with this idea, but in its broad outline we are moving closer to what I have been characterizing as the Minsky proposal.

Furthermore, there are some problems with the clue intersection method which go beyond the mere suggestive. Consider the following example

> Jack took a can of tunafish from the shelf. Then he turned on a light.

After the first line the intersection method shoul leave us undecided between KITCHEN and SUPERMARKET. The next line should resolve the issue, but how is it that it does so? It must have something to do with the fact that normally a shopper at a store would not be the person to turn lights on or off, while it would be perfectly normal for Jack to do it in what presumably is his own kitchen. But this sort of reasoning is not easily modeled by clue intersection because it would seem to depend on making inferences which are themselves dependent on having the context frames available. That is to say, before we can rule out SUPERMARKET, we need some piece of information from the SUPERMARKET frame which will enable us to say that Jack should not be turning

on a light, given that he is cast in the role of SHOPPER in that frame.

Interestingly enough, Fahlman (who I earlier noted is a proponent of the clue intersection method) had a major role in the evolution of the Minsky proposal which I advocate. As such it behoves us to consider why he then rejected the idea in (Fahlman 1977). His primary reason is his observation that frequently in vision one does not have any single clue which could serve as the basis for the first guess at the appropriate frame. Rather it would seem that one has a multitute of very vague features, each one of which could belong to a wide variety of objects or scenes. To select one of them for a first guess would be quite arbitrary and would involve one in an incredible amount of backtrack. It would seem much more plausible to simply do an intersection on the clues and in this way weed out the obvious implausibilites.

While this analysis of the situation in vision is quite plausibile, I estimate that high level vision is still in a sufficiently rudimentary state that these conclusions need not be taken as anything near the final word. Furthermore, even if it were proved that vision does need an intersection type process, I can easily believe that the process which goes on in vision is not the same as that which goes on in language. For one thing in vision there is a natural cut-off for clue selection - the single scene. For another, within the scene there is a natural metric on the likelyness of two features belonging to the same frame - distance. Weither or not these in fact work in vision, they do suggest why someone primarily worried about the vision problem would not see clue selection as the problem it appears to be in language.

3 DIFFERENT KINDS OF INDICES

As I have already said, the scheme I believe can surmount the difficulities presented in the last section is a variant on one proposed by Minsky, and elaborated by Fahlman (1974) and Kuipers (1975). The basic idea is that one major feature or clue is used to select an initial frame. Other facts are then interpreted in light of this frame. If they fit, fine. If not then another frame must be found which either complements or replaces the original frame. In the previous propolsals the original frame contained information about alternate frames to be tried in case of certain types of incompatabilities. This may or may not work in vision (which was the primary concern of those mentioned earlier) however I shall drop this part of the theory. In discourse there are simply too many ways a frame can be inappropriate to make this feasible. For example, it stretches credibility to believe that SUPERMARKET would suggest looking at KITCHEN in the case the shopper turns on the lights.

So let us consider a very simple example.

Jack walked over to the phone. He had to talk to Bill.

It seems reasonable to assume that we guess even before the second sentence that Jack will make a call. To anticipate this we must have TELEPHONING indexed under TELEPHONE. When we see the first line we first try to integrate it into what we already know. Since there will be nothing there to integrate it into, we try to construct something. To do this we look to see what we have indexed under TELEPHONE, find TELEPHONING, and try that out. Indeed it will work quite well, since one of the things under TELEPHONING is that the AGENT must be in the proximity of the phone, and Jack just accomplished that. Hence we are able to integrate (AT JACK-1 TELEPHONE-1) into the TELEPHONING frame, and everything is fine.

Nothing is ever really this simple however, and even in this example, which has been selected for its comparative simplicity, there are complications. I suspect most people have assumed in the course of this example that Jack is in a room, and perhaps have even gone so far as to assume he is at home. Nothing in the story says so of course, and if the next line went on to say that Jack put a dime into the phone we would quickly revise our theory.

To account for our tendency to place Jack in room, we must have a second index under TELEPHONE which points to places where phones are typically found. (An possible alternative is to have this stated under TELEPHONING but this would make it difficult to use the information in cases where no call is actually being made, so TELEPHONING, even if hypothesized, would not stay around long.) So we will hypothesize two kinds of indices, an ACTION index and a LOCATION index. This distinction should mirror the intuitive difference between placing and object in a typical local and placing an action in a typical sequence. Other distinctions of this sort exist and may well lead to the introduction of other such index types locating objects and actions in time for example. However I would anticipate that the total number is small (under 10, say).

To illustrate how these index types might hook up to TELEPHONE I will use a slightly extended version of the frame representation introduced in (Charniak 1977) and (Charniak forthcomming). From the point of view of this paper nothing is dependent on this choice. It is simply to give us a sepecific notation with which to work.

```
(TELEPHONE (OBJECT) ;The frame describes an OBJECT
                    ;(and not, say, an event).
     VARS:(THING)   ,I only introduce one variable
          ...       ;THING which is bound to the
                    ;token in the story repre-
                    ,senting the phone

 LOCATION:((ROOM (HOME-PHONE . THING))
           (PUBLIC-LOC (PAY-PHONE . THING)))

       ;If we instantiate the ROOM frame then the
       ;HOME-PHONE variable in it should be bound
       ;to the token which is bound to THING.
       ;Similarly for PUBLIC-LOC and PAY-PHONE.

   ACTION:((TELEPHONING (PHONE . THING)))
          ...) ,Other portions of the frame would
               ;describe its appearance, etc.
```

We will not be able to integrate the first line of our story into any other frame, so we will hypothesize the TELEPHONING frame and either the room frame or the public place frame. Given my subject data on what people assume, the room frame is placed, and hence tried, first. This will cause the creation of two new statements which serve to specify the frames now active, and their bindings

```
(TELEPHONING (PHONE . TELEPHONE-1))
(ROOM (ROOM . ROOM-1)
      (HOME-PHONE . TELEPHONE-1))
```

The syntax here is the name of the frame followed by dotted pairs (VARIABLE . BINDING). Earlier I used a place notation for simplicity, e.g.,

```
(TELEPHONE TELEPHONE-1)
```

In fact this would be converted internally to the dotted pair format:

```
(TELEPHONE (THING . TELEPHONE-1))
```

I might note that my variables are what Minsky (1975) calles "slots". They are also equivalent (to a first approximation) to KRL slots such as HOME-PHONE in.

```
[ROOM-1 (UNIT)
        <SELF (a ROOM with
               HOME-PHONE = TELEPHONE-1)>]
```

So we are hypothesizing 1) an instance of telephoning, where the only thing we know about it is the telephone involved, and 2) a room (ROOM-1) which at the moment is only furnished with a telephone. Note that this assumes that in our room frame we have an explicit slot for a telephone. This is equivalent to assuming that rooms typically have phones in them.

We can now integrate the fact that Jack is at the phone into the telephoning frame, assuming that this state is explicitly mentioned there (i.e. we know that as part of telephoning the AGENT must be AT the TELEPHONE). With this added our TELEPHONING statement will now be:

```
(TELEPHONING (AGENT . JACK-1)
             (TELEPHONE . TELEPHONE-1))
```

When the second line comes in we must see how this fits into the TELEPHONING frame, but this is a problem of integration. The frame determination problem is over for this example.

CONSTRAINTS ON THE HYPOTHESIS OF NEW FRAMES

Early on we noted that it was only necessary to worry about a new frame if we received information which did not fit in the old ones. Then when we introduced the two kinds of indecies we noted that we wanted to place events in a sequence of events, and objects in their typical local. This immediately suggests that when we get an unintegratable action we use the ACTION index on the predicate, while for objects we would use the LOCATION index. However, this is not general enough in at least two ways.

For one thing, often we will have a non-integratable action where it is not the action frame, but rather the objects involved in the action which suggest the appropriate frame. Our example of someone going over to a phone is a case in point. Here GO tells us nothing, but TELEPHONE is quite suggestive. To handle this the search for ACTION indices must include those which are on OBJECT frames describing the tokens involved in the action. So since Jack is going to something which is a telephone, we look on the ACTION index of TELEPHONE.

We must also extend our analysis to handle states. If we are told that Jack is <u>in</u> a restaurant we must activate RESTAURANTING. In our current analysis (RESTAURANT (THING . RESTAURANT-1)) will not do this since it is an OBJECT frame and hence will only be looking for LOCATIONs in which the restaurant will fit. Hence in this case the IN frame must act like the GO frame in looking for ACTION indecies in which it might fit. More generally, any state which is typically modified by an action should cause us to look for ACTION indicies. So IN or STICKY-ON would do so, SOLID or AGE would not. (But if in the case at hand we are told that something did change the SOLID status then we would treat it like an action, as in "In the morning the water in the pond was solid".

Up to this point then the frame selection process looks like this:

1) When a statement comes in try to integrate it into the frames which are already active. In general this can require inference and a major open problem is how much inference one performs before giving up. If the integration is successful, then go on to the next statement.
2) If the statement is a description of an object (i.e. an OBJECT frame) then use the LOCATION index on the frame to find a frame which incorporates the statement. Keep track of yet untried suggested LOCATION frames.
3) If the statement is an action or changable state, then look for an ACTION frame into which the action (or state) can be integrated. First look on the frame for the

action (or state) and then on the object frames describing the arguments of the action (or state). Again, keep track of any remaining ones.

4) There must be a complicated process by which we test frames for consistancy with what we know about the story already. If it is not consistant we must involve an even more complicated process of deciding which is more believable, previous hypothesis about the story, or the current frame. I have nothing to say on this aspect of the problem.

There is however one type of example which raises some doubts about the above algorithm. These mention some object with associated ACTION frames, but only in connection with states which do not demand an ACTION frame for their integration. For example:

The car was green. Jack had to be home by three.

In this example the above algorithm will not consider DRIVING because GREEN will not demand that we look at the action index assoicated with its arguments (the car). (Even if it did nothing would happen because the fact that the car is green would not integrate into DRIVING.) However, much to my surprise, when I gave this example to people they did not get the DRIVING frame either. However, with a modified example they do.

The steering wheel was green. Jack had to be home by three.

This is most mysterious. One suggestion (Lehnert personal communication) is that to "see" the steering wheel the "viewer" must be in the car, which inturn suggests driving (since IN would demand action integration). This may indeed be correct, but we must then explain why in the first example the fact that the viewer must be NEAR the car does not cause the same thing. In any case however, these examples are sufficiently odd that it seems inadvisable to mold a theory around them.

5 MORE COMPLEX INDICES

There is one way in which the telephone example makes the problem look simpler than it is. In the case of TELEPHONE it seems reasonable to have a direct link between the object TELEPHONE and the context frame TELEPHONING. In other cases this is not so clear. For example, we earlier consider the example:

The woman waved as the man on the stage sawed her in half.

Here it would seem that the notion of sawing a person in half is the crutial concept which leads us to magic, although the fact that the woman does not seem concerned, and the entire thing is happening on a stage certainly help re-enforce this idea. But presulably the output of our parser will simply state that we have here an incident of SAWING. Does this mean that we have under SAWING a pointer to MAGIC-PERFORMANCE? At first glance this seems odd at best. Some other examples where the same problem arise are:

The ground shook.
(EARTHQUAKE) (Example due to J. DeJong)

There were tin cans and streamers tied to the car. (WEDDING)

There were pieces of the fusilage scattered on the ground. (AIRPLANE ACCIDENT)

In the final analysis the real problem here is one of efficiency. If, for example we attach EARTHQUAKE to EARTH, then we will be looking at it in many circumstances when it is not applicable. (The alternative of attaching it to SHAKE is little better, and possibly worse since it would not handle "Jack felt the earth MOVE beneath him" assuming the average person gets EARTHQUAKE out of this also.)

One way to cut down the number of false suggestions is to complicate the indices we have on each frame. So far they have simply been lists of possibilities. Suppose we make them discrimination nets. So, under SAWING we would have various tests. On one branch would appear MAGIC-PERFORMANCE, but we would only get to it after many tests, one of which would see if the thing sawed was a person. In much the same way the discrimination net for EARTH could enquire about the action or state which caused us to access it. If it were a MOVE with the EARTH as the thing moved then EARTHQUAKE.

Note however that if there were few enough things attached to SAWING our net would not save significant time. Even if we were to access the MAGIC-PERFORMANCE frame the first thing we would do is check that the thing proposed for the SAWED-PERSON variable was indeed a person. The net only saves time when a single test in the net rules out a number of frames. At the present time I have not thought of enough frames associated with SAWING to make this worth while. But as I suspect this is primarily do to lack of work on my part, I will assume that discrimination nets will be required.

If we allow a discrimination net to ask arbitrary questions there will be the problem that it may ask questions which are not yet answered in the story. However a reasonable restriction which would prevent this would go as follows. Suppose statement A causes us to look at frames on an index of B. The discrimination net may only enquire about the predicate of A (EARTH looks to see if A was a MOVE), and what object frames describe the arguments of A or B (SAW looks to see if the thing sawed was a PERSON).

6 OTHER USES OF FRAME DETERMINATION

Earlier I noted that integrating a statement into a frame requires inference. Here I would like to point out that a modification of the above ideas would be helpful in this process as well. Consider the following:

Jack went to a restaurant. The menu was in Chinese. "What will I do now", thought Jack.

Our rules here will get us to RESTAURANTING after the first line. But if we are to understand the significance of the last line we must realize the import of line two; Jack can't read the menu. It would seem unlikely that RESTAURANTING would ask about the language of the menu, hence sentence two cannot be immediately integrated into RESTAURANTING. More reasonable would be to know that if something is in a foreign language it cannot be read, and one normally reads the menu so one can order. Only the second of these can plausibly be included in RESTAURANTING.

Given our algorithm the following will occur. The second line will become something like (IN-LANGUAGE MENU-1 CHINESE). Since the statement is not integrated we look to see if there is an ACTION pointer on IN-LANGUAGE. Indeed there is, and it will be to the following rule.

```
(READ (MOTIVATIONAL-ACTIVITY)
   VARS:  ...
  EVENT:
   (AND
      (SEE READER READING-MATERIAL)
      (IN-LANGUAGE READING-MATERIAL LANGUAGE)
      (KNOW READER LANGUAGE))
    ENABLES
   (KNOW-CONTENTS READER READING-MATERIAL))
```

In effect we are saying here that the typical signficance of something being in a certain language is whether a person can read it or not. This will cause us to activatve the READ frame. Initially there is little else we can do since at this point the we do not even know who is trying to read. However when we try to integrate READ we will be successful, and we will have further bound READER to JACK-1. At this point (and this is the modification required) we should return to READ and note that we can assume he does not know Chineese and hence will not be able to read the menu.

7 CONCLUSION

There is, of course, much I have not covered. The most glaring ommision is the lack of any discussion of how one detects a discrepency between a suggested frame and what we already know of the story. The problem is that a frame cannot afford to mention everything which is incompatable with it - there is simply too much. And the same is true for everything which is compatable. Furthermore, what would be enough to switch to a new frame under some circumstances would not be sufficient at other times. So "Jack walked down the isle and picked up a can of tunafish" takes us from CHURCH to SUPERMARKET. But if we added "from a pew" things are different. These are major problems and aside from (McDermott 72) and (Collins et. al. forthcomming) they have hardly been confronted, much less solved.

Early on I commented that the only controversial aspect of my representation was the use of very specific predicates (BASKET, AISLE, TELEPHONE, etc) rather than a break down into more primitive concepts. We might, for example, define AISEL as a path which is bounded on each side by things which are considered pieces of furniture (e.g., shelves or chairs). The problem with using a primitive representation here is that while it is somewhat plausible having SUPERMARKET and CHURCH indexd under AISLE, indexing them under PATH or some other component of the primitive definition is much less plausible. However, we can circumvent this problem by the use of discrimination nets, just as we did to get EARTHQUAKE from MOVE and EARTH. But, it should be noted that by using this method we are eliminating one of the benefits of a primitive analysis - we can no longer assume that we can get our information in a piecemeal fashion and come out with the same analysis. In particular we must get "aisle", or else we must get all of its components at the same time. If we do not then the discrimination net will fail to notice that we do not have any old path, we have an AISLE. Given this restriction the primitive and non primitive analyses come out pretty much the same. A primitive decomposition just becomes a long name for a higher level concept. Or to turn this around, the use of high level discriptions is not so controversial after all - it is simply a short name for a primitive decomposition.

ACKNOWLEDGEMENTS

I have benefited from conversations with J. Carbonelle, J. DeJong W. Lehnert, D. McDermott, and R. Wilensky, all of whom have been thinking about these problems for a long time. Many of their ideas have gone into this paper. This research was done at the Yale A.I. Project which is funded in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under contract N00014-75-C-1111.

REFERENCES

Charniak, E., A framed PAINTING on the representation of a common sense knowledge fragment. Journal of Cognitive Science, 1, 4, August 1977.

Charniak, E., On the use of framed knowledge in language comprehension, forthcomming.

Collins, A, Brown, J. S., and Larkin, K. M., Inference in text understanding, in: R. J. Spiro, B. C. Bruce, and W. F. Brewer (Eds.) Theoretical issues in reading comprehension. Hillsdale, N. J., Lawrence Erlbaum Associates, forthcomming.

Fahlman, S. E., A hypothesis-frame system for recognition problems, Working Paper 57, M.I.T. Artificial Intelligence Lab, 1974.

Fahlman, S. E., A system for representing and using real-world knowledge. Unpublished Ph.D. thesis, M.I.T., September 1977.

Hayes, P. J., Some association-based techniques for lexical disambiguation by machine.

TR25, University of Rochester Computer Science Department, June 1977.

Kuipers, B., A frame for frames, In D. Bobrow and A. Collins (Eds.) Representation and understanding, New York, Academic Press, 1975

McDermott, D., Assimilation of new information by a natural language understanding system, TR 291, M.I.T Artificial Intelligence Lab, 1972.

Minsky, M., A framework for representing knowledge. In P.H. Winston (Ed.), The psychology of compher vision, New York, McGraw-Hill, 1975, pp. 211-277.

Fragments of a Theory
f Human Plausible Reasoning

Allan Collins
Bolt Beranek and Newman Inc.

ABSTRACT

The paper outlines a computational theory of human plausible reasoning constructed from analysis of people's answers to everyday questions. Like logic, the theory is expressed in a content-independent formalism. Unlike logic, the theory specifies how different information in memory affects the certainty of the conclusions drawn The theory consists of a dimensionalized space of different inference types and their certainty conditions, including a variety of meta-inference types where the inference depends on the person's knowledge about his own knowledge. The protocols from people's answers to questions are analyzed in terms of the different inference types. The paper also discusses how memory is structured in multiple ways to support the different inference types, and how the information found in memory determines which inference types are triggered.

INTRODUCTION

The goal of this paper is to briefly describe a theory of human plausible reasoning I am currently developing (Collins, 1978) The theory is a procedural theory and hence one which can be implemented in a computer, as parts of it have been in the SCHOLAR and MAP-SCHOLAR systems (Carbonell & Collins, 1973; Collins & Warnock, 1974, Collins, Warnock, Aiello & Miller, 1975). The theory is expressed in the production-rule formalism of Newell (1973). Unlike logic, the theory specifies how different configurations of information affect the certainty of the conclusions drawn. These certainty conditions are in fact the major contribution of the theory.

Methodology of Constructing the Theory

To construct a theory of human plausible reasoning, I collected about 60 answers to everyday questions from 4 different subjects. The questions ranged from whether there are black princess phones to when the respondent first drank beer

The analysis of the protocols attempts to account for the reasoning and the conclusions drawn in the protocols in terms of 1) a taxonomy of plausible inferenoe types, 2) a taxonomy of default assumptions, and 3) what the subject must have known a priori As will be evident, this is an inferential analysis I am trying to construct a deep structure theory from the surface structure traces of the reasoning process.

The protocols have the following characteristics.

1) There are usually several different inference types used to answer any question.
2) The same inference types recur in many different answers.
3) People weigh all the evidence they find that bears on a question.
4) People are more or less certain depending on the certainty of the information, the certainty of the inferences, and on whether different inferences lead to the same or opposite conclusions

I can illustrate some of these characteristics of the protocols as well as several of the inference types in the theory with a protocol taken from a tutorial session on South American geography (Carbonell & Collins, 1973).

(T) There is some jungle in here (points to Venezuela) but this breaks into a savanna around the Orinoco (points to the Llanos in Venezuela and Colombia).

(S) Oh right, that is where they grow the coffee up there?

(T) I don't think that the savanna is used for growing coffee. The trouble is the savanna has a rainy season and you can't count on rain in general. But I don't know. This area around Sao Paulo (in Brazil) is coffee region, and it is sort of getting into the savanna region there.

In the protocol the tutor went through the following reasoning on the question of whether coffee is grown in the Llanos. Initially, the tutor made a hedged "no"

response for two reasons. First, the tutor did not have stored that the Llanos was used for growing coffee. Second, the tutor knew that coffee growing depends on a number of factors (e.g., rainfall, temperature, soil, and terrain), and that savannas do not have the correct value for growing coffee on at least one of those factors (i.e., reliable rainfall). However, the tutor later hedged his initial negative response, because he found some positive evidence. In particular, he thought the Brazilian savanna might overlap the coffee growing region in Brazil around Sao Paulo and that the Brazilian savanna might produce coffee. Thus by analogy the Llanos might also produce coffee. Hence, the tutor ended up saying "I don't know."

The answer exhibits a number of the important aspects of the protocols. In general, a number of inferences are used to derive an answer. Some of these are inference chains where the premise of one inference depends on the conclusion of another inference. In other cases the inferences are independent sources of evidence. When there are different sources of evidence, the subject weighs them together to determine his conclusion.

It is also apparent in this protocol how different pieces of information are found over time. What appears to happer is that the subject launches a search for relevant information, (Collins & Loftus, 1975). As relevant pieces of information are found (or are found to be missing), they trigger particular inferences. The type of inference applied is determined by the relation between the information found and the question asked. For example, if the subject knew that savannas are in general good for growing coffee, that would trigger a deduction. If the subject knew of one savanna somewhere that produced coffee, that would trigger an analogy. The search for information is such that the most relevant information is found first. In the protocol, the more relevant information about the unreliable rainfall in savannas was found before the more far fetched information about the coffee growing region in Brazil and its relation to the Brazilian savanna. Thus, information seems to be found at different times by an autonomous search process, and the particular information found determines inferences that are triggered.

THE THEORY

The theory specifies a large number of different inference types, together with the conditions that affect the certainty of each inference type. In the theory the different types of inference are arrayed in a five dimensional space.

The dimensions of the inference space are:

(1) Inferences on Knowledge vs Inferences on Meta-Knowledge

There are inference patterns based on people's knowledge, such as deduction and induction, and inference patterns based on people's knowledge about their own or other's knowledge (i.e. meta-knowledge) (Brown, 1977), such as lack-of-knowledge and confusability inferences. I refer to these latter as meta-inferences They are ubiquitous in the protocols, and yet they fall outside the scope of most theories of logic. The other four dimensions refer to the space of inferences but may also partially apply to the space of meta-inferences.

(2) Functional vs Set Inferences

For each type of inference, there is a functional variation and a set variation. The set variation involves mapping the property of one set (which may be a single-member set or instance) onto another set. The functional variation has an additional premise that the property to be mapped (the dependent variable) depends on other properties (the independent variables). The mapping of the property from one set to another makes use of this functional dependency. The set variation, in fact, is a degenerate form of the functional variation, which is used when people have little or no knowledge of the functional dependencies involved.

People's knowledge about functional dependencies consists of a kind of directional correlation. A judgment about whether a place can grow coffee might depend on factors that are causal precursors for coffee growing (e.g., temperature), correlated factors (e.g., other types of vegetation), or factors causally subsequent to coffee growing (e.g., export trade) For example, one might decide a place does not produce coffee, because it produces apples which seem incompatible with coffee, or because there is little export trade from the region. The directional nature of the correlation shows up in the last example. A region easily could have export trade without producing coffee, but it would be unlikely that a region would produce coffee without having export trade.

(3) Semantic, Spatial, vs Temporal Inferences

For each type of inference, there is a semantic, spatial, or temporal variation of the inference. Semantic inferences involve mapping properties across semantic space, spatial inferences across Euclidean space, and temporal inferences across time. These are treated as different types of inferences in the theory because the procedures for computing them are somewhat different. Semantic inferences are based on information structured in a semantic or conceptual memory (Quillian, 1968; Schank, 1972). Spatial inferences are based on information (or images) derived from a spatial structure (Collins & Warnock, 1975; Kosslyn & Schwartz, 1977). Temporal inferences are based on information derived from an event (or

episodic) structure (Tulving, 1972). Correlates of each of these types of memory structures are found in Winograd's SHRDLU (1972).

(4) Superordinate sets, similar sets, vs. subordinate sets

Inferences can involve mapping properties from superordinate sets, similar sets, or subordinate sets The property can be mapped from one set or from many sets (either exhaustively or not). The different kinds of mappings delineated in the theory are:

(a) Deduction (Superordinate Inferences) maps properties of the set onto subsets.
(b) Analogy (Similarity Inferences) maps properties from one set to a similar set.
(c) Induction maps properties of subsets of a set onto other subsets.
(d) Generalization (proof-by-cases) maps properties of subsets of a set onto the set.
(e) Abduction maps a subset with the same property as some set into the set

(5) Positive vs. Negative Inferences

Each type of inference has both a positive and negative version, depending on whether the mapping involves the presence or absence of a property

Assumptions of the Theory

The theory rests on a number of assumptions about the way information is represented and processed by people. I will describe briefly what these assumptions are.

Semantic Information I assume information about different concepts is represented in a cross-referenced, semantic structure (Quillian, 1968; Schank, 1972). The nodes in the network are schemas, which are the kind of structured objects implied by the notion of frames (Minsky, 1975) or scripts (Schank & Abelson, 1977). The links between nodes represent different relations between the concepts. The correlate of this kind of semantic structure in Winograd's SHRDLU (1972) was the cross-referenced information structure constructed by MICROPLANNER.

Spatial Information. I assume spatial information about concepts, such as the size, shape, color, or location of objects and places, is represented in a spatial structure, apart from but connected to the semantic structure (Collins & Warnock, 1974). The correlate of such a spatial representation in Winograd's SHRDLU (1972) was the Cartesian representation of the blocks on the table top.

Event information. Similarly event information is assumed to be stored in a form that preserves its temporal, causal, and goal structure. This requires a hierarchical structure of events and subevents nested according to the goals and subgoals of the actors involved in the events (Brown, Collins, & Harris, 1978). Such an event memory was constructed by Winograd's SHRDLU (1972) to record the movements of blocks and the goals they accomplished, in order to answer "why" and "how" questions about events in the Blocks World.

Retrieval I assume there are autonomous search processes that find relevant information with respect to any query (Collins & Loftus, 1975). The search process has access to semantic, spatial and temporal information in parallel, and whenever relevant information of any kind is found, it triggers an inference (Collins & Quillian, 1972, Kosslyn, Murphy, Bemesderfer & Feinstein, 1977.) The information found by the search processes determines what inference patterns are applied.

Matching Processes. I assume there are decision processes for determining whether any two concepts can be identified as the same. The semantic matching process could be that proposed by Collins & Loftus (1975) or by Smith, Shoben & Rips (1974). The spatial matching process compares places or objects to decide their spatial relation. Similarly, there must be a temporal matching process that determines the relation between two events.

Importance and Certainty. I assume that for each concept and relation a person has a notion of its relative importance (i.e. its criteriality), and his degree of certainty about its truth. In a computer, these could be stored as tags on the concepts and relations (Carbonell & Collins, 1973).

EXAMPLES OF INFERENCE RULES AND PROTOCOLS

Because it is impossible to present the entire theory here, I will give the formulations for three types of inference and show three protocols which illustrate these three types, as well as others. The three types are the lack-of-knowledge inference, the functional analogy, and the spatial superpart inference. They are all common inferences and serve to illustrate the different kinds of inferences in the theory.

The formal analysis of the protocols attempts to specify all the underlying inferences that the subject was using in his response. For the inferences that bear directly on the question, I have marked whether they are evidence for a negative or positive answer. Where a premise was not directly stored, but derived from another inference, I have indicated the inference from which it is derived. I have indicated the approximate degree of certainty by marking the conclusion with "Maybe", "Probably", or leaving it unmarked. Where a subject may be making a particular inference which the protocol does not clearly indicate, I have marked the inference "possible". Separating inferences in this manner is oversimplified, but has the virtue of being understandable.

Lack-of-Knowledge Inference

The lack-of-knowledge inference is the most common of all the meta-inferences. The protocol I selected to show the lack-of-knowledge inference shows the subject using a variety of meta-inferences to reach an initial conclusion which he then backs off a bit.

Q. Is the Nile longer than the Mekong River?

JB. I think so.

Q Why?

JB Because (pause) in junior high I read a book on rivers and I kept looking for the Hudson River because that was the river I knew about and it never appeared, and the Amazon was in there and the Nile was in there and all these rivers were in there, and they were big, and long, and important. The Mekong wasn't in there (pause) It could be just

Q. So therefore, it is not important.

JB. That's right It could be just an American view. At that time the Mekong wasn't so important

Underlying Inferences

1) Functional Abduction on Importance Level (Possible)
The importance of a river depends in part on how long it is
The Nile is very important
Probably the Nile is extremely long

2) Meta-Induction From Cases
I know the Amazon is extremely long
I know the Nile is extremely long (from 1)
I would know the Mekong is extremely long if it were

3) Lack-of-Knowledge Inference
I don't know the Mekong is extremely long
I would know the Mekong is extremely long if it were (from 2)
Probably the Mekong is not extremely long

4) Functional Abduction on Importance Level (Possible)
The importance of a river depends in part on length
The Mekong is not very important
Probably the Mekong is not extremely long

5) Simple Comparison (Positive Evidence)
The Mekong is not extremely long (from 3 and 4)
The Nile is extremely long (from 1)
The Nile is longer than the Mekong

6) Functional Attribution on Importance Level (Possible)
The importance of something depends on how remote it is
The Nile is very important
The Nile is less remote than the Mekong
Maybe the Nile is more important than the Mekong because it's less remote

7) Functional Alternative on Importance Level (Negative Evidence) (Possible)
The importance of a river depends on how close it is and how long it is
The Nile is more important than the Mekong because it's closer (from 6)
Maybe the Nile is not longer than the Mekong

Contributing to the certainty of these inferences are several meta-inferences working on importance level. The functional abductions (1 and 4) are suggested by the subject's tying length to importance. He seems to know that importance depends in part on length, and since he assigns different degrees of importance to the Nile and the Mekong, he must be using that in part to infer that the Mekong is not as long as the Nile. There also is a meta-induction he is making: that since he knows the Amazon and the Nile are very long, he would know the Mekong is long if it were. This meta-induction is acting on one of the certainty conditions for the lack-of-knowledge inference: the more similar cases stored with the given property, the more certain the inference Taken together, these inferences make the lack-of-knowledge inference very certain.

However at the end the subject backs off his conclusion because he finds another chain of reasoning that makes him less certain (inferences 6 and 7). The idea of "remoteness" only represents the underlying argument when interpreted in terms of conceptual distance What the subject is really doing is evaluating how remote Southeast Asia was at the time he was in junior high (before the Vietnam War). This notion of remoteness is the outcome of matching processes The Mekong was remote because it was far away culturally, historically, physically, etc. from America. Based on this the subject realizes that the Mekong's lack of importance may be due to this remoteness rather than its shortness in length His reasoning then depends on his notion of what alternative factors importance depends on, and how it might mislead him in this case. So this chain of reasoning is also acting on the certainty conditions affecting the lack-of-knowledge inference, but in the opposite direction from the other meta-inferences

The rule for a lack-of-knowledge inference is shown in the table below. It generally has the form. If it were true, I would know about it; I don't, so it must not be true. It is computed by comparing the importance level of the proposition in question against the depth of knowledge about the concepts involved (Collins et al, 1975; Gentner & Collins, 1978).

Lack-of-Knowledge Inference

1) If a person would know about a property for a given set if it were in a given range, and
2) if the person does not know about that property,
3) then infer that the property is not in the given range for that set.

Example

If Kissinger were 6' 6" tall, I would know he is very tall. I don't, so he must not be that tall.

Conditions that increase certainty:

1) The more important the particular set.
2) The less likely the property is in the given range.
3) The more information stored about the given set.
4) The more similar properties stored about the given set.
5) The more important the given property.
6) The more information stored about the given property.
7) The more similar sets stored that have the given property.

The conditions affecting the certainty of a lack-of-knowledge inference can be illustrated by the example in the table:

1) Condition 1 refers to the importance of the given set. In the example Kissinger is quite important, so one is more likely to know whether he is 6' 6" than whether Senator John Stennis is 6' 6" for example.
2) Condition 2 refers to the likelihood that the property is in the given range. Likelihood affects the inference in two ways: low likelihood makes a negative inference more certain a priori, and low likelihood also makes a property more unusual and therefore more likely to come to a person's attention. For example, it is less likely that Kissinger is 7' 2" than 6' 6", because 7' 2" is more unusual. If Kissinger were a basketball player, on the other hand, his being 6' 6" would not be unusual at all.
3) Condition 3 relates to the depth-of-knowledge about the given set. The more one knows about Kissinger, the more certainly one would know that he is 6' 6", if he is.
4) Condition 4 relates to the number of similar properties stored about the set (i.e. the relatedness of the information known about the set). If one knows a lot about Kissinger's physical appearance, one feels more certain one would know he is extremely tall, if he is.
5) Condition 5 relates to the importance of the particular property. Being extremely tall isn't as important as missing a leg say, so people are more likely to know if Kissinger is missing a leg.
6) Condition 6 relates to the depth-of-knowledge about the particular property. For example, a person who has particular expertise about the physical stature of people is more likely to know that Kissinger is extremely tall, if he is.
7) Condition 7 relates to the number of similar sets known to have the given property. For example, if one knows that Ed Muskie and Tip O'Neil are unusually tall, then one ought to know that Kissinger is unusually tall, if in fact he is 6' 6".

Functional Analogy

The initial protocol on coffee growing in the Llanos illustrated two functional inferences: a functional calculation concerning rainfall, and a functional analogy between the Brazilian savanna and the Llanos. One of the more common functional inferences is the functional analogy. The protocol I selected to illustrate it contrasts the use of a simple analogy and a functional analogy.

Q. Can a goose quack?

BF. No, a goose - Well, its like a duck, but its not a duck. It can honk, but to say it can quack. No, I think its vocal cords are built differently. They have a beak and everything, but no, it can't quack.

Underlying Inferences

1) Simple Analogy (Positive Evidence)
   A goose is similar to a duck
   A duck quacks
   Maybe a goose quacks

2) Importance-Level Inequality (Possible)
   I know a goose honks
   Quacking is as important as honking
   Probably I would know about a goose quacking if it did

3) Lack-of-Knowledge Inference (Negative Evidence) (Possible)
   I don't know that a goose quacks
   I would know about a goose quacking if it did (from 2)
   Probably a goose doesn't quack

4) Negative Functional Analogy (Negative Evidence)
   The sound a bird makes depends on its vocal cords
   A goose is different from a duck in its vocal cords
   A duck quacks
   Probably a goose doesn't quack

The simple analogy, which is based on a match of all the properties of ducks and geese, leads to the possible conclusion that a goose can quack, because a duck quacks. This inference shows up in the reference to "its like a duck" and in the uncertainty of the negative conclusion the student is drawing. It is positive evidence and only shows up to the degree it argues against the general negative conclusion.

The importance-level inequality and lack-of-knowledge inference are suggested by the sentence "It can honk, but to say it can quack." Here knowledge about honking seems to imply that a goose doesn't quack. I would argue that such an inference has to involve the lack-of-knowledge inference, since it is possible that a goose might sometimes honk and sometimes quack.

The functional analogy is apparent in the concern about vocal cords, which the subject thinks are the functional determinants of the sounds made. I think the sound is determined by the length of the neck, which is probably what the subject was thinking of. Honking may just be quacking resonated through a longer tube. But in any case, the mismatch the subject finds on the relevant factor leads to a negative conclusion which supports the lack-of-knowledge inference.

The table shows the rule for a functional analogy.

Functional Analogy

1) If a dependent variable depends on a number of independent variables, and
2) if one set matches another set on the independent variables, and
3) if the value of the dependent variable for one set is in a given range,
4) then infer that the value of the dependent variable for the other set is in the given range.

Example

The Brazilian savanna is like Llanos in its temperature, rainfall, soil, and vegetation. Thus, if the Brazilian savanna produces coffee, then the Llanos ought to also.

Conditions that increase certainty:

1) The more independent variables on which the two sets match, and the fewer on which they mismatch.
2) The greater the dependency on any independent variables on which the two sets match, and the less the dependency on any independent variables that mismatch.
3) The better the match on any independent variable.
4) The greater the dependency on those independent variables that match best.
5) The more certain the dependent variable is in the given range for the one set.
6) The more likely the value of the dependent variable is in the given range a priori.
7) The more certain the independent variables are in the given ranges for both sets.

I can illustrate the different certainty conditions for a functional analogy in terms of the example in the table.

1) Condition 1 refers to the number of factors on which the two sets match. If the two regions match only in climate and vegetation, that would be less strong evidence that they produce the same products than if they match on all four variables.
2) Condition 2 refers to the degree the dependent variable depends on different factors that match or mismatch. Coffee growing depends more on temperature and rainfall than on soil or vegetation. Thus a match on these first two factors makes the inference more certain than a match on the latter two factors.
3) Condition 3 relates to the quality of the match on any factor. The better the match with respect to temperature, rainfall, etc. the more certain the inference.
4) Condition 4 refers to the degree of dependency on those factors that match best. A good match with respect to the rainfall pattern leads to more certainty than a good match with respect to the vegetation.
5) Condition 5 relates to the certainty that the property is in the given range for the first set. The more certain one is that the Brazilian savanna produces coffee, the more certain the inference.
6) Condition 6 relates to the a priori likelihood that the property will be in the given range. The more likely that any region grows coffee, the more certain the inference.
7) Condition 7 relates to the certainty that the factors are in the given ranges for both sets. For example, the more certain that both savannas have the same temperature, etc., the more certain the inference.

Spatial Superpart Inference

The theory assumes that spatial inferences are made by constructing an image of the concepts involved, and making various computations on that image (Collins & Warnock, 1974; Kosslyn & Schwartz, 1977). An example of a spatial inference occurred in the earlier protocol about coffee growing, when the respondent concluded that a savanna might be used for growing coffee because he thought the coffee growing region around Sao Paulo might overlap the Brazilian savanna. This spatial matching process, which occurs in a variety of protocols, involves constructing a spatial image with both concepts in it, and finding their spatial relationship (e.g., degree of overlap, relative size or direction) from the constructed image.

The protocol I selected illustrates a spatial subpart inference, together with several other spatial and meta-inferences.

Q. Is Texas east of Seattle?

JB. Texas is south and east of Seattle.

Q. How did you get that?

JB. I essentially looked at a visual image of the U.S. where I remembered that Seattle was in Washington and know that its up in the left corner and I know that Texas is in the middle on the bottom. Sometimes you get fooled by things like that, like for example Las Vegas being further west than San Diego. This case I think we're O.K.

Underlying inferences

1) Spatial line slope inference
Washington is in upper left corner of the U.S.
Texas is on the middle bottom of U.S.
Line from Washington to Texas slopes east.

2) Spatial subpart inference (Positive evidence)
Line from Washington to Texas slopes east.
Seattle is part of Washington.
Line from Seattle to Texas slopes east

3) Meta Analogy (Negative evidence)
People are often mistaken in thinking that Las Vegas is east of San Diego, because Las Vegas is inland and San Diego is on the Pacific Coast.
Seattle, like San Diego, is on the Pacific coast.
Texas, like Las Vegas, is inland.
Maybe I am mistaken in thinking that Texas is east of Seattle.

4) Functional Modus Tollens (Positive evidence) (possible)
The Pacific coast misconception depends on the inland place being north of the coastal place.
Seattle is on the coast.
Texas is inland.
Texas is south of Seattle.
The Pacific coast misconception does not apply to Texas and Seattle.

In the protocol the subject constructs a line from Washington to Texas for the purpose of evaluating its slope. The constructed line does slope east, so he answers yes. Implicit in this protocol is a spatial subpart inference or spatial deduction, that Seattle is part of Washington and the slope of the line found earlier applies to Seattle. This kind of subpart inference was found to show up in response time by Stevens (1976).

The subject briefly reconsidered his conclusion because he thought of the "Pacific Coast Misconception," that people mistakenly think that places inland are always east of places on the coast. By the meta-analogy in 3, he inferred that maybe Seattle-Texas was like San Diego-Las Vegas in that the inland location was west of the coastal location. But the subject ruled out the analogy by some inference such as that shown in 4. Actually, the functional modus tollens in 4 hides the spatial processing that the subject probably used to rule out the analogy in 3. Probably, he knew that the reason for the "Pacific Coast Misconception" has to do with the southeasterly slant of the Pacific coast. By knowing that, you can figure out that the misconception depends on the inland location being north of the coastal location. I have finessed the spatial reasoning process by stating that conclusion as a premise in 4.

The next table shows the rule for a spatial superpart inference (or spatial deduction).

Spatial Superpart Inference

1) If a property is in a given range for some set, and
2) if another set is a subpart of that set,
3) then infer that the property is in that range for the subpart.

Example

It is raining in New England and Boston is in New England. Therefore it may be raining in Boston.

Conditions that increase certainty:

1) The more central the subpart is to the set.
2) The greater the average spatial extent of the property.
3) The greater the distance of the nearest set with a contradictory property.
4) The greater the extent of the subpart within the set.
5) The more likely a priori that the property is in the given range for the subpart.
6) The more certain the property is in the given range for the set.

The certainty conditions can be illustrated in terms of the example in the table:

1) Condition 1 relates to the centrality of the subpart. For example, if it's raining in New England it is more likely to be raining in Massachusetts than Maine because Massachusetts is more central.
2) Condition 2 relates to whether the property tends to be spatially distributed or not. For example, rain tends to be distributed over smaller areas than electric service, so it is a less certain inference that it is raining in Maine than that there is electric service in Maine, given that the property applies to New England.
3) Condition 3 relates to the distance to the nearest concept with a contradictory property. For example, if you know it's not raining in New Brunswick, that is stronger evidence against it's raining in Maine than if it's not raining in Montreal.
4) Condition 4 relates to the extent of the subpart. For example, if it's raining in New England it is more likely to be raining in Rhode Island than in Boston, because Rhode Island is larger.
5) Condition 5 relates to the a priori likelihood of the property. for example, if it's raining in Washington State, it's more likely to be raining in Seattle than in Spokane because Seattle gets more rain on the average.
6) Condition 6 relates to the person's certainty that the property holds for the concept. For example, the more certain the person is that it is raining in New England, the more certain that it's raining in Boston.

CONCLUSION

The theory I am developing is based on these and similar analyses of a large number of human protocols. Because the same inference types recur in many different answers, it is possible to abstract the systematic patterns in the inferences

themselves, and many of the different conditions that affect people's certainty in using different inference types.

ACKNOWLEDGEMENTS

I want to thank my colleagues who have influenced my views about inference over the years namely Marilyn Adams, Nelleke Aiello, John Seely Brown, Jaime Carbonell, Dedre Gentner, Mark Miller, Ross Quillian, Albert Stevens, and Eleanor Warnock. I particularly would like to thank Marilyn Adams for encouraging me to fit the inference types into a dimensionalized space, and John Seely Brown for bullying me into stating the rules and protocol analyses in a form understandable to readers.

This research was supported in part by the Advanced Research Projects Agency of the Department of Defense under Contract No. MDA 903-77-C-0025, and in part by a fellowship from the John Simon Guggenheim Memorial Foundation.

REFERENCES

Brown, A. L. Knowing when, where & how to remember. In R. Glaser (Ed.), Advances in instructional psychology. Hillsdale, NJ: Lawrence Erlbaum Associates, 1977, in press.

Brown, J.S., Collins, A., & Harris, G. Artificial intelligence and learning strategies. To appear in H.F. O'Neil (Ed.), Learning strategies. New York: Academic Press, 1978, in press.

Carbonell, J.R. & Collins, A. Natural Semantics in Artificial Intelligence. Proceedings of Third International Joint Conference on Artificial Intelligence, 1973, pp. 344-351. (Reprinted in the American Journal of Computational Linguistics, 1974, 1, Mfc. 3).

Collins, A. & Warnock, E.H. Semantic networks. BBN Report No. 3833, Bolt Beranek and Newman Inc., Cambridge, Mass., 1974.

Collins, A. M. & Loftus, E. F. A spreading activation theory of semantic processing. Psychological Review, 1975, 82, 407-428.

Collins, A., Warnock, E.H., Aiello, N. & Miller, M.L. Reasoning from Incomplete Knowledge, in D. Bobrow & A. Collins (eds.). Representation & understanding. New York: Academic Press, 1975.

Collins, A.M., & Quillian, M.R. Experiments on semantic memory and language comprehension. In L.W. Gregg (Ed.), Cognition in learning and memory. New York: Wiley, 1972.

Collins, A.M., Adams, M.J. & Pew, R.W. The Effectiveness of an interactive map display in tutoring geography. Journal of Educational Psychology, 1978, 70, 1-7.

Gentner, D., & Collins, A. Knowing about knowing: Effects of meta-knowledge on inference. Submitted to Cognitive Psychology.

Kosslyn, S.M., & Schwartz, S.P. A simulation of visual imagery. Cognitive Science, 1977, 1, 265-295.

Kosslyn, S.M., Murphy, G.L., Bemesderfer, M.E., & Feinstein, K.J. Category and continuum in mental comparisons. Journal of Experimental Psychology: General, 1977, 106, 341-375.

Minsky, M. A framework for representing knowledge. In P. H. Winston (Ed.), The psychology of computer vision. New York: McGraw-Hill, 1975.

Quillian, M. R. Semantic memory. In M. Minsky (Ed.), Semantic information processing. Cambridge, Mass.: MIT Press, 1968.

Schank, R. Conceptual Dependency: A Theory of Natural Language Understanding, Cognitive Psychology, 1972, 3, 552-631.

Schank, R. & Abelson, R. Scripts, plans, goals, and understanding. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1977.

Smith, E.E., Shoben, E.J., & Rips, L.J. Comparison processes in semantic memory. Psychological Review, 1974, 81, 214-241.

Stevens, A.L. The role of inference and internal structure in the representation of spatial information. Doctoral dissertation. University of California at San Diego, 1976.

Tulving, E. Episodic & semantic memory. In E. Tulving & W. Donaldson (Eds.), Organization & memory. New York: Academic Press, 1972.

Winograd, T. Understanding natural language. New York: Academic Press, 1972.

INDIRECT RESPONSES TO LOADED QUESTIONS*

S. Jerrold Kaplan

Department of Computer and Information Science
University of Pensylvania
Philadelphia, Pa. 19104

Casual users of Natural Language (NL) computer systems are typically inexpert not only with regard to the technical details of the underlying programs, but often with regard to the structure and/or content of the domain of discourse. Consequently, NL systems must be designed to respond appropriately when they can detect a misconception on the part of the user. Several conventions exist in cooperative conversation that allow a speaker to indirectly encode their intentions and beliefs about the domain into their utterances, ("loading" the utterances), and allow (in fact, often require) a cooperative respondent to address those intentions and beliefs beyond a literal, direct response. To be effective, NL computer systems must do the same. The problem, then, is to provide practical computational tools which will determine both when an indirect response is required, and what that response should be, without requiring that large amounts of domain dependent world knowledge be encoded in special formalisms.

This paper will take the position that distinguishing language driven inferences from domain driven inferences provides a framework for a solution to this problem in the Data Base (DB) query domain. An implemented query system (CO-OP) is described that uses this distinction to provide cooperative responses to DB queries, using only a standard (CODASYL) DB and a lexicon as sources of world knowledge.

WHAT IS A LOADED QUESTION?

A loaded question is one that indicates that the questioner presumes something to be true about the domain of discourse that is actually false. Question 1A presumes 1B. A cooperative speaker must find 1B assumable (i.e. not believe it to be false) in order to appropriately utter 1A in a cooperative conversation, intend it literally, and expect a correct, direct response.

1A. What day does John go to his weekly piano lesson?
1B. John takes weekly piano lessons.
1C. Tuesday.

Similarly, 2A presumes 2B.

2A. How many Bloody Marys did Bill down at the banquet?
2B. Hard liquor was available at the banquet.
2C. Zero.

If the questioner believed 2B to be false, there would be no point in asking 2A - s/he would already know that the correct answer had to be "Zero." (2C).

Both examples 1 and 2 can be explained by a convention of conversational cooperation: that a questioner should leave the respondent a choice of direct answers. That is, from the questioner's viewpoint upon asking a question, more than one direct answer must be possible.

It follows, then, that if a question presupposes something about the domain of discourse, as 1A does, that a questioner cannot felicitously utter the question and believe the presupposition to be false. This is a result of the fact that each direct answer to a question entails the question's presuppositions. (More formally, if question Q presupposes proposition P, then each question-direct answer pair (Q, Ai) entails P*.) Therefore,

* This work partially supported by NSF grant MCS 76-19466

* This entailment condition is a necessary but not sufficient condition for presupposition. The concept of presupposition normally includes a condition that the negation of a

if a questioner believes a presupposition to be false, s/he leaves no options for a correct, direct response - violating the convention. Conversely, a respondent can infer in a cooperative conversation from the fact that a question has been asked, that the questioner finds it's presuppositions assumable. (In the terms of [Keenan 71], the logical presupposition is pragmatically presupposed.)

Surprisingly, a more general semantic relationship exists that still allows a respondent to infer a questioner's beliefs. Consider the situation where a proposition is entailed by all but one of a question's direct answers. (Such a proposition will be called a presumption of the question.) By a similar argument, it follows that if a questioner believes that proposition to be false, s/he can infer the direct, correct answer to the question - it is the answer that does not entail the proposition. Once again, to ask such a question leaves the respondent no choice of (potentially) correct answers, violating the conversational convention. More importantly, upon being asked such a question, the respondent can infer what the questioner presumes about the context.

Question 2A above presumes 2B, but does not presuppose it: 2B is not entailed by the direct answer 2C. Nonetheless, a questioner must find 2B assumable to felicitously ask 2A in a cooperative conversation - to do otherwise would violate the cooperative convention. Similarly, 3B below is a presumption but not a presupposition of 3A (it is not entailed by 3C).

3A. Did Sandy pass the prelims?
3B. Sandy took the prelims.
3C. No.

If a questioner believes in the falsehood of a presupposition of a question, the question is inappropriate because s/he must believe that no direct answer can be correct; similarly, if a questioner believes in the falsehood of a presumption, the question is inappropriate because the questioner must know the answer to the question - it is the direct answer that does not entail the presumption. In short, the failure of a presupposition renders a question infelicitous because it leaves no options for a direct response; the failure of a presumption renders a question infelicitous because it leaves at most one option for a direct response. (Note that the definition of presumption subsumes the definition of presupposition in this context.)

---

proposition (in this case, the negation of the proposition expressed by a question-direct answer pair) should also entail its presuppositions. Consequently, the truth of a presupposition of a question is normally considered a prerequisite for an answer to be either true or false (for a more detailed discussion see [Keenan 73]). These subtleties of the concept of presupposition are irrelevant to this discussion, because false responses to questions are considered a-priori to be uncooperative.

CORRECTIVE INDIRECT RESPONSES

In a cooperative conversation, if a respondent detects that a questioner incorrectly presumes something about the domain of discourse, s/he is required to correct that misimpression. A failure to do so will implicitly confirm the questioner's presumption. Consequently, it is not always the case that a correct, direct answer is the most cooperative response. When an incorrect presumption is detected, it is more cooperative to correct the presumption than to give a direct response. Such a response can be called a Corrective Indirect Resporse. For example, imagine question 4A uttered in a cooperative conversation when the respondent knows that no departments sell knives.

4A. Which departments that sell knives also sell blade sharpeners?
4B. None.
4C. No deparments sell knives.

Although 4B is a direct, correct response in this context, it is less cooperative than 4C. This effect is explained by the fact that 4A presumes that some departments sell knives. To be cooperative, the respondent should correct the questioner's misimpression with an indirect response, informing the questioner that no departments sell knives (4C). (The direct, correct response 4B will reinforce the questioner's mistaken presumption in a cooperative conversation through it's *failure to state otherwise.*) A *failure* to produce corrective indirect responses is highly inappropriate in a cooperative conversation, and leads to "stonewalling" - the giving of very limited and precise responses that fail to address the larger goals and beliefs of the questioner.

RELEVANCE TO DB QUERIES

Most NL computer systems stonewall, because their designs erroneously assume that simply producing the correct, direct response to a query insures a cooperative response. (To a great extent, this assumption results from the view that NL

functions in this domain simply as a high-level query language.) Unfortunately, the domain of most realistic DB's are sufficiently complex that the user of a NL query facility (most likely a naive user) will frequently make incorrect presumptions in his or her queries. A NL system that is only capable of a direct response will necessarily produce meaningless responses to failed presuppositions, and stonewall on failed presumptions. Consider the following hypothetical exchange with a typical NL query system:

Q: Which students got a grade of F in CIS500 in Spring, '77?
R: Nil. [the empty set]
Q: Did anyone fail CIS500 in Spring, 77?
R: No.
Q: How many people passed CIS500 in Spring, '77?
R: Zero.
Q: Was CIS500 given in Spring '77?
R: No.

A cooperative NL query system should be able to detect that the initial query in the dialog incorrectly presumed that CIS500 was offered in Spring, '77, and respond appropriately. This ability is essential to a NL system that will function in a practical environment, because the fact that NL is used in the interaction will imply to the users that the normal cooperative conventions followed in a human dialog will be observed by the machine. The CO-OP query system, described below, obeys a number of conversational conventions.

While the definition of presumption given above may be of interest from a linguistic standpoint, it leaves much to be desired as a computational theory. Although it provides a descriptive model of certain aspects of conversational behavior, it does not provide an adequate basis for computing the presumptions of a given question in a reasonable way. By limiting the domain of application to the area of data retrieval, it is possible to show that the linguistic structure of questions encodes considerable information about the presumptions that the questioner has made. This structure can be exploited to compute a significant class of presumptions and provide appropriate corrective indirect responses.

LANGUAGE DRIVEN VS. DOMAIN DRIVEN INFERENCE

A long standing observation in AI research is that knowledge about the world - both procedural and declarative - is required in order to understand NL.* Consequently, a great deal of study has gone into determining just what type of knowledge is required, and how that knowledge is to be organized, accessed, and utilized. One practical difficulty with systems adopting this approach is that they require the encoding of large amounts of world knowledge to be properly tested, or even to function at all. It is not easy to determine if a particular failure of a system is due to an inadequacy in the formalism or simply an insufficient base of knowledge. Frequently, the collection and encoding of the appropriate knowledge is a painstaking and time consuming task, further hindering an effective evaluation. Most NL systems that follow this paradigm have a common property: they decompose the input into a suitable "meaning" representation, and rely on various deduction and/or reasoning mechanisms to provide the "intelligence" required to draw the necessary inferences. Inferences made in this way can be called domain** driven inferences, because they are motivated by the domain itself***.

While domain driven inferences are surely essential to an understanding of NL (and will be a required part of any comprehensive cognitive model of human intelligence), they alone are not sufficient to produce a reasonable understanding of NL. Consider the following story:

> John is pretty crazy, and sometimes does strange things. Yesterday he went to Sardi's for dinner. He sat down, examined the menu, ordered a steak, and got up and left.

For a NL system to infer that something unusual has happened in the story, it must distinguish the story from the events the story describes. A question answering system that would respond to "What did John eat?" with "A steak." cannot be said to understand the story. As a sequence of events, the passage contains nothing unusual - it simply omits details that can be filled in on the basis of common knowledge about restaurants. As a story,

---

* For example, to understand the statement "I bought a briefcase yesterday, and today the handle broke off." it is necessary to know that briefcases typically have handles.

** "Domain" here is meant to include general world knowledge, knowledge about the specific context, and inferencial rules of a general and/or specific nature about that knowledge.

*** Of course, these inferences are actually made on the basis of descriptions of the domain (the internal meaning representation) and not the domain itself. What is to be evaluated in such systems is the sufficiency of that description in representing the domain.

however, it raises expectations that the events do not. Drawing the inference "John didn't eat the steak he ordered." requires knowledge about the language in addition to knowledge about the domain. Inferences that require language related knowledge can be called language driven inferences.

Language driven inferences can be characterized as follows: they are based on the fact that a story, dialog, utterance, etc. is a description, and that the description itself may exhibit useful properties not associated with the thing being described.* These additional properties are used by speakers to encode essential information - a knowledge of language related conventions is required to understand NL.

Language driven inferences have several useful properties in a computational framework. First, being based on general knowledge about the language, they do not require a large infusion of knowledge to operate in differing domains. As a result, they are somewhat more amenable to encoding in computer systems (requiring less programming effort), and tend to be more transportable to new domains. Second, they do not appear to be as subject to runaway inferencing, i.e. the inferencing is driven (and hence controlled) by the phrasing of the input. Third, they can often achieve results approximating that of domain driven inference techniques with substantially less computational machinery and execution time.

As a simple example, consider the case of factive verbs. The sentence "John doesn't know that the Beatles broke up." carries the inference that the Beatles broke up. Treated as a domain driven inference, this result might typically be achieved as follows. The sentence could be parsed into a representation indicating John's lack of knowledge of the Beatles' breakup. Either immediately or at some suitable later time, a procedure might be invoked that encodes the knowledge "For someone to not know something, that something has to be the case." The inferencial procedures can then update the knowledge base accordingly. As a language driven inference, this inference can be regarded as a lexical property, i.e. that factive verbs presuppose their complements, and the complement immediately asserted, namely, that the Beatles broke up. (Note that this process cannot be reasonably said to "understand" the utterance, but achieves the same results.) Effectively, certain inference rules have been encoded directly into the lexical and syntactic structure of the language - facilitating the drawing of the inference without resorting to general reasoning processes.

Another (simpler) type of language driven inferences are those that relate specifically to the structure of the discourse, and not to it's meaning. Consider the interpretation of anaphoric references such as "former", "latter", "vice versa", "respectively", etc. These words exploit the linear nature of language to convey their meaning. To infer the appropriate referents, a NL system must retain a sufficient amount of the structure of the text to determine the relative positions of potential referents. If the system "digests" a text into a non-linear representation (a common procedure), it is likely to lose the information required for understanding.

The CO-OP system, described below, demonstrates that a language driven inference approach to computational systems can to a considerable extent produce appropriate NL behavior in practical domains without the overhead of a detailed and comprehensive world model. By limiting the domain of discourse to DB queries, the lexical and syntactic structure of the questions encodes sufficient information about the user's beliefs that a significant class of presumptions can be computed on a purely language driven basis.

CO-OP: A COOPERATIVE QUERY SYSTEM

The design and a pilot implementation of a NL query system (CO-OP) that provides cooperative responses and operates with a standard (CODASYL) DB system has been completed. In addition to producing direct answers, CO-OP is capable of producing a variety of indirect responses, including corrective indirect responses. The design methodology of the system is based on two observations:

1) To a large extent, the inferencing required to detect the need for an indirect response and to select the appropriate one can be driven directly from the lexical and syntactic structure of the input question, and

2) the information already encoded in standard ways in DB systems complements the language related knowledge sufficiently to produce appropriate conversational behavior without the need for separate "world knowledge" or "domain specific knowledge" modules.

Consequently, the inferencing mechanisms required to produce the cooperative responses are domain transparent, in the

---

* In the story example, assumptions about the connectedness of the story and the uniformity of the level of description give rise to the inference that John didn't eat what he ordered. These assumptions are conventions in the language, and not properties of the situation being described.

sense that they will produce appropriate behavior without modification from any suitable DB system. These mechanisms can therefore be transported to new DB's without modification.

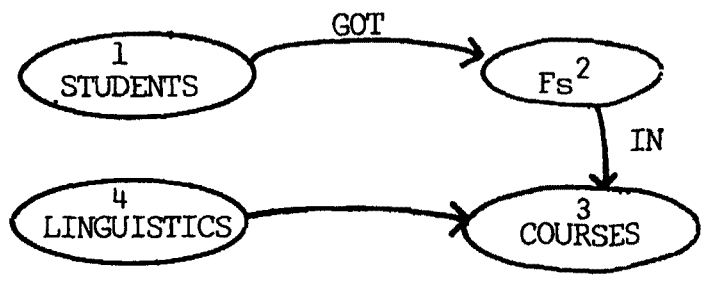
To illustrate this claim, a detailed description of the method by which corrective indirect responses are produced follows.

THE META QUERY LANGUAGE

Most DB queries can be viewed as requesting the selection of a subset (the response set) from a presented set of entities (this analysis follows [Belnap 76]). Normally, the presented set is put through a series of restrictions, each of which produces a subset, until the response set is found. This view is formalized in the procedures that manipulate an intermediate representation of the query, called the Meta Query Language (MQL).

The MQL is a graph structure, where the nodes represent sets (in the the mathematical, not the DB sense) "presented" by the user, and the edges represent binary relations defined on those sets, derived from the lexical and syntactic structure of the input query. Conceptually, the direct response to a query is an N-place relation realized by obtaining the referent of the sets in the DB, and composing them according to the binary relations. Each composition will have the effect of selecting a subset of the current sets. The subsets will contain the elements that survive (participate) in the relation. (Actually, the responses are realized in a much more efficient fashion - this is simply a convenient view.)

As an example, consider the query "Which students got Fs in Linguistics courses?" as diagrammed ir FIGURE 1.



Meta Query Language representation of "Which students got Fs in Linguistics courses?"

FIGURE 1

This query would be parsed as presenting 4 sets: "students", "Fs", "Linguistics", and "courses". (The sets "Linguistics" and "Fs" may appear counterintuitive, but should be viewed as singleton entities assumed by the user to exist somewhere in the DB.) The direct answer to the query would be a 4 place relation consisting of a column of students, grades (all Fs), departments (all Linguistics), and courses. For convenience, the columns containing singleton sets (grades and departments) would be removed, and the remaining list of students and associated courses presented to the user.

Executing the query consists of passing the MQL representation of the query to an interpretive component that produces a query suitable for execution on a CODASYL DB using information associated for this purpose with the lexical items in the MQL. (The specific knowledge required to perform this translation is encoded purely at the lexical level: the only additional domain dependent knowledge required is access to the DB schema.)

The MQL, by encoding some of the syntactic relationships present in the NL query, can hardly be said to capture the meaning of the question: it is merely a convenient representation formalizing certain linguistic characteristics of the query. The procedures that mainipulate this representation to generate inferences are based on observations of a general nature regarding these syntactic relationships. Consequently, these inferences are language driven inferences.

COMPUTING CORRECTIVE INDIRECT RESPONSES

The crucial observation required to produce a reasonable set of corrective indirect responses is that the MQL query presumes the non-emptiness of it's connected subgraphs. Each connected subgraph corresponds to a presumption the user has made about the domain of discourse. Consequently, should the initial query return a null response, the control structure can check the users presumptions by passing each connected subgraph to the interpretive component to check it's non-emptiness (notice that each subgraph itself constitutes a well formed query). Should a presumption prove false, an appropriate indirect response can be generated, rather than a meaningless or misleading direct response of "None."

For example, in the query of FIGURE 1, the subgraphs and their corresponding corrective indirect responses are (the numbers represent the sets the subgraphs consist of):

1) "I don't know of any students."
2) "I don't know of any Fs."
3) "I don't know of any courses."
4) "I don't know of any Linguistics."
1,2) "I don't know of any students that got Fs."
2,3) "I don't know of any Fs in

courses."
3,4) "I don't know of any Linguistics courses."
1,2,3) "I don't know of any students that got Fs in courses."
2,3,4) "I don't know of any Fs ir linguistics courses."

Suppose that there are no linguistics courses in the DB. Rather than presenting the direct, correct answer of "None." the control structure will pass each connected subgraph in turn to be executed against the DB. It will discover that no linguistics courses exist in the DB, and so will respond with "I don't know of any linguistics courses." This corrective indirect response (and all responses generated through this method) will entail the direct answer, since they will entail the emptiness of the direct response set.

Several aspects of this procedure are worthy of note. First, although the selection of the response is dependent on knowledge of the domain (as encoded in a very general sense in the DB system - not as separate theorems, structures, or programs), the computation of the presumptions is totally independent of domain specific knowledge. Because these inferences are driven solely by the parser output (MQL representation), the procedures that determine the presumptions (by computing subgraphs) require no knowledge of the DB. Consequently, producing corrective indirect responses from another DB, or even another DB system, requires no changes to the inferencing procedures. Secondly, the mechanism for selecting the indirect response is identical to the procedure for executing a query. No additional computational machinery need be invoked to select the appropriate indirect response. Thirdly, the computational overhead involved in checking and correcting the users presumptions is not incurred unless it has been determined that an indirect response may be required. Should the query succeed initially, no penalty in execution time will be paid for the ability to produce the indirect responses. In addition, the only increase in space overhead is a small control program to produce the appropriate subgraphs (the linguistic generation of the indirect response is essentially free - it is a trivial addition to the paraphrase component already used in the parsing phase).

Corrective indirect responses, produced in this fashion, are language driven inferences, because they are derived directly from the structure of the query as represented by the MQL. If the query were phrased differently, a different set of presumptions would be computed. (This is not a drawback, as it might seem at first - it insures that the response will be in terms that the user understands.) For these reasons, corrective indirect responses, made possible by a careful choice of representations and associated algorithms, are produced in a domain transparent fashion with minimal system overhead using knowledge already available in the DB.

A SHORT SAMPLE SESSION

This section contains an example interaction with the preliminary implementation of the CO-OP system as it now stands. An actual session has been abstracted and commented here.

CO-OP is currently implemented on a DEC KI-10 under TOPS-10 mainly in UCI-RUTGERS-LISP, and runs in 95K words, including the SEED DB system. SEED is a commercially available CODASYL DB system, marketed by International Data Base Systems, Inc. [Gerritsen 78]. The DB is a live one, used by the National Center For Atmospheric Research (NCAR) in Boulder Co. to keep track of their computer users, projects, research sponsors, etc. The primary purpose of the DB is to track the use of NCAR facilities, for accounting and other uses.

The user in the example is a hypothetical Computer Science department chairperson who has heard that the NCAR DB contains information about who is sponsoring research in various areas, and is interested in securing some research funds. S/he believes (incorrectly, it turns out) that funds are available to study mass storage. S/he begins by asking:

Q: Who sponsors active projects in mass storage?

R: I don't know of any projects in mass storage

The system is unaware of any projects in mass storage, and so explicitly corrects the users presumption that there are some. The user tries a more general question.

Q: Who sponsors research in computer science?

R: I don't know of anyone that sponsors research in computer science.

Again, a corrective indirect response - but this time stating that no sponsors are known, rather than that no suitable projects are known. In addition to being able to generate corrective indirect responses, the system can choose the most appropriate one when more than one is applicable. This is done by observing that

the subgraphs of an MQL query are partially ordered according to a subgraph relation, and it is most appropriate to correct the "least" failing subgraphs with respect to this ordering.

Getting nowhere, the chairperson now asks:

Q: What are the areas?

and receives a list of the areas of interest (s/he knew that the projects at NCAR are grouped according to a predetermined list of areas of interest), identified by the area number and name. S/he observes that area #6 (data processing) and area #12 (numerical methods) look interesting, and follows up with:

Q: Who sponsors projects in area 6?

The response is a list of sponsor names with a supportive indirect component of the projects they sponsor in area 6, the name of the area (because only the number was supplied - the system doesn't currently remember that it just provided the area name to the user), and the project numbers of the sponsored projects. The user now decides that Nasa Headquarters looks the most promising, and so asks:

Q: What is sponsored in numerical methods by Nasa Headquarters?

After checking the DB, the system discovers that Nasa Headquarters doesn't sponsor anything in numerical methods. Additionally, it is unable to detect any failed presumptions on the part of the user. It therefore provides a negative response followed by a suggestive indirect response listing the projects that Nasa Headquarters sponsors in any area, in the hope that this will be helpful to the user.

R: I don't know of anything in numerical methods that Nasa Headquarters sponsors. But you might be interested in anything that Nasa Headquarters sponsors...

After perusing this list, the chairperson concludes that although the projects don't look very promising, s/he will get in touch with Nasa Headquarters. S/he asks:

Q: Who is the contact at Nasa Headquarters?

It turns out that there is a contact at Nasa Headquarters for each project sponsored, and so the system prints out the list (sorted by contact), along with the projects they sponsor. Although the user has presumed that there is only one contact at Nasa Headquarters, the system provides the entire list, without objecting. This and other forms of sloppy reference are tolerated by the system.

CONCLUSION

The problem of producing apparently intelligent behavior from a NL system has traditionally been viewed in Artificial Intelligence as a problem of modelling human cognitive processes, or modelling knowledge about the real world. It has been demonstrated here that such approaches must include a pragmatic theory of the conventions and properties of the use of language, to function effectively. Domain driven inferences must be complemented by language driven inferences to appropriately process NL. Further, it has been argued that language driven inference mechanisms help to control the inference process, and can provide a more general and computationally attractive solutions to many problems previously thought to require domain driven inference.

A descriptive theory of one type of cooperative indirect response to inappropriate questions has been presented, and extended to a prescriptive (computational) theory by restricting the domain of application to DB query systems. This theory has been implemented using language driven mechanisms in the design of CO-OP, a cooperative query system. The result is the generation of appropriate corrective indirect responses in a computationally efficient and domain transparent fashion.

REFERENCES

Austin, J.L., How To Do Things With Words, J.O. Urmson, Ed., Oxford University Press, N.Y. 1965.

Belnap, N. D., and T. B. Steel, The Logic of Questions and Answers, Yale University Press, New Haven, Conn., 1976.

Gerritsen, Rob, SEED Reference Manual, Version C00 - B04 draft, International Data Base Systems, Inc., Philadelphia, Pa., 19104, 1978.

Grice, H. P., "Logic and Conversation", in Syntax and Semantics: Speech Acts, Vol. 3, (P. Cole and J. L. Morgan, Ed.)

Academic Press, N.Y., 1975.

Harris, L. R., "Natural Language Data Base Query: Using the Data Base Itself as the Definition of World Knowledge and as an Extension of the Dictionary", Technical Report #TR 77-2, Mathematics Dept., Dartmouth College, Hanover, N.H., 1977.

Joshi, A. K., S. J. Kaplan, and R. M. Lee, "Approximate Responses from a Data Base Query System: An Application of Inferencing in Natural Language", in Proceedings of the 5th IJCAI, Vol. 1, 1977.

Kaplan, S. Jerrold, "Cooperative Responses from a Natural Language Data Base Query System: Preliminary Report", Technical Report, Dept. of Computer and Information Science, Moore School, University of Pennsylvania, Philadelphia, Pa., 1977.

Kaplan, S. J., and Joshi, A. K., "Cooperative Responses: An Application of Discourse Inference to Data Base Query Systems", to appear in proceedings of the Second Annual Conference of the Canadian Society for Computational Studies of Intelligence, Toronto, Ontario, July, 1978.

Joshi, A. K., Kaplan, S. J., and Sag, I. A., "Cooperative Responses: Why Query Systems Stonewall", to appear in proceedings of the 7th International Conference on Computational Linguistics, Bergen, Norway, August, 1978.

Keenan, E. L., "Two kinds of Presupposition in Natural Language", in Studies in Linguistic Semantics, (C. J. Fillmore and D. T. Langendoen, Ed.), Holt, Rinehart, and Winston, N.Y., 1971.

Keenan, E. L., and Hull, R. D., "The Logical Presuppositions of Questions and Answers", in Prasuppositionen in Philosophie und Linguistik, (Petofi and Frank, Ed.), Athenaum Verlag, Frankfurt, 1973.

Lee, Ronald M. "Informative Failure in Database Queries", Working Paper #77-11-05, Dept. of Decision Sciences, Wharton School, University of Pennsylvania, 1977.

Lehnert, W., "Human and Computational Question Answering", in Cognitive Science, Vol. 1, #1, 1977.

Searle, J. R., Speech Acts, an Essay in the Philosophy of Language, Cambridge University Press, London, 1969.

Weischedel, R. M., Computation of a Unique Class of Inferences: Presupposition and Entailment, Ph.D. dissertation, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, Pa. 1975.

ON REASONING BY DEFAULT

Raymond Reiter

Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada

ABSTRACT

This paper surveys a number of kinds of default reasoning in Artificial Intelligence, specifically, default assignments to variables, the closed world assumption, the frame default for causal worlds, exceptions as defaults, and negation in Artificial Intelligence programming languages. Some of these defaults provide clear representational and computational advantanges over their corresponding first order theories. Finally, the paper discusses various difficulties associated with default theories.

If I don't know I don't know
I think I know
If I don't know I know
I think I don't know

R.D. Laing, Knots

1. INTRODUCTION

Default reasoning is commonly used in natural language understanding systems and in Artificial Intelligence in general. We use the term "default reasoning" to denote the process of arriving at conclusions based upon patterns of inference of the form "In the absence of any information to the contrary, assume..." In this paper, we take this pattern to have the more formal meaning "If certain information cannot be deduced from the given knowledge base, then conclude..." Such reasoning represents a form of plausible inference and is typically required whenever conclusions must be drawn despite the absence of total knowledge about a world.

In order to fix some of these ideas, we begin by surveying a number of instances of default reasoning as they are commonly invoked in A.I. Specifically, we discuss default assignments to variables, the closed world assumption, the frame default for causal worlds, exceptions as defaults, and negation in A.I. programming languages. We shall see that these may all be formalized by introducing a single default operator M where MW is taken to mean "W is not deducible from the given knowledge base".

In addition, we shall discover that the closed world and frame defaults provide clear representational and computational advantages over their corresponding first order theories. The former eliminates the need for an explicit representation of negative knowledge about a world, while the latter eliminates the so-called frame axioms for dynamic worlds.

Finally, we discuss various problems which arise as a result of augmenting first order logic with a default operator.

2. SOME INSTANCES OF DEFAULT REASONING IN A.I.

The use of default reasoning in A.I. is far more widespread than is commonly realized. The purpose of this section is to point out a variety of seemingly different situations in which such reasoning arises, to accent common patterns which emerge when defaults are formalized, and to indicate certain representational and computational advantages of default reasoning.

2.1 Default Assignments to Variables

A number of knowledge representation schemes, e.g. FRL [Roberts and Goldstein 1977], KRL [Bobrow and Winograd 1977], explicitly provide for the assignment of default values to variables (slots, terminals). For example, in KRL the unit for a person in an airline travel system has the form:

[Person UNIT Basic

hometown((a City) PaloAlto; DEFAULT)

We can view this declaration as an instruction to the KRL interpreter to carry out the following:
If x is a person, then in the absence of any information to the contrary, assume hometown(x)=PaloAlto, or phrased in a way which makes explicit the fact that a default assignment is being made to a variable:
If x is a person and no value can be determined for the variable y such that hometown(x)=y, then assume y=PaloAlto.
Notice that in assigning a default value to a variable, it is not sufficient to fail to find an explicit match for the variable in the data base. For example, the non existence in the data base of a fact of the form hometown(JohnDoe)=y for some city y does not necessarily permit the default assignment y=PaloAlto. It might be the case that the following information is available:

$$(x/\text{EMPLOYER})(y/\text{PERSON})(z/\text{CITY})\text{EMPLOYS}(x,y) \wedge \text{location}(x)=z \supset \text{hometown}(y)=z$$ [1]

i.e. a person's hometown is the same as his or her employer. In this case the default assignment y=PaloAlto can be made only if we fail to deduce the existence of an employer x and city z such that

$$\text{EMPLOYS}(x,\text{JohnDoe}) \wedge \text{location}(x)=z$$

in general then, default assignments to variables are permitted only as a result of failure of some attempted deduction. We can formulate a general inference pattern for the default assignment of values to variables:
For all $x_1,\ldots,x_n$ in classes $\tau_1,\ldots,\tau_n$ respectively, if we fail to deduce $(Ey/\theta)P(x_1,\ldots,x_n,y)$ then infer the default statement

$$P(x_1,\ldots,x_n,\langle\text{default value for } y\rangle)$$

or more succinctly,

$$\frac{(x_1/\tau_1)\ldots(x_n/\tau_n) \quad \not\vdash (Ey/\theta)P(x_1,\ldots,x_n,y)}{P(x_1,\ldots,x_n,\langle\text{default value for } y\rangle)} \qquad (D1)$$

Here $\not\vdash$ is to be read "fail to deduce", θ and the τ's are types, and $P(x_1,\ldots,x_n,y)$ is any statement about the variables $x_1,\ldots,x_n,y$. There are some serious difficulties associated with just what exactly is meant by "$\not\vdash$" but we shall defer these issues for the moment and rely instead on the reader's intuition. The default rule for home towns can now be seen as an instance of the above pattern:

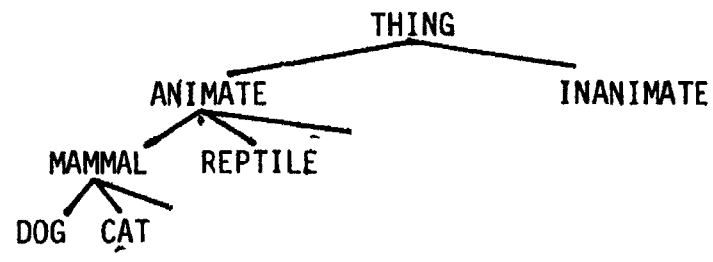
$$(x/\text{PERSON})\ \frac{\not\vdash (Ey/\text{CITY})\text{hometown}(x)=y}{\text{hometown}(x)=\text{PaloAlto}}$$

2.2 THE CLOSED WORLD ASSUMPTION

It seems not generally recognized that the reasoning components of many natural language understanding systems have default assumptions built into them. The representation of knowledge upon which the reasoner computes does not explicitly indicate certain default assumptions. Rather these defaults are realized as part of the code of the reasoner, or, as we shall say, following [Hayes 1977], as part of the reasoner's process structure. The most common such default corresponds to what has elsewhere been referred to as the closed world assumption [Reiter 1978]. In this section we describe two commonly used closed world defaults.

2.2.1 Hierarchies

As an illustration of the class of closed world defaults, consider standard taxonomies (IS-A hierarchies) as they are usually represented in the A.I. literature, for example the following:



This has, as its first order logical representation, the following:

[1] Throughout this paper we shall use a typed logical representation language. Types, e.g. EMPLOYER, PERSON, CITY correspond to the usual categories of IS-A hierarchies. A typed universal quantifier like (x/EMPLOYER) is read "for all x which belong to the class EMPLOYER" or simply "for all employers x". A typed existential quantifier like (Ex/CITY) is read "there is a city x". The notation derives from that used by Woods in his "FOR function" [Woods 1968].

$$\left.\begin{array}{l}(x)DOG(x) \supset MAMMAL(x)\\ (x)CAT(x) \supset MAMMAL(x)\\ (x)MAMMAL(x) \supset ANIMATE(x)\\ \text{etc.}\end{array}\right\} \qquad (2.1)$$

Now if Fido is known to be a dog we can conclude that Fido is animate in either of two essentially isomorphic ways:

1. If the hierarchy is implemented as some sort of network, then we infer ANIMATE(fido) if the class ANIMATE lies "above" DOG i.e. there is some pointer chain leading from node DOG to node ANIMATE in the network.
2. If the hierarchy is implemented as a set of first order formulae, then we conclude ANIMATE(fido) if we can forward chain (modus ponens) with DOG(fido) to derive ANIMATE(fido). This forward chaining from DOG(fido) to ANIMATE(fido) corresponds exactly to following pointers from node DOG to node ANIMATE in the network.

Thus far, there is no essential difference between a network representation of a hierarchy with its pointer-chasing interpreter and a first order representation with its forward chaining theorem proving interpreter. A fundamental distinction arises with respect to negation. As an example, consider how one deduces that Fido is not a reptile. A network interpreter will determine that the node REPTILE does not lie "above" DOG and will thereby conclude that DOGs are not REPTILEs so that ¬REPTILE(fido) is deduced. On the other hand, theorem prover will try to prove ¬REPTILE(fido). Given the above first order representation, no such proof exists. The reason is clear - nothing in the representation (2.1) states that the categories MAMMAL and REPTILE are disjoint. For the theorem prover to deal with negative information, the knowledge base (2.1) must be augmented by the following facts stating that the categories of the hierarchy are disjoint:

$$\left.\begin{array}{l}(x)ANIMATE(x) \supset \neg INANIMATE(x)\\ (x)MAMMAL(x) \supset \neg REPTILE(x)\\ (x)DOG(x) \supset \neg CAT(x)\end{array}\right\} \qquad (2.2)$$

It is now clear that a first order theorem proving interpreter can establish ¬REPTILE(fido) by a pure forward chaining proof procedure from DOG(fido) using (2.1) and (2.2). However, unlike the earlier proof of ANIMATE(fido), this proof of REPTILE(fido) is not isomorphic to that generated by the network interpreter. (Recall that the network interpreter deduces ¬REPTILE(fido) by failing to find a pointer chain linking DOG and REPTILE). Moreover, while the network interpreter must contend only with a representation equivalent to that of (2.1), the theorem prover must additionally utilize the negative information (2.2). Somehow, then, the process structure of the network interpreter implicitly represents the negative knowledge (2.2), while computing only on declarative knowledge equivalent to (2.1).

We can best distinguish the two approaches by observing that two different logics are involved. To see this, consider modifying the theorem prover so as to simulate the network process structure. Since the network interpreter tries, and fails, to establish a pointer chain from DOG to REPTILE using a declarative knowledge base equivalent to (2.1), the theorem prover can likewise attempt to prove REPTILE(fido) using only (2.1). As for the network interpreter, this attempt will fail. If we now endow the theorem prover with the additional inference rule:

"If you fail to deduce REPTILE(fido) then conclude REPTILE(fido)"

the deduction of REPTILE(fido) will be isomorphic to that of the network interpreter. More generally, we require an inference schema, applicable to any of the monadic predicates MAMMAL, DOG, CAT, etc. of the hierarchy:

"If x is an individual and P(x) cannot be deduced, then infer ¬P(x)"

or in the notation of the previous section

$$(x)\ \frac{\not\vdash P(x)}{\neg P(x)} \qquad (D2)$$

What we have argued then is that the process structure of a network interpreter is formally equivalent to that of a first order theorem prover augmented by the ability to use the inference schema (D2). In a sense, a network interpreter is the compiled form of such an augmented theorem prover.

There are several points worth noting:

1. The schema (D2) is not a first order rule of inference since the operator $\not\vdash$ is not a first order notion. (It is a meta notion.) Thus a theorem

prover which evokes (D2) in order to establish negative conclusions by failure is not performing first order deductions.

2. The schema (D2) has a similar pattern to the default schema (D1)

3. In the presence of the default schema (D2), the negative knowledge (2.2), which would be necessary in the absence of (D2), is not required. As we shall see in the next section, this property is a general characteristic of the closed world default, and leads to a significant reduction in the complexity of both the representation and processing of knowledge.

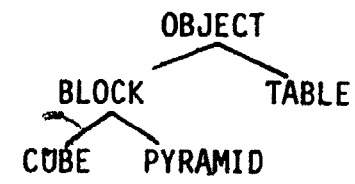
2.2.2 The Closed World Default

The schema (D2) is actually a special case of the following more general default schema:

$$(x_1/\tau_1)\ldots(x_n/\tau_n)\ \frac{\not\vdash P(x_1,\ldots,x_n)}{\neg P(x_1,\ldots,x_n)} \qquad \text{(D3)}$$

If (D3) is in force for all predicates P of some domain, then reasoning is being done under the closed world assumption [Reiter 1978]. In most A.I. representation schemes, hierarchies are treated as closed world domains. The use of the closed world assumption in A.I. and in ordinary human reasoning extends beyond such hierarchies, however. As a simple example, consider an airline schedule for a direct Air Canada flight from Vancouver to New York. If none is found, one assumes that no such flight exists. Formally, we can view the schedule as a data base, and the query as an attempt to establish DIRECTLY-CONNECTS(AC, Van,NY). This fails, whence one concludes ¬DIRECTLY-CONNECTS(AC,Van,NY) by an application of schema (D3). Such schedules are designed to be used under the closed world assumption. They contain only positive information; negative information is inferred by default. There is one very good reason for making the closed world assumption in this setting. The number of negative facts vastly exceeds the number of positive ones. For example, Air Canada does not directly connec Vancouver and Moscow, or Toronto and Bombay, or Moscow and Bombay, etc. etc. It is totally unfeasible to explicitly represent all such negative information in the data base, as would be required under a first order theorem prover. It is important to notice, however, that the closed world assumption presumes perfect knowledge about the domain being modeled. If it were not known, for example, whether Air Canada directly connects Vancouver and Chicago, we would no longer be justified in making the closed world assumption with respect to the flight schedule. For by the absence of this fact from the data base, we would conclude that Air Canada does not directly connect Vancouver and Chicago, violating our assumed state of ignorance about this fact.

The flight schedule illustrates a very common use of the closed world default rule for purely extensional data bases. In particular, it illustrates how this default factors out the need for any explicit representation of negative facts. This result holds for more general data bases. As an example, consider the ubiquitous blocks world, under the following decomposition hierarchy of objects in that world:



Let SUPPORTS(x,y) denote "x directly supports y" and FREE(x) denote "x is free" i.e. objects may be placed upon x. Then the following general facts hold:

(x/OBJECT)(y/TABLE)¬SUPPORTS(x,y) (1)

(x/OBJECT)¬SUPPORTS(x,x) (2)

(x/PYRAMID)(y/BLOCK)¬SUPPORTS(x,y) (3)

(x y/BLOCK)SUPPORTS(x,y)
¬SUPPORTS(y,x) (4)

(x/PYRAMID)¬FREE(x) (5)

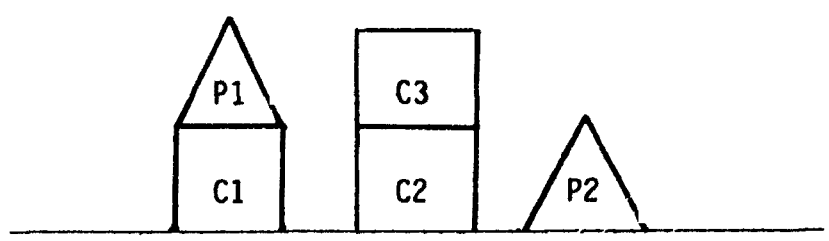
(x y/BLOCK)(z/TABLE)SUPPORTS(x,y)
¬SUPPORTS(z,y) (6)

(x/CUBE)FREE(x) ⊃
(y/BLOCK)¬SUPPORTS(x,y) (7)

(x/CUBE)(y/BLOCK)¬SUPPORTS(x,y) ⊃
FREE(x) (8)

(x/TABLE)FREE(x) (9)

Consider the following scene



This is representable by

$$\left.\begin{array}{ll} \text{SUPPORTS(T,C1)} & \text{SUPPORTS(T,C2)} \\ \text{SUPPORTS(C1,P1)} & \text{SUPPORTS(C2,C3)} \\ \text{SUPPORTS(T,P2)} & \end{array}\right\} \quad (10)$$

together with the following negative facts

$$\left.\begin{array}{ll} \neg\text{SUPPORTS(C1,C2)} & \neg\text{SUPPORTS(C2,C1)} \\ \neg\text{SUPPORTS(C3,C1)} & \neg\text{SUPPORTS(C1,P2)} \\ \neg\text{SUPPORTS(C3,P1)} & \neg\text{SUPPORTS(C3,P2)} \\ \neg\text{SUPPORTS(C1,C3)} & \neg\text{SUPPORTS(C2,P1)} \end{array}\right\} \quad (11)$$

Notice that virtually all of the knowledge about the blocks domain is negative, namely the negative specific facts (11), together with the negative facts (1)-(7)[1] This is not an accidental feature. Most of what we know about any world is negative.

Now a first order theorem prover must have access to all of the facts (1)-(11). For example, in proving ¬SUPPORTS(C3,C2) it must use (4). Consider instead such a theorem prover endowed with the additional ability to interpret the closed world default schema (D3). Then, in attempting a proof of ¬SUPPORTS(C3,C2) it tries to show that SUPPORTS(C3,C2) is not provable. Since SUPPORTS(C3,C2) cannot be proved, it concludes ¬SUPPORTS(C3,C2), as required.

It should be clear intuitively that in the presence of the closed world default schema (D3), none of the negative facts (1)-(7), (11) need be represented explicitly nor used in reasoning. This can be proved, under fairly general conditions [Reiter 1978]. One function, then, of the closed world default is to "factor out" of the representation all negative knowledge about the domain. It is of some interest to compare the blocks world representation (1)-(11) with those commonly used in blocks world problem-solvers (e.g.[Winograd 1972, Warren 1974]). These systems do not represent explicitly the negative knowledge (1)-(7), (11) but instead use the closed world default for reasoning about negation. (See Section 3 below for a discussion of negation in A.I. programming languages.)

Although the closed world default factors out negative knowledge for answering questions about a domain, this knowledge must nevertheless be available. To see why, consider an attempted update of the example blocks world scene with the new "fact" SUPPORTS(C3,C2). To detect the resulting inconsistency requires the negative fact (4). In general then, negative knowledge is necessary for maintaining the integrity of a data base. A consequence of the closed world assumption is a decomposition of knowledge into positive and negative facts. Only positive knowledge is required for querying the data base. Both positive and negative knowledge are required for maintaining the integrity of the data base.

2.3 DEFAULTS AND THE FRAME PROBLEM

The frame problem [Raphael 1971] arises in the representation of dynamic worlds. Roughly speaking, the problem stems from the need to represent those aspects of the world which remain invariant under certain state changes. For example, moving a particular object or switching on a light will not change the colours of any objects in the world. Painting an object will not affect the locations of the objects. In a first order representation of such worlds, it is necessary to represent explicitly all of the invariants under all state changes. These are referred to as the frame axioms for the world being modeled. For example, to represent the fact that painting an object does not alter the locations of objects would require, in the situational calculus of [McCarthy and Hayes 1969] a frame axiom something like

$$(x/\text{OBJECT})(y/\text{POSITION})(s/\text{STATE})(C/\text{COLOUR})$$
$$\text{LOCATION}(x,y,s) \supset \text{LOCATION}(x,y,\text{paint}(z,C,s))$$

The problem is that in general we will require a vast number of such axioms e.g. object locations also remain invariant when lights are switched on. when it thunders, when someone speaks etc. so there is a major difficulty in even articulating a deductively adequate set of frame axioms for a given world.

A solution to the frame problem is a representation of the world coupled with appropriate rules of inference such that the frame axioms are neither represented explicitly nor used explicitly in reasoning about the world. We will focus on a

[1] The notion of a negative fact has a precise definition. A fact is negative iff all of the literals in its clausal form are negative.

proposed solution by [Sandewall 1972][1]. A related approach is described in [Hayes 1973]. Sandewall proposes a new operator, UNLESS, which takes formula W as argument. The intended interpretation of UNLESS(W) is "W can not be proved" i.e. it is identical to the operator $\not\vdash$ of this paper. Sandewall proposes a single "frame inference rule" which, in the notation of this paper, can be paraphrased as follows:

For all predicates P which take a state variable as an argument

$$(x_1/\tau_1)\ldots(x_n/\tau_n)(s/\text{STATE})(f/\text{ACTION-FUNCTION})\quad \frac{\not\vdash \neg P(x_1,\ldots,x_n,f(x_1,\ldots,x_n,s))}{P(x_1,\ldots,x_n,f(x_1,\ldots,x_n,s))} \qquad (D4)$$

Intuitively, (D4) formalizes the so-called "STRIPS assumption" [Waldinger 1975]: Every action (state change) is assumed to leave every relation unaffected unless it is possible to deduce otherwise. This schema can be used in the following way, say in order to establish that cube33 is at location λ after box7 has been painted blue:

To establish LOCATION(cube33,λ,paint(box7,blue,s))
fail to prove $\neg$LOCATION(cube33,λ,paint(box7,blue,s))

There are several observations that can be made:

1. The frame inference schema (D4) has a pattern similar to the default schemata (D2) and (D3) of earlier sections of this paper. It too is a default schema.
2. The frame schema (D4) is in some sense a dual of the closed world schema (D3). The former permits the deduction of a positive fact from failure to establish its negation. The latter provides for the deduction of a negative fact from failure to derive its positive counterpart. This duality is preserved with respect to the knowledge "factored out" of the representation. Whereas the frame default eliminates the need for certain kinds of positive knowledge (the frame axioms), the closed world default factors out the explicit representation of negative knowledge.

2.4 DEFAULTS AND EXCEPTIONS

A good deal of what we know about the world is "almost always" true, with a few exceptions. For example, all birds fly except for penguins, ostriches, fledglings, etc. Given a particular bird, we will conclude that it flies unless we happen to know that is satisfies one of these exceptions. Nevertheless, we want it true of birds "in general" that they fly. How can we reconcile these apparently conflicting points of view? The natural first order representation is inconsistent:

(x/BIRD)FLY(x) "In general, birds fly"
(x)PENGUIN(x) $\supset$ BIRD(x) "Penguins are birds
(x/PENGUIN)$\neg$FLY(x) which don't fly."

An alternative first order representation explicitly lists the exceptions to flying

$$(x/\text{BIRD})\neg\text{PENGUIN}(x) \wedge \neg\text{OSTRICH}(x) \wedge \ldots \supset \text{FLY}(x)$$

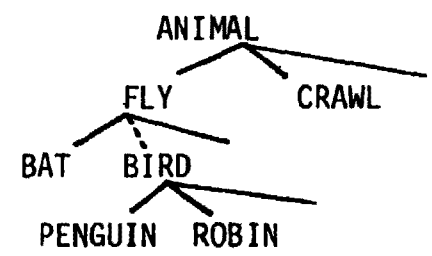
But with this representation, we cannot conclude of a "general" bird, that it can fly. To see why, consider an attempt to prove FLY(tweety) where all we know of tweety is that she is a bird. Then we must establish the subgoal

$$\text{PENGUIN}(\text{tweety}) \wedge \neg\text{OSTRICH}(\text{tweety}) \wedge \ldots$$

which is impossible given that we have no further information about tweety. We are blocked from concluding that tweety can fly even though, intuitively we want to deduce just that. In effect, we need a default rule of the form

$$(x/\text{BIRD})\quad \frac{\not\vdash (\text{PENGUIN}(x) \vee \text{OSTRICH}(x) \vee \ldots)}{\text{FLY}(x)}$$

With this rule of inference we can deduce FLY(tweety), as required. Notice, however, that whenever there are exceptions to a "general" fact in some domain of knowledge we are no longer free to arbitrarily structure that knowledge. For example, the following hierarchy would be unacceptable, where the dotted link indicates the existence of an exception



Clearly there is no way in this hierarchy of establishing that penguins are animals. For hierarchies the constraint imposed by exceptions is easily

[1] [Kramosil 1975] claims to have proved that Sandewall's approach is either meaningless or equivalent to a first order approach. See Section 4 for a discussion of this issue.

articulated: If P and Q are nodes with P below Q, and if $(x)P(x) \supset Q(x)$ is true without exception, then there must be a sequence of solid links connecting P and Q. For more general kinds of knowledge the situation is more problematic. One must be careful to ensure that chains of implications do not unwittingly inherit unintended exceptions.

3. DEFAULTS AND "NEGATION" IN A.I. PROGRAMMING LANGUAGES

It has been observed by several authors [Haye. 1973, Sandewall 1972, Reiter 1978] that the basic default operator M has, as its "procedural equivalent" the negation operator in a number of A.I. programming languages e.g. THNOT in MICROPLANNER [Hewitt 1972, Sussman et al. 1970], NOT in PROLOG [Roussel 1975]. For example, in MICROPLANNER, the command (THGOAL <pattern>) can be viewed as an attempt to prove <pattern> given a data base of facts and theorems. (THNOT(THGOAL <pattern>)) then succeeds iff (THGOAL <pattern>) fails i.e. iff <pattern> is not provable, and this of course is precisely the interpretation of the default operator M.

Given that "negation" in A.I. procedural languages corresponds to the default operator and not to logical negation, it would seem that some of the criticism often directed at theorem proving from within the A.I. community is misdirected. For the so-called procedural approach, often proposed as an alternative to theorem proving as a representation and reasoning component in A.I. systems, is a realization of a default logic, whereas theorem provers are usually realizations of a first order logic, and as we have seen, these are different logics.

In a sense, the so-called procedural vs. declarative issue in A.I. might better be phrased as the default vs. first order logic issue. Many of the advantages of the procedural approach can be interpreted as representational and computational advantages of the default operator. There is a fair amount of empirical evidence in support of this point of view, primarily based upon the successful use of PROLOG [Roussel 1975] - a pure theorem prover augmented with a "THNOT" operator for such diverse A.I. tasks as problem solving [Warren 1974], symbolic mathematics [Kanoui 1976], and natural language question-answering [Colmerauer 1973].

On the theoretical level, we are just beginning to understand the advantages of a first order logic augmented with the default operator:

1. Default logic provides a representation language which more faithfully reflects a good deal of common sense knowledge than do traditional logics. Similarly, for many situations, default reasoning corresponds to what is usually viewed as common sense reasoning.
2. For many settings, the appropriate default theories lead to a significant reduction in both representational and computational complexity with respect to the corresponding first order theory. Thus, under the closed world default, negative knowledge about a domain need not explicitly be represented nor reasoned with in querying a data base. Similarly under the frame default, the usual frame axioms are not required.

There are, of course, other advantages of the procedural approach - specifically, explicit control over reasoning - which are not accounted for by the above logical analysis. We have distinguished the purely logical structure of such representational languages from their process structure, and have argued that at least some of their success derives from the nature of the logic which they realize.

4. SOME PROBLEMS WITH DEFAULT THEORIES

Given that default reasoning has such widespread applications in A.I. it is natural to define a default theory as a first order theory augmented by one or more inference schemata like (D1), (D2) etc. and to investigate the properties of such theories. Unfortunately, some such theories display peculiar and intuitively unacceptable behaviours.

One difficulty is the ease with which inconsistent theories can be defined, for example $\frac{M A}{B}$ coupled with a knowledge base with the single fact $\neg B$. Another, pointed out by [Sandewall 1972] is that the theorems of certain default theories will depend upon the order in which they are derived As an example, consider the theory

$$\frac{M A}{B} \qquad \frac{M B}{A}$$

Since A is not provable, we can infer B. Since B

is now proved, we cannot infer A, so this theory has the single theorem B. I instead, we had started by observing that B is not provable, then the theory would have the single theorem A. Default theories exhibiting such behaviour are clearly unacceptable. At the very least, we must demand of a default theory that it satisfy a kind of Church-Rosser property: No matter what the order in which the theorems of the theory are derived, the resulting set of theorems will be unique.

Another difficulty arises in modeling dynamically changing worlds e.g. in causal worlds or in text understanding where the model of the text being built up changes as more of the text is assimilated. Under these circumstances, inferences which have been made as a result of a default assumption may subsequently be falsified by new information which now violates that default assumption. As a simple example, consider a travel consultant which has made the default assumption that the traveller's starting point is Palo Alto and has, on the basis of this, planned all of the details of a trip. If the consultant subsequently learns that the starting point is Los Angeles, it must undo at least part of the planned trip, specifically the first (and possibly last) leg of the plan. But how is the consultant to know to focus just on these changes? Somehow, whenever a new fact is deduced and stored in the data base, all of the facts which rely upon a default assumption and which supported this deduction must be associated with this new fact. These supporting facts must themselves have their default supports associated with them, and so on. Now, should the data base be updated with new information which renders an instance of some default rule inapplicable, delete all facts which had been previously deduced whose support sets relied upon this instance of the default rule. There are obviously some technical and implementation details that require articulation, but the basic idea should be clear. A related proposal for dealing with beliefs and real world observations is described in [Hayes 1973].

One way of viewing the role of a default theory is as a way of implicitly further completing an underlying incomplete first order theory. Recall that a first order theory is said to be complete iff for all closed formulae W, wither W or ¬W is provable. Most interesting mathematical theories turn out to be incomplete - a celebrated result due to Godel. Most of what we know about the world, when formalized, will yield an incomplete theory precisely because we cannot know everything - there are gaps in our knowledge. The effect of a default rule is to implicitly fill in some of those gaps by a form of plausible reasoning. In particular, the effect of the closed world default is to fully complete an underlying incomplete first order theory. However, it is well known that there are insurmountable problems associated with completing an incomplete theory like arithmetic. Although it is a trivial matter conceptually to augment the axioms of arithmetic with a default rule $\frac{\vdash W}{W}$ where W is any closed formula, we will be no further ahead because the non theorems of arithmetic are not recursively enumerable. What this means is that there is no way in general that, given a W, we can establish that W is not a theorem even if W happens not to be a theorem. This in turn means that we are not even guaranteed that an arbitrary default rule of inference is effective i.e. there may be no algorithm which will inform us whether or not a given default rule of inference is applicable! From this we can conclude that the theories of a default theory may not be recursively enumerable. This situation is in marked contrast to what normally passes for a logic where, at the very least, the rules of inference must be effective and the theorems recursively enumerable.

Finally, it is not hard to see that default theories fail to satisfy the extension property [Hayes 1973] which all "respectable" logics do satisfy. (A logical calculus has the extension property iff whenever a formula is provable from a set P of premises, it is also provable from any set P' such that $P \subseteq P'$.)

[Kramosil 1975] attempts to establish some general results on default theories. Kramosil "proves" that for any such theory, the default rules are irrelevant in the sense that either the theory will be meaningless or the theorems of the theory will be precisely the same as those obtainable by ignoring the default rules of inference. Kramosil's result, if correct, would invalidate the

main point of this paper, namely that default theories play a prominent role in reasoning about the world. Fortunately, his "proof" relies on an incorrect definition of theoremhood so that the problem of characterizing the theorems of a default theory remain open.

CONCLUSIONS

Default reasoning may well be the rule, rather than the exception, in reasoning about the world since normally we must act in the presence of incomplete knowledge. Moreover, aside from mathematics and the physical sciences, most of what we know about the world has associated exceptions and caveats. Conventional logics, such as first order logic, lack the expressive power to adequately represent the knowledge required for reasoning by default. We gain this expressive power by introducing the default operator.

In order to provide an adequate formal (as opposed to heuristic) foundation for default reasoning we need a well articulated theory of default logic. This requires, in part, a theory of the semantics of default logic, a suitable notion of theoremhood and deduction, and conditions under which the default inference rules are effective and the set of theorems unique. Since in any realistic domain, all of the default schemata of Section 2 will be in force (together, no doubt, with others we have not considered) we require a deeper understanding of how these different schemata interact. Finally, there is an intriguing relationship between certain defaults and the complexity of the underlying representation. Both the closed world and frame defaults implicitly represent whole classes of first order axioms. Is this an accidental phenomemon or is some general principal involved?

ACKNOWLEDGEMENTS

This paper was written with the financial support of NRC grant A 7642. I am indebted to Brian Funt, Randy Goebel and Richard Rosenberg for their criticisms of an earlier draft of this paper.

REFERENCES

Bobrow, D.G. and Winograd, T., (1977). "An Overview of KRL-0, a Knowledge Representation Language," Cognitive Science, Vol.1, No.1, Jan. 1977.

Colmerauer, A., (1973). Un System de Communication Home-Machine en Français, Rapport interne, UER de Luminy, Universite d'Aix-Marseille, 1973.

Hayes, P.J., (1973). "The Frame Problem and Related Problems in Artificial Intelligence," in Artificial and Human Thinking, A. Elithorn and D. Jones (Eds.), Jossey-Bass Inc., San Francisco, 1973, pp.45-49.

Hayes, P.J., (1977). "In Defence of Logic," Proc. IJCAI-5, M.I.T., Cambridge, Mass., August 22-25, 1977, pp. 559-565.

Hewitt, C., (1972). Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot, A.I.Memo No. 251, M.I.T. Project MAC, Cambridge, Mass., April 1972.

Kanoui, H., (1976). "Some Aspects of Symbolic Integration via Predicate Logic Programming," SIGSAM Bulletin, 10, Nov. 1976, pp. 29-42.

Kramosil, I., (1975). "A Note on Deduction Rules with Negative Premises," Proc. IJCAI-4, Tbilisi, USSR, Sept. 3-8, 1975, pp. 53-56.

McCarthy J. and Hayes, P.J., (1969). "Some Philosophic Problems from the Standpoint of Artificial Intelligence," in Machine Intelligence 4, B. Meltzer and D. Michie (Eds.), Edinburgh University Press, Edinburgh, 1969, pp. 463-502.

Raphael, B., (1971). "The Frame Problem in Problem-Solving Systems," in Artificial Intelligence and Heuristic Programming, N.V. Findler and B. Meltzer (Eds.), Edinburgh University Press, Edinburgh.

Reiter, R., (1978). "On Closed World Data Bases," in Logic and Data Bases, H. Gallaire and J. Minker (Eds.), Plenum Press, New York, to appear.

Roberts, R.B. and Goldstein, I., (1977). The FRL Manual, A.I. Memo No. 409, M.I.T., Sept. 1977.

Roussel, P., (1975). PROLOG, Manuel de Reference et d'Utilisation, Group d'Intelligence Artificielle. U.E.R. de Marseille, France, 1975.

Sandewall, E., (1972). "An Approach to the Frame Problem, and its Implementation," in Machine Intelligence 7, B. Meltzer and D. Michie (Eds.), Edinburgh University Press, Edinburgh, pp. 195-204.

Sussman, G., Winograd, T., and Charniak,E., (1970). MICRO-PLANNER Reference Manual, A.I. MEMO No. 203, M.I.T., Cambridge, Mass., 1970.

Waldinger, R., (1975). Achieving Several Goals Simultaneously, Artificial Intelligence Center Technical Note 107, Stanford Research Institute, Menlo Park, Calif., July 1975.

Warren, D., (1974). WARPLAN: A System for Generating Plans, Memo No. 76, Dept. of Computational Logic, University of Edinburgh, June 1974.

Winograd, T., (1972). Understanding Natural Language, Academic Press, New York, 1972.

Woods, W.A., (1968). "Procedural Semantics for a Question-Answering Machine," AFIPS Conference Proceedings, Vol. 3, Part I, 1968, pp. 457-471.

PATH-BASED AND NODE-BASED INFERENCE IN SEMANTIC NETWORKS

Stuart C. Shapiro

Department of Computer Science
State University of New York at Buffalo
Amherst, New York 14226

Abstract

Two styles of performing inference in semantic networks are presented and compared. Path-based inference allows an arc or a path of arcs between two given nodes to be inferred from the existence of another specified path between the same two nodes. Path-based inference rules may be written using a binary relational calculus notation. Node-based inference allows a structure of nodes to be inferred from the existence of an instance of a pattern of node structures. Node-based inference rules can be constructed in a semantic network using a variant of a predicate calculus notation. Path-based inference is more efficient, while node-based inference is more general. A method is described of combining the two styles in a single system in order to take advantage of the strengths of each. Applications of path-based inference rules to the representation of the extensional equivalence of intensional concepts, and to the explication of inheritance in hierarchies are sketched.

1. Introduction

Semantic networks have developed since the mid sixties [10;11] as a formalism for the representation of knowledge. Methods have also been developing for performing deductive inference on the knowledge represented in the network. In this paper, we will compare two styles of inference that are used in semantic networks, path-based inference and node-based inference. In sections 2 and 3, these terms will be explained and references to systems that use them will be provided. In sections 4 and 5, the advantages and disadvantages of each will be discussed. Sections 6, 7 and 8 will show how they can be used to complement each other in a single semantic network system, how path-based inference can help represent the extensional equivalence of intensional concepts, and how a formalism for writing path-based inference rules can be used to explicate the notion of "inheritance" in a semantic network.

2. Path-Based Inference

Let us refer to a relation (perforce binary) that is represented by an arc in a network as an arc-relation. If R is an arc-relation, an arc labelled R from node a to node b represents that the relationship aRb holds. It may be that this arc is not present in the network, but aRb may be inferred from other information present in the network and one or more inference rules. If the other information in the network is a specified path of arcs from a to b, we will refer to the inference as *path-based*. The ways in which such paths may be specified will be developed as this paper proceeds.

The two clearest examples of the general use of path-based inference are in SAMENLAQ II [18] and Protosynthex III [13]. Both these systems use what might be called "relational" networks rather than "semantic" networks since arc-relations include conceptual relations as well as structural relations (see [14] for a discussion of the difference). For example, in Protosynthex III there is an arc labelled COMMANDED from the node representing Napoleon to the node representing the French army, and in SAMENLAQ II an arc labelled EAST.OF goes from the node for Albany to the node for Buffalo. Both systems use relational calculus expressions to form path-based inference rules. The following relational operators are employed (we here use a variant of the earlier notations):

1. Relational Converse -- If R is a relation, R^C is its converse. So, $\forall x,y\,(xR^Cy \leftrightarrow yRx)$.
2. Relational Composition -- If R and S are relations, R/S is R composed with S. So, $\forall x,y\,(xR/Sy \leftrightarrow \exists z\,(xRz \,\&\, zSy))$.
3. Domain Restriction -- If R and S are relations, $(S\ z)R$ is the relation R with its domain restricted to those objects that bear the relation S to z. So, $\forall x,y,z\,(x(S\ z)Ry \leftrightarrow (xSz \,\&\, xRy))$.
4. Range Restriction -- If R and S

are relations, $R(S\ z)$ is the relation R with its range restricted to those objects that bear the relation S to z. So, $\forall x,y,z(xR(S\ z)y \leftrightarrow (xRy \ \& \ ySz))$.

5. Relational Intersection -- If R and S are relations, $R\&S$ is the intersection of R and S. So, $\forall x,y(xR\&Sy \leftrightarrow (xRy \ \& \ xSy))$.

Notice that $\forall Q,R,S,x,y,z(xR(Q\ z)/Sy \leftrightarrow xR/(Q\ z)Sy)$ so we can use the notation $R(Q\ z)S$ unambiguously.

In SAMENLAQ II, path-based inference rules are entered by using the relational operators to give alternate definitions of simple arc labels. For example (again in a variant notation):

EAST.OF → EAST.OF/EAST.OF
declares that EAST.OF is transitive

SOUTH.OF → NORTH.OF^C
declares that
$\forall x,y(y\text{NORTH.OF}x \rightarrow x\text{SOUTH.OF}y)$

FATHER.OF → (GENDER MALE)PARENT.OF
declares that a father is a male parent.

SIR [11] is another relational network system that uses path-based inference. Although the original expressed inference rules in the form of general LISP functions, the reproduction in [16, Chap. 7] uses the notion of path grammars. The relation operators listed above are augmented with R^*, meaning zero or more occurrences of R composed with itself, R^+, meaning one or more occurrences of R composed with itself, and $R \vee S$, meaning the union of R and S. The following relations are used:

x EQUIV y	means x and y are the same individual
x SUBSET y	means x is a subset of y
x MEMBER y	means x is a member of the set y
x POSSESS y	means x owns a member of the set y
x POSSESS-BY-EACH y	means every member of the set x owns a member of the set y.

To determine if x POSSESS y, the network is searched using the following rule:

POSSESS → EQUIV*
/(POSSESS
∨(MEMBER/SUBSET*/POSSESS-BY-EACH))
/SUBSET*

The widest use of path-based inference is in ISA hierarchies. Fig. 1 is based on probably the most famous ISA hierarchy, that of Collins and Quillian [2]. The two important rules here are

ISA → ISA*
and PROP → ISA*/PROP

As McDermott [8] points out, ISA hierarchies have been abused as well as used. In Section 8, we will propose a method authors can use to describe their hierchies precisely.

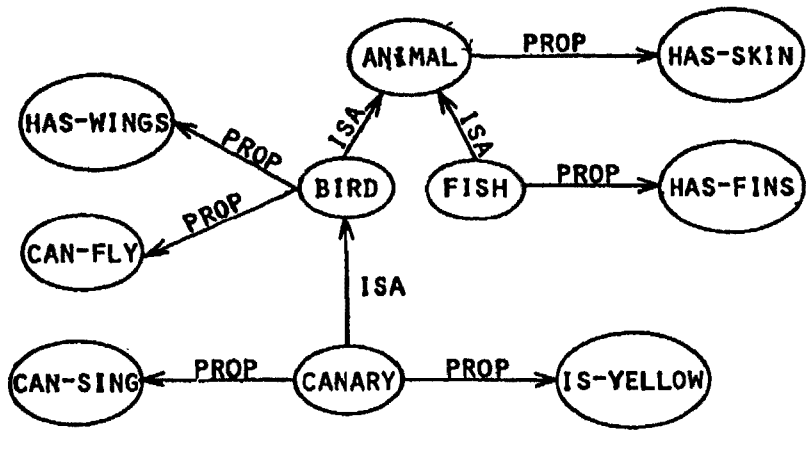


FIGURE 1: ISA hierarchy based on that of Collins and Quillian

3. Node-Based Inference

Several semantic network systems incorporate methods of representing general rules in a semantic network version of predicate calculus. Among these systems are those of Shapiro [14;15;17], Kay [7], Hendrix [6], Schubert [12], and Fikes and Hendrix [3]. Figure 2 shows such a network deduction rule representing

$\forall x[x \varepsilon \text{MAN} \rightarrow \exists y(y \varepsilon \text{WOMAN} \ \& \ x\text{LOVES}y)]$.

Figure 3 shows a rule for

$\forall r[r \varepsilon \text{TRANSITIVE} \rightarrow \forall x,y,z(xry \ \& \ yrz \rightarrow xrz)]$.

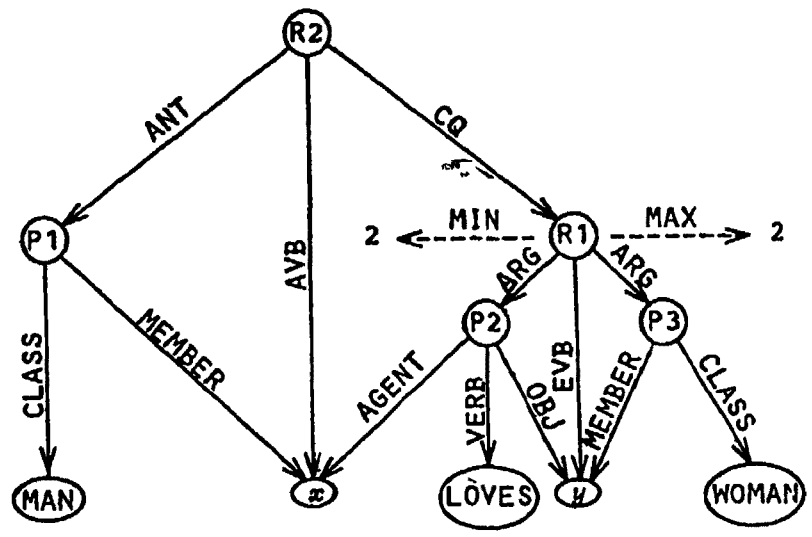


FIGURE 2: A semantic network deduction rule for $\forall x[x \varepsilon \text{MAN} \rightarrow \exists y(y \varepsilon \text{WOMAN} \ \& \ x\text{LOVES}y)]$

The network formalism employed is that of Shapiro [15;17]. These deduction rules employ pattern nodes (P1, P2, P3, P4, P5 P6, P7), each one of which represents a pattern of nodes that might occur in the network. We will therefore call this kind of inference rule a *node-based* inference rule. Pattern nodes are related to each other by *rule nodes*, each of which represent a propositional operator, or, equivalently, an inference mechanism. For example, R2 represents the rule that if an instance of P1 occurs in the network, an instance of R1 with the same substitution

for x may be deduced. Quantification is represented in this notation by an arc-relation between a rule node and the variable nodes bound in the rule. For example, x is bound by a universal quantifier in R2 and y is bound by an existential quantifier in R1.

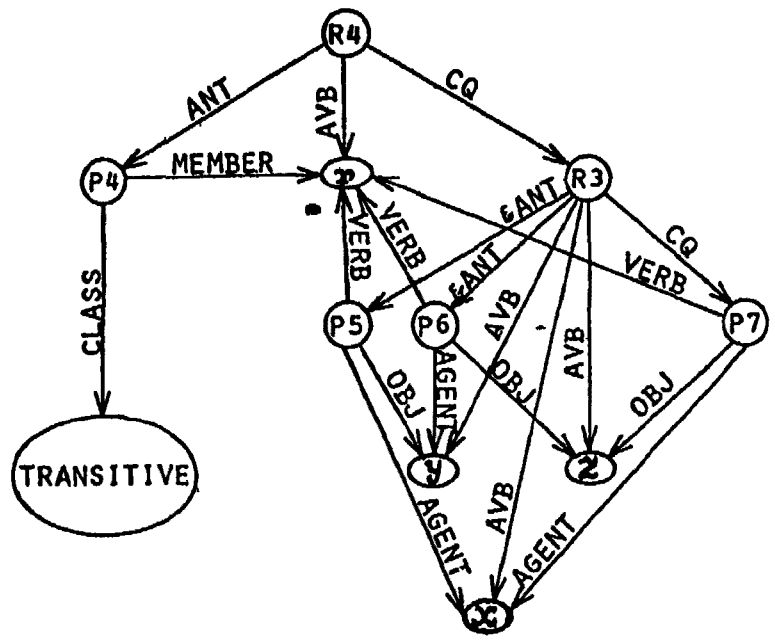


FIGURE 3: A semantic network deduction rule for $\forall r[r \varepsilon \text{TRANSITIVE} \rightarrow \forall x,y,z(xry \ \& \ yrz \rightarrow xrz)]$

To see how a node-based inference proceeds, consider the network of Figure 4 in conjunction with the rule of Figure 3, and say that we wish to decide if A SUPPORTS C. The network that would represent that A SUPPORTS C matches P7 with the variable binding [x/A, r/SUPPORTS, z/C]. P4 in the binding [r/SUPPORTS] is matched against the network and is found to successfully match M1. P5[x/A, r/SUPPORTS, y/y] and P6[y/y, r/SUPPORTS, z/C] are then both matched against the network and each succeeds with a consistent binding of y to B. The rule thus succeeds and A SUPPORTS C is deduced. (Details of the bindings and the match routine are given in [15].)

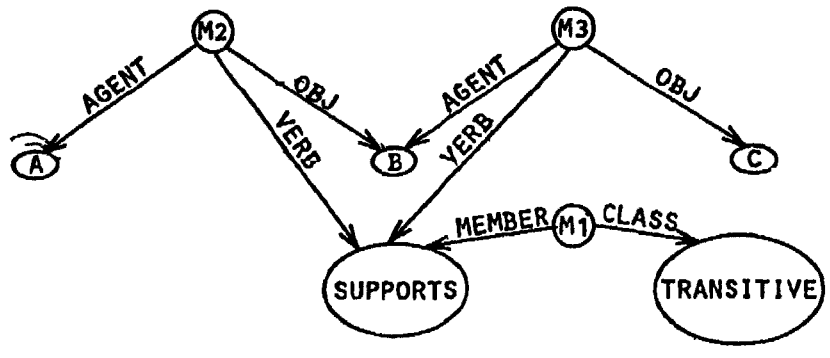


FIGURE 4: A network data base asserting that A SUPPORTS B, B SUPPORTS C and SUPPORTS ε TRANSITIVE.

It should be noted that set inclusion was represented by an arc (ISA) in Section 2, but set membership is being represented by a node (with a MEMBER, CLASS "case frame") in this section. The nodal representation is required by node-based inference rules and is consistent with the notion that everything that the network "knows", and every concept to which the network can refer is represented by a node.

4. Advantages of Node-Based Inference

The advantages of node-based inference stem from the generality of the syntax of node-based inference rules. Path-based rules are limited to binary relations, have a restricted quantification structure and require that an arc between two nodes be implied by a path between the same two nodes. Rule R2 of Figure 2 could not be written as a path-based rule, and, although the transitivity of SUPPORTS could be expressed by a path-based rule (SUPPORTS $\leftarrow$ SUPPORTS$^+$), the "second order" rule R4 of Figure 3 could not.

Let us briefly consider how rule R4 is constructed, whether it really is or is not a second order rule, and why it could not be expressed as a path-based rule. Rule R4 supplies a rule for use with transitive relations. In order to assert that a relation is transitive (e.g. assertion node M1 of Figure 4), the relation must be represented as a node, rather than as an arc. This also allows quantification over such relations, since in all node-based inference rule formalisms variables may only be substituted for nodes, not for arcs. Since the relation is a node, another node must be used to show the relationship of the relation to its arguments (e.g. nodes M2 and M3 in Figure 4). Thus, R4 is really a first order rule derived from the second order rule $\forall r[r \varepsilon \text{TRANSITIVE} \rightarrow \forall x,y,z(xry \ \& \ yrz \rightarrow xrz)]$ by reducing r to an individual variable and introducing a higher order relation, AVO, whose second argument is a conceptual relation and whose other arguments are conceptual individuals. So R4 is more accurately seen as the first order rule

$$\forall r[r \varepsilon \text{TRANSITIVE} \rightarrow \forall x,y,z(\text{AVO}(x,r,y) \ \& \ \text{AVO}(y,r,z) \rightarrow \text{AVO}(x,r,z))].$$

In this view, the predicates of semantic networks are not the nodes representing conceptual relations, but the different case frames. Rule R4 cannot be represented as a path-based rule because it is a rule about the relation AVO, and AVO is a trinary, rather than a binary relation.

Although some node-based inference rules cannot be expressed by path-based inference rules, it is easy to see that any path-based inference rule can be expressed by a node-based inference rule, as long as we are willing to replace some arc-relations by nodes and higher order predicates.

5. Advantages of Path-Based Inference

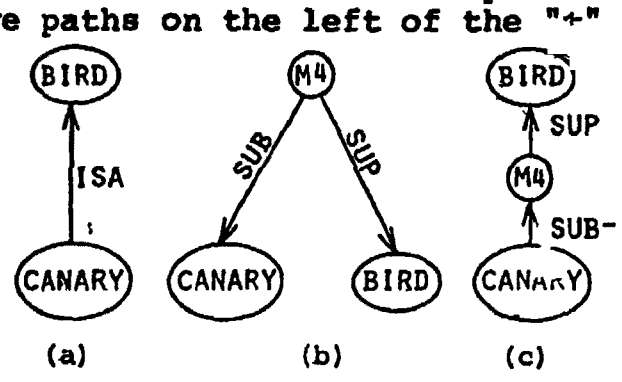
The major advantage of path-based inference is efficiency. Carrying out a path-based inference involves merely checking that a specified path exists in the network between two given nodes (plus,

perhaps, some side paths to specified nodes required by domain and range restrictions). This is a well understood and relatively efficient operation, especially compared to the backtracking, intersection, or unification operations required to check the consistency of variable substitutions in node-based inference rules.

Moreover, path following seems to many people to be what semantic networks were originally designed for. The major search algorithm of Quillian's Semantic Memory is a bi-directional search for a path connecting two nodes [10, p. 249]. Also, the ability to do path tracing is a motivation underlying ISA hierarchies, and is why the Collins and Quillian results [2] gained such attention. These efficiencies are lost by replacing path-based inference rules by node-based inference rules.

6. Combining Path-Based and Node-Based Inference

We begin the task of unifying path-based and node-based inferences by noting the formal equivalence between an arc-relation and a two case case frame. Figure 5 illustrates this using ISA vs. SUB-SUP. Figure 5a shows the use of the ISA arc-relation to represent that canaries are birds. Figure 5b represents the same relationship by a SUB-SUP case frame, and has the advantage that the relationship is represented by a node, M4. Figure 5c is a redrawing of 5b, using the arc label SUB- to represent the relation SUB^C. (It is generally understood in semantic network formalisms that whenever an arc representing a relation R goes from some node n to some node m, there is also an arc representing R^C going from m to n). Figure 5c clarifies the notion that we may think of an instance of a two case case frame (such as M4) as both an arc and a node if we are willing to recalibrate the measurement of time it takes to follow one arc-relation to be the time it takes to follow two arcs. We can replace all instances of ISA in the path-based inference rules of Section 2 by the composition SUB-/SUP and still have valid rules except that we now have paths on the left of the "←" symbol.



FIGUPE 5: An illustration of the equivalence of an arc-relation to a two case case frame. a) Representing set membership as the ISA arc-relation. b) Representing set membership as a SUB-SUP case frame. c) Redrawing (b) so it looks like (a).

Let us, therefore, extend our syntax of path-based inference rules to allow a path of arc compositions on the left of the "←" symbol. The rule ISA ← ISA* states that whenever there is a path of ISA arcs from node n to mode m, we can infer a "virtual" ISA arc directly from n to m which we may, if we wish, actually add to the network. Similarly, let the rule SUB-/SUP ← (SUB-/SUP)* state that whenever a path of alternating SUB- and SUP arcs goes from node n to node m, we can infer a "virtual" node with SUB to n and SUP to m which we may, if we wish, actually add to the network.

We now have a formalism for specifying path-based inference rules in a network formalism that represents binary conceptual relations by two case case frames. This would allow, for example, for a more unified representation in the SNIFFER system [3], in which node-based inference rules are implemented and built-in path based inference rules are used for set membership and set inclusion, both of which are represented only by arc-relations. The formalism presented here would allow set membership and set inclusion assertions to be represented by nodes, permitting other assertions to reference them, without giving up the efficiency of built-in routines to implement the set inclusion hierarchies.

We desire, however, a more general unification of path-based and node-based inferences. There are two basic routines used to implement node-based inferences (although specific implementations may differ). One is the match routine that is given a pattern node and finds instances of it in the network, and the other is the routine that interprets the quantifiers and connectives to carry out the actual deduction. The match routine can be enhanced to make use of path-based inference rules. Consider a typical match routine used in the deduction in Section 3 of A SUPPORTS C from the network of Figure 4 and the rule of Figure 3, and let us introduce the notation that if P is a path of arcs and n is a node, $P[n]$ represents the set of nodes found by following the path P from the node n. In the example, the match routine was given the pattern P4 to match in the binding [r/SUPPORTS]. It was able to find M1 by intersecting $CLASS^C$[TRANSITIVE] with $MEMBER^C$[SUPPORTS]. Now, let us suppose that the path-based inference rule CLASS ← CLASS/(SUB-/SUP)* has been declared in such a way that the match routine could use it. The match routine would intersect $MEMBER^C$[SUPPORTS] with $(CLASS/(SUB-/SUP)^*)^C$[TRANSITIVE] and be able to find a virtual node asserting that SUPPORTS is TRANSITIVE even if a long chain of set inclusions separated them. The proposal, therefore, is this: any arc-relation in a semantic network may be defined in terms of a path-based inference rule which the match routine is capable of using when finding instances of pattern

nodes. This completes the general unification of path-based and node-based inference we desired. Since path-based inference is embedded in the match routine, while node-based inference requires the quantifier and connective interpreter, the difference is reminiscent of the difference between subconscious inference and conscious reasoning.

7. Application to Extensional Equivalence of Intensional Concepts

A basic assumption of semantic networks is that each concept is represented by a single node and that all information about a concept is reachable from its node. Yet, since Woods' discussion [20], most semantic network authors have agreed that a node represents an intensional, rather than an extensional concept. How should we handle the information that two different intensional concepts are extensionally equivalent?

Let us illustrate this by a story (entirely fictional). For the last year we have heard of a renowned surgeon in town, Dr. Smith, known for his brilliance and dexterity, who saved the life of the famous actress Maureen Gelt by a difficult heart transplant operation. Meanwhile, at several social gatherings, we have met someone by the name of John Smith, about five feet, six inches tall, black hair and beard, generally disheveled and clumsy. We now discover, much to our amazement that John Smith and Dr. Smith are one and the same! In our semantic network, we have one node for Dr. Smith connected to his attributes, and another for John Smith connected to his attributes. What are we to do? Although we now know that John Smith saved the life of Maureen Gelt and that Dr. Smith has black hair, surely we cannot retrieve that information as fast as that Dr. Smith is a surgeon and that John Smith is 5'6" tall. If we were to combine the two nodes by taking all the arcs from one node, tying them to the other and throwing away the first, we would lose this distinction. We must introduce an assertion, say an EQUIV-EQUIV case frame, that represents the fact that Dr. Smith and John Smith, different intensional concepts, are extensionally the same.[1] How are we to use this assertion?

Ignoring for the moment referentially opaque contexts ("We didn't know that John Smith was Dr. Smith."), how can we express the rule that if n EQUIV-/EQUIV m, than anything true of n is true of m? Our node based inference rules cannot express this rule because expressing "anything true of n" requires quantifying over those higher order case frame predicates such as AVO and MEMBER-CLASS. One possibility is to use lambda abstraction as Schubert does [12]. Each n-ary higher order predicate involving some node becomes a unary predicate by replacing that node by a lambda variable. Thus, "Dr. Smith saved Maureen Gelt's life" becomes an instance of the unary predicate $\lambda(x)$[x saved Maureen Gelt's life] applied to Dr. Smith. Using a PRED-ARG case frame, it is easy to represent the rule

$$\forall x,y,z\,[\text{EQUIV-EQUIV}(x,y)\ \&\ \text{PRED-ARG}(x,z) \rightarrow \text{PRED-ARG}(y,z)].$$

The trouble with this solution is, how are we to retrieve this information as a fact about Maureen Gelt? Must we also store

$$\lambda(x)[\text{Dr. Smith saved } x\text{'s life}](\text{Maureen Gelt})?$$

This duplication is unsatisfying. An alternative is to include in the path-based inference rule defining each arc-relation the path (EQUIV-/EQUIV)*. For example, AGENT $\leftarrow$ AGENT/(EQUIV-/EQUIV)*, and CLASS $\leftarrow$ CLASS/((EQUIV-/EQUIV)*/(SUB-/SUP)*)*. Although this solution requires more rules than the lambda abstraction solution, and the rules look complicated, it avoids the duplication of the same assertion in different forms and the postulation of conceptual predicates such as $\lambda(x)$[x saved Maureen Gelt's life].

Hays' cognitive networks [4;5] include a scheme similar to the one proposed here. Each assertion about Dr. Smith would refer to a different node, each with an MST (manifestation) arc to a common node. This node would represent the intension of Dr. Smith, while the others represent Dr. Smith as surgeon, Dr. Smith as saviour of Maureen Gelt, etc. Presumably, when Hays' network learns of the identity of Dr. Smith with John Smith, a new node is introduced with MST arcs from both Dr. Smith and John Smith.[2] Dr. Smith and John Smith are then seen as two manifestations of the newly integrated Dr. John Smith. Hays presumably uses an MST*/(MST^C)* path where we propose an (EQUIV-/EQUIV)* path.

Blocking referentially opaque contexts seems to require introducing *relational complement*. For any path P and nodes x and y, let xPy hold just in case a path P from x to y does not exist in the network.* We might block referentially opaque contexts by including the domain or range restriction (OBJ-/VERB/MEMBER-/CLASS OPAQUE) in the arc definitions.

8. Application to the Explication of Inheritance

As was mentioned in Section 2, many

---

[1] The psychological plausibility of this discussion is supported by the experiments of Anderson and Hastie [1] and of McNabb [9].

[2] Actually, Hays' networks have not yet been implemented, and I have been warned [R. Fritzson, personal communication] that the implementation may differ from what I have supposed.

semantic networks include inheritance (ISA) hierarchies. Often these are at best vague and at worst inconsistent. We propose that the inheritance properties of these hierarchies be clearly defined by path-based inference rules using the syntax we are presenting here or some other well defined syntax. We do not say that all systems should be able to input and interpret such rules, but only that authors use such rules to explain clearly to their readers how their hierarchies work.

Before this proposal is feasible, we must be able to handle two more situations. The first is the exception principle, first expressed by Raphael [11, p.85] and succinctly stated by Winograd as, "Any property true of a concept in the hierarchy is implicitly true of anything linked below it, unless explicitly contradicted at the lower level" [19, p.197]. To allow for this, let us introduce an *exception operator*. If P and Q are paths and x and y are nodes, let $xP\backslash Qy$ hold just in case there is a path described by P from x to y and no path of equal or shorter length described by Q from x to y. To see that this suffices to handle the exception principle, consider the hierarchy of Figure 6, where, to make things more interesting, we have postulated a variety of flying penguins. We have also taken the liberty of explicitly representing that CAN-FLY and CAN-NOT-FLY are negations of each other. The rule for inheritance in this hierarchy is

$$\text{PROP} \leftarrow (\text{ISA}^*/\text{PROP})\backslash(\text{ISA}^*/\text{PROP}/\widehat{\text{NOT}}).$$

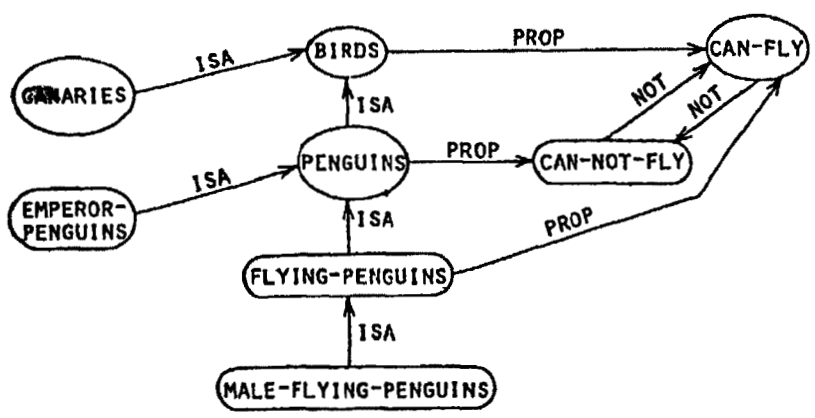


FIGURE 6 An ISA hierarchy illustrating the exception principle.

The other situation we must discuss is "almost transitive" relations such as SIBLING. SIBLING is certainly symmetric, but it cannot be transitive since it is irreflexive. Yet your sibling's sibling is your sibling as long as he/she is not yourself. This is what we mean by "almost transitive." Note that for any relation, R, $R^* \& \overline{(R^+)}$ is the identity relation. Let us call it I. Then for any relation P, let P^R be $P \& \overline{I}$. P^R is the *irreflexive restriction* of P. We can use this to define SIBLING as $\text{SIBLING} \leftarrow (\text{SIBLING} \vee \text{SIBLING}^C)^{*R}$.

We suggest that the syntax for path-based inference rules is now complete enough to explicate the inheritance rules of various hierarchies. The complete syntax will be summarized in the next section

9. Summary

We have presented and compared two styles of inference in semantic networks, path-based inference and node-based inference. The former is more efficient, while the latter is more general. We showed the equivalence of an arc-relation to a two case case frame, and described how path-based inference could be incorporated into the match routine of a node-based inference mechanism, thereby combining the strengths of the two inference styles. We discussed the use of equivalence paths to represent the extensional equivalence of intensional concepts. Finally, we urged authors of inheritance hierarchies to explicate their hierarchies by displaying the path-based inference rules that govern inheritance in them.

We also presented a syntax for path-based inference rules which can be summarized as follows:

1. A path is either an arc-relation or a path as described in part 2 enclosed in parentheses. Parentheses may be omitted whenever an ambiguity does not result.
2. If P and Q are paths and x, y, and z are nodes, paths may be formed as follows:
   a. Converse: if P is a path from x to y, $\overleftarrow{P}$ is a path from y to x.
   b. Composition: if P is a path from x to z and Q is a path from z to y, P/Q is a path from x to y.
   c. Composition zero or more times: If P composed with itself zero or more times describes a path from x to y, P^* is a path from x to y.
   d. Composition one or more times: If P composed with itself one or more times is a path from x to y, P^+ is a path from x to y.
   e. Union: If P is a path from x to y or Q is a path from x to y, $P \vee Q$ is a path from x to y.
   f. Intersection: If P is a path from x to y and Q is a path from x to y, $P \& Q$ is a path from x to y.
   g. Complement: If there is no path P from x to y, $\overline{P}$ is a path from x to y.
   h. Irreflexive restriction: If P is a path from x to y and $x \neq y$, P^R is a path from x to y.
   i. Exception: If P is a path from x to y and there is no path Q of length equal to or less than the length of P, $P\backslash Q$ is a path from x to y.
   j. Domain restriction: If P is a

path from x to y and Q is a path from x to z, $(Q\ z)P$ is a path from x to y.

k. Range restriction: If P is a path from x to y and Q is a path from y to z, $P(Q\ z)$ is a path from x to y.

3. A path-based inference rule is of the form *<defined path>* + *<defining path>* where *<defining path>* is any path described by parts 1 or 2 above, and *<defined path>* is either a) a single arc-relation, or b) a composition of n arc relations for some fixed n, i.e. using only "/", not "*" or "+". The rule is interpreted to mean that if the *<defining path>* goes from some node x to some node y then: a) the arc that is the *<defined path>* is inferred to exist from x to y; b) the n arcs that are the *<defined path>* and $n-1$ new intermediate nodes are inferred to exist from x to y.

References

1. Anderson, J. and Hastie, R. Individuation and reference in memory: proper names and definite descriptions. Cognitive Psychology 6, 4 (October, 1974), 495-514.
2. Collins, A.M. and Quillian, R. Retrieval time from semantic memory. J. of Verbal Learning and Verbal Behavior 8, (1969), 240-247.
3. Fikes, R. and Hendrix, G. A network-based knowledge representation and its natural deduction system. Proc. Fifth Int. Jt. Conf. on Artificial Intelligence, Dept. of Computer Science, Carnegie-Mellon University, Pittsburgh, 1977, 235-246.
4. Hays, D.G. Cognitive Structures. unpublished ms. Dept. of Linguistics, SUNY at Buffalo, Amherst, NY.
5. Hays, D.G. Types of processes on cognitive networks. In L.S. Olschki, ed. Mathematical Linguistics, Frienze, Pisa, 1977, 523-532.
6. Hendrix, G.G. Expanding the utility of semantic networks through partitioning. Advance Papers of the Fourth Int. Jt. Conf. on Artificial Intelligence, MIT AI Laboratory, Cambridge, MA, 1975, 115-121.
7. Kay, M. The MIND system. In R. Rustin, ed. Natural Language Processing, Algorithmics Press, New York, 1973, 155-188.
8. McDermott, D. Artificial intelligence meets natural stupidity. SIGART Newsletter, 57 (April, 1976), 4-9.
9. McNabb, S.D. The effects of encoding strategies and age on the memory representation for sentences containing proper names and definite descriptions Report No. 77-3, Indiana Mathematical Psychology Program, Department of Psychology, Indiana University, Bloomington, IN. August, 1977.
10. Quillian, M.R. Semantic memory. In M. Minksy, ed. Semantic Information Processing, MIT Press, Cambridge, MA, 1968, 227-270.
11. Raphael, B. SIR: semantic information retrieval. In M. Minsky, ed. Semantic Information Processing, MIT Press, Cambridge, MA., 1968, 33-145.
12. Schubert, L.K. Extending the expressive power of semantic networks. Artificial Intelligence 7, 2 (Summer, 1976), 163-198.
13. Schwarcz, R.M., Burger, J.F., and Simmons, R.F. A deductive question-answerer for natural language inference. CACM 13, 3 (March, 1970), 167-183.
14. Shapiro, S.C. A net structure for semantic information storage, deduction and retrieval. Proc. Second Int. Jt. Conf. on Artificial Intelligence, The British Computer Society, London, 1971, 512-523.
15. Shapiro, S.C. Representing and locating deduction rules in a semantic network. Proc. Workshop on Pattern-Directed Inference Systems. In SIGART Newsletter, 63 (June, 1977), 14-18.
16. Shapiro, S.C. Techniques of Artificial Intelligence. D. Van Nostrand, New York, 1979.
17. Shapiro, S.C. The SNePS semantic network processing system. In N. Findler, ed. Associative Networks -- The Representation and Use of Knowledge in Computers, Academic Press, New York, in press.
18. Shapiro, S.C. and Woodmansee, G.H. A net structure based relational question answerer: description and examples. Proc. Int. Jt. Conf. on Artificial Intelligence, The MITRE Corp., Bedford, MA., 1969, 325-346.
19. Winograd, T. Frame representations and the declarative/procedural controversy. In D.G. Bobrow and A. Collins, eds. Representation and Understanding, Academic Press. Inc., New York, 1975, 185-210.
20. Woods, W.A. What's in a link: Foundations for semantic networks. In D. G. Bobrow and A. Collins, eds. Representation and Understanding, Academic Press, Inc., New York, 1975, 35-82.

The Representation of Derivable Information in Memory: When What Might Have Been Left Unsaid Is Said

Rand J. Spiro, Joseph Esposito, and Richard J. Vondruska
Center for the Study of Reading
University of Illinois at Urbana-Champaign

It is now widely accepted that natural language comprehension is a constructive process. Information in discourse interacts with a variety of impinging contextual factors (including, most prominently, the comprehender's pre-existing knowledge) in an active, creative process that results in understandings not derivable by any solely linguistic or logical analysis (c.f., Bransford & McCarrell, 1975; Spiro, 1977, in press). Acceptance of the constructive view of comprehension entails a concomitant delimitation of the range of possible theories of mental representation. Knowledge structures must possess some capability for detecting the pragmatic, as well as logical, implications of the incomplete data contained in discourse (c.f., Charniak, 1974; Minsky, 1975; Rumelhart & Ortony, 1977, Schank & Abelson, 1977). In other words, knowledge structures must contain considerable information about the way the world usually works. This characteristic of representation is useful and efficient because natural and social contexts do produce sufficient constraints on worldly events and ideas as to make them, to a limited extent, orderly and predictable.

However, a point often overlooked is that these same knowledge structures, with their information about the world's orderliness, may allow for more efficient processing and memorial representation of explicit information in discourse, in addition to their role in deriving implicit information. This paper will be concerned with the psychological processing of (imperfectly) predictable or derivable information that is nevertheless explicit in discourse.

Predictable Information in Discourse

Despite the fact that most research on inferential processes in comprehension has been concerned with generation of implicit information, much inferentially related information is embodied explicitly in discourse. We are referring here primarily to pragmatic inferences, i.e., implications that are usually but not necessarily true. Language is infrequently characterized by absolute redundancy; semantic content is rarely "repeated," except for special purposes such as emphasis. However, pragmatic inferences are only imperfectly predictable. If you read that a karate champion hit a block, uncertainty is reduced by also reading that the block broke, despite the fact that that outcome is usually to be expected. Similarly, it would not be considered unusual when relating the events at a birthday party to mention that there was a cake with candles blown out by the celebrant. Many things go in stereotyped ways but require explicit mention because the stereotype does not describe all possible cases. Throughout this paper, "predictable" is used as a shorthand for "imperfectly predictable, or characterized by significantly less than perfect uncertainty."

How is explicit but predictable information processed? As was mentioned above, attention has been primarily devoted to the processing of implicit predictable information, leaving little guidance on the present issue. However, in a variety of theoretical orientations, there is a common implication about how predictable information would be dealt with: simply put, explicit information, whether predictable or not, receives sufficient processing to be encoded in long-term memory. For example, Kintsch (1974) assumes "that subjects process and store [an inference] whether or not it is presented explicitly" (p. 154). It is difficult to imagine discourse representation theorists, who argue for the explicit representation in memory of implicit inferences (e.g., Frederiksen, 1975, Meyer, 1974), arguing that explicit inferences are not represented. In schema theories (e.g., Rumelhart & Ortony, 1977), explicit discourse information is used to bind schema variables, again suggesting that predictable information would receive explicit mental representation. If anything, one would expect existing theories to predict that explicit inferences would receive a stronger memorial representation than unpredictable information, given their greater contextual support. For example, in their associative network model, HAM, Anderson and Bower (1973) argued that the greater the number of interconnections between information, the greater the likelihood that information within the interconnected network would be recalled. This view will be referred to as the "storage of explicit inferences" (SEI) hypothesis.

An alternative hypothesis is that predictable information, however central to a discourse, is taken for granted, processed only superficially and receives an attenuated cognitive representation or no enduring representation

at all. If needed subsequently, it can be derived. This view will be referred to as the "superficial processing of explicit inferences" (SPEI) hypothesis. Processing explicit inferences in such a manner has the advantage of a cognitive economy of representation (besides a likely reduction in processing time). Most information that is acquired will never be used again. It would then seem to be more efficient to devote extra processing effort to the occasions when the information is needed (i.e., by deriving it when remembering) rather than exerting effort toward stable encoding at the time of comprehension.

Experiments on the Representation of Explicit Inferences

There are considerable problems in designing an empirical test of the hypothesis that explicit pragmatic inferences in discourse are not represented in long-term memory. If one merely tests memory for the inference, failure to remember could be attributed to not storing the information or to storing and then forgetting it, if the inference is remembered, it could be because it was stored and then retrieved, or it may have been generated at the time of test without having been stored.

Spiro and Esposito (1977) developed a paradigm not subject to the ambiguities of interpretation of the more simple design discussed above. The primary manipulation of interest involved subsequently vitiating the force of an earlier explicit inference. If the inference is not stored, certain predictable errors in recalling it should be made.

In the first experiment, subjects were presented stories which contained information A, B, and C such that B was strongly implied by A except in the presence of C. For example, the A, B, and C elements in one story (about a demonstration by a karate champion) could be paraphrased as follows:

A: The karate champion hit the block.
B: The block broke.
C: He had had a fight with his wife earlier. It was impairing his concentration.

C was either presented prior to A and B (C-Before), after A and B (C-After), or not at all (No-C). When C was not included in the story, if SPEI is correct, the B element should be taken for granted, processed only superficially, and not stably represented. It would be derivable if needed. However, if C is presented after A and B, memory for B should be impaired since B was not stored and C will block its derivation from A at the time of test. On the other hand, if C occurs in the text prior to A and B, then B is not strongly implied by A. B cannot be taken for granted with the assumption that it can be generated later if needed. Here B should be stably represented and memory for B should not be impaired.

However, if SEI is correct, memory for B should not be affected by whether C is before or after A and B, since B is stored whether it is implied by A (C-After) or not implied by A (C-Before). Two objections to this argument can be made. The information might be stored, but remembering C might lead to a decision that the memory for B must be mistaken (a kind of output interference). However, C is present whether it occurs before or after A and B, so such an explanation would not account for differential effects of C-placement. The other possibility is that B is represented in C-After, but the representation is altered or corrected when the C information is encountered. This possibility was investigated in the second experiment.

In the first experiment, the following predictions of the SPEI hypothesis were tested. More errors in response to questions about the presented predictable information (B) should be made in the C-After than in the C-Before conditions. Errors can be erroneous judgments that nothing about the implied information was presented, called B-Mention errors (e.g., the story did not mention whether the block was broken), or, when the subject believes that something about B was mentioned, remembering incorrectly what was specifically said in the direction of conforming with the C information, called B-Incorrect errors (e.g., it said in the story that the block did not break when he hit it). Confidence in errors of the latter kind were also analyzed. If subjects are as confident about these errors as they are about their accurate responses, it would be even more difficult to maintain the hypothesis that the explicit inferences were represented.

In the No-C condition, B-Mention errors may occur since B would not be represented according to the SPEI hypothesis. The more important prediction regarding the No-C condition is that B-Incorrect errors should not occur more often than in the C-Before condition. Otherwise, the differences between C-Before and C-After might be attributable to heightened accuracy due to greater salience of the implied information in the former condition rather than greater inaccuracy due to a failure to store the implied information in the latter condition.

College subjects read eight target vignettes each containing A and B information, and C information included or not and placed as a function of which of the three conditions subjects were randomly assigned to. C information was always on a separate page from the A and B information, and subjects were instructed to not look back after reading a page. After reading all the vignettes, the subjects were tested for their memory for the vignettes. Of particular interest were the two types of questions, mentioned above, concerning the B information (remember, B was always explicit in the stories).

The results supported the hypothesis that pragmatic inferences presented in text are superficially processed and do not receive a

stable and enduring representation in memory. In the C-After condition, subjects tended either to report that the inference was not presented in the text or that the opposite of the inference was presented. Furthermore, confidence in these errors was as high as confidence in correct memories. It is difficult to retain the notion that inferences are deeply processed and stably encoded when the C-After manipulation can produce errors like remembering the block was not broken when the karate champion hit it. The results cannot be attributed to interference produced by the inference-vitiating C information at output, since the C-Before subjects would also be subject to such interference. Neither can the results be attributed to differential availability of C at output, perhaps due to primacy/recency effects related to the position of C in the text, since the information was almost always recalled. Also, unimportance of the B informtion is not a viable alternative since B tended to be central to the story (e.g., in a story about a karate champion's performance, information about his success in the demonstration is certainly important).

One alternative interpretation that remains is that subjects do deeply process and stably encode the presented inference, but "correct" their representation when the inference-vitiating information is presented. If subjects are storing B and then changing or correcting it at the time C is presented, errors on B should occur in the C-After condition no matter how soon the test is administered after reading. However, if the SPEI hypothesis is correct, when delay intervals are brief enough some surface memory for the superficially processed B information may remain, reducing the number of B errors. Accordingly, in the second experiment subjects were tested either immediately after reading each story (Interspersed Questions condition) or, as in the first experiment, after the entire set of stories had been read (Questions-After condition). Again, the C-Before and C-After manipulations were employed.

The results of the second experiment replicated those of the first one in the Questions-After condition. Furthermore, the C-after effect was largely absent in the Interspersed Questions condition, demonstrating that the effect is not due to storing and then changing the representation of the B information (the explicit inference).

Related Issues

The discussion of implications of the superficial processing effect will at times be limited to reading rather than listening. Most of the following is of a speculative nature.

Representation and Underlying Mechanisms

Assuming some compatible representation system, what characterizes the processes that produce the superficial processing effect? At this time, only speculations about alternative possibilities can be offered. There are three potentially beneficial aspects of superficial processing of explicit predictable information: cognitive economy (the information need not be specifically stored in long-term memory), speed of processing (you can process and understand such information rapidly), and automaticity of processing (less conscious effort and working memory space are required).

Two simple, preliminary accounts of the first factor, cognitive economy, can be offered. The superficial processing phenomenon appears most compatible with a schema-theoretic mode of representation. Perhaps variable bindings that are default (or at least high probability) values are not explicitly instantiated when they are explicit in discourse (but see the discussion of Determinants of Performance Variability below). However, one should not be overly persuaded by the simplicity of such an account. Other types of representation systems could also account for the phenomenon. For example, a spreading activation model (e.g., Collins & Loftus, 1975) might predict that explicit information is not tagged in memory when it has been recently activated with some greater than criterion strength. This issue will receive further discussion in the next section.

Regarding speed of processing, several possibilities may be offered: the information is actually predicted, perhaps followed by a selective scanning for partial clues of confirmation (e.g., the word "broke" in the karate champion example; perhaps such checks could be made in the visual periphery and, when positive, result in saccades that skip the predicted information), or the expectation may be formed after beginning to read the predictable information followed by skipping ahead to the next linguistic unit ("Oh. They're talking about this now. Well there's no doubt how it will turn out. I can pass this by."); or temporary binding of a schema variable (essentially a verification of fit) may be more rapid than more durable instantiation, or less metacognitive activity (pondering, studying, rehearsing, etc.) may be devoted to predictable information, given its derivability (this also relates to automaticity, obviously). Regarding automaticity, it seems likely that the amount of conscious processing required would be negatively correlated with the goodness of fit to prior knowledge. Thus conscious attempts to make sense of predictable information would be expected less often. Also, related to the suggestions above regarding expectations and rapidity of processing, the operation of some preattentive process (in the sense of Neisser, 1967) is a possibility. Naturally, it may be the case that all of these factors are contributing. However, some of the factors may be mutually exclusive. For example, if default values are processed automatically, an expectation and confirmation process may be redundant.

Determinants of Performance Variability

Occurrence of superficial processing and failure to store information probably depends on more than predictability or derivability considered in isolation. For one thing, the

derivability of other information in the discourse will have an effect. The greater the proportion of fit to one's schemata for the discourse as a whole, the more likely it is that conforming information will be left to be derived. If a story takes place in a restaurant, and all the restaurant-related information is typical, then that aspect of the story can be stored with the abstract schema node "typical restaurant activities." However, when the proportion of fit is poor, i.e., some atypical events occur, even typical, predictable events may have to be stored.

Occurrence of superficial processing is also likely to be affected by the extent to which the system is taxed. When the system is overloaded, as when there is a large amount of information to be acquired or the time to acquire the information is limited, more superficial processing and leaving of information to be derived probably goes on. Perhaps the system has flexible criteria for derivability, reducing criteria under overload conditions and increasing them when processing load is light (and when demands for recall accuracy are high or when subsequent availability of the information is limited). Briefly digressing, there may be a temptation to confuse superficial processing of derivable information with skimming. However, skimming is a selective seeking and then deep processing of situationally important information (see FRUMP, in Schank & Abelson, 1975) whereas superficial processing involves selectively not processing deeply information perceived as derivable, however important it might be. In other words, the same information that might receive more attention while skimming may receive less attention in normal situations if the information is derivable. This will happen to the extent that skimming results in shallow processing of earlier information that is the basis for the derivability of the later information.

Besides context-based variability in derivability criteria, research in the psychology of prediction indicates the potential operation of a general bias in determining the criterion for derivability and superficial processing. For example, Fischoff (1975, 1977) has found that when people are told that some event has occurred, they increase their subjective probability estimate of the likelihood that the event was going to occur. Similarly, estimation of how much was known before being given a correct answer increases when the answer is provided. In the case of superficial processing of information in discourse, it is possible that the derivability of information is overestimated after it is explicitly encountered. It seems to be a fairly common experience, for example, to not write down an idea that you are sure will be derivable again later, only to find subsequent derivation impossible. What is being suggested here is a source of forgetting not usually discussed in memory theories: superficial processing of information whose derivability has been overestimated.

The Form of Expression of Derivable Information

Semantic content, prior knowledge, and task contexts are not the only determinants of perceived derivability. The linguistic form in which information is expressed will sometimes provide signals of what information is already known or can be taken for granted, as when information is expressed near the beginning of a sentence (c.f., Clark & Haviland, 1977, on the given-new strategy). Taking an example from Morgan and Green (in press), compare sentences (1) and (2).

(1) The government has not yet acknowledged that distilled water causes cancer.
(2) That distilled water causes cancer has not yet been acknowledged by the government.

In (2) there is a stronger implied presumption of the truth of the proposition regarding distilled water and cancer than there is in (1).

In general, it seems that placing information in a sentence-initial subordinate clause lowers the superficial processing criterion. Consider continuations (3) and (4) of "The karate champion hit the block."

(3) The block broke, and then he bowed.
(4) After the block broke, he bowed.

The block's breaking would appear to be more taken for granted in (4) than in (3).

Linguistic signals of predictability or derivability need not be implicit. Consider continuations (5), (6), and (7) of the same sentence as above.

(5) Obviously, the block broke.
(6) As you would expect, the block broke.
(7) Naturally, the block broke.

Words like "clearly" and phrases like "of course" are explicit linguistic signals that information to follow is predictable and can be superficially processed. However, one would expect that such signals could have their effect only for information within an acceptable range of plausibility. That is, a plausible but not predictable continuation may be more likely to be taken (erroneously) as predictable when preceded by a linguistic signal. However, if the information contains salient implausible aspects or something clearly irrelevant, a signalling phrase such as "as you would expect" might result in more attention being devoted to the continuation information.

Implications for the Nature of Discourse Memory

To the extent that discourse is superficially processed, memory must be reconstructive rather than reproductive. Rather than retrieving traces or instantiations of past experience, the past must be inferred or derived. Just as a paleontologist reconstructs a dinosaur from bone fragments, the past must be reconstructed from the incomplete data explicitly stored. Evidence for such reconstructive

processes has been provided by Spiro (1977), who found a pervasive tendency for subjects to produce predictable meaning-changing distortions and importations in text recall under certain conditions. In general, when subsequently encountered information contradicted continuation expectations derived from a target story, the story frequently was reconstructed in such a way as to reconcile or cohere with the continuation information. This process of inferring the past based on the present was termed accommodative reconstruction. After a long retention interval, subjects tended to be more confident that their accommodative recall errors had actually been included in the story than they were confident about the accurate aspects of their recall. Why should such gross errors occur and then be assigned such high confidence? Part of the answer surely involves their function in producing coherence. Still, it is somewhat surprising that subjects should be so sure they read information that bore not even a distant inferential relationship to what they actually did read.

Spiro suggested that the basis for such an effect may be in the way information is treated at the time of comprehension; namely, it is superficially processed and not stored in long-term memory. Then, when remembering, individuals should know (at least tacitly) that considerable amounts of predictable or derivable information they have encountered will not be available in memory. In that case, recall would typically involve deriving a lot of missing information. Accordingly, it would not be surprising that subjects faced with memories that lack coherence would assume that missing reconciling information was presented but only superficially processed at comprehension. The information could then be derived at recall with high confidence. Hence the capacity for restructuring the past based on the present.

Individual Differences

A final caveat should be offered regarding the superficial processing effect, but also applicable to all research on schema-based processes in comprehension and memory. The assumption is usually made that there are no qualitative differences between individuals in the manner in which discourse is processed. However, Spiro and his colleagues have recently found that reliable style differences can be predicted in children (Spiro & Smith, 1978) and in college students (Spiro & Tirre, in preparation). Some individuals appear to be more discourse bound, tending toward over-reliance on bottom-up processes. Others are more prior knowledge bound, tending toward over-reliance on top-down processes. For the adult bottom-up readers, prior knowledge obviously must be used to a certain extent in comprehension. However, where use of prior knowledge is more optional, e.g., in providing a scaffolding for remembering information (Anderson, Spiro, & Anderson, 1978), the bottom-up readers capitalize less. Whether the latter type of individual will evince less knowledge-based superficial processing (again an optional use of prior knowledge) is a question currently under investigation.

References

Anderson, J. R., & Bower, G. H. Human associative memory. New York: Wiley, 1973.

Anderson, R. C., Spiro, R. J., & Anderson, M. C. Schemata as scaffolding for information in text. American Educational Research Journal, 1978, in press.

Bransford, J. D., & McCarrell, N. S. A sketch of a cognitive approach to comprehension. In W. B. Weimer and D. S. Palermo (Eds.), Cognition and the symbolic processes. Hillsdale, N.J.: Erlbaum, 1975.

Charniak, E. Organization and inference in a frame-like system of common sense knowledge. In proceedings of Theoretical issues in natural language processing. Cambridge, Mass. Bolt Beranek & Newman Inc., 1975.

Clark, H. H., & Haviland, S. E. Comprehension and the given-new contract. In R. Freedle (Ed.), Discourse processing. Hillsdale, N.J. Erlbaum, 1978.

Collins, A. M., & Loftus, E. F. A spreading activation theory of semantic processing. Psychological Review, 1975, 82, 407-428.

Fischoff, B. Hindsight ≠ foresight: The effect of outcome knowledge on judgment under uncertainty. Journal of Experimental Psychology: Human Perception and Performance, 1975, 1, 288-299.

Kintsch, W. The representation of meaning in memory. Hillsdale, N.J.: Erlbaum, 1974.

Minsky, M. A framework for representing knowledge. In P. H. Winston (Ed.), The psychology of computer vision. New York: McGraw-Hill, 1975.

Morgan, J. L., & Green, G. M. Pragmatics and reading comprehension. In R. J. Spiro, B. C. Bruce, and W. F. Brewer (Eds.), Theoretical issues in reading comprehension: Perspectives from cognitive psychology, linguistics, artificial intelligence, and education. Hillsdale, N.J.: Erlbaum, in press.

Neisser, U. Cognitive psychology. New York: Appleton-Century-Crofts, 1967.

Rumelhart, D. E., & Ortony, A. The representation of knowledge in memory. In R. C. Anderson, R. J. Spiro, and W. E. Montague (Eds.), Schooling and the acquisition of knowledge. Hillsdale, N.J.: Erlbaum, 1977.

Schank, R. C., & Abelson, R. P. Scripts, plans, goals, and understanding. Hillsdale, N.J.: Erlbaum, 1977.

Spiro, R. J. Remembering information from text: The "State of Schema" approach. In R. C. Anderson, R. J. Spiro, and W. E. Montague (Eds.), Schooling and the acquisition of knowledge. Hillsdale, N.J.: Erlbaum, 1977.

Spiro, R. J. Constructive processes in text comprehension and recall. In R. J. Spiro, B. C. Bruce, and W. F. Brewer (Eds.), Theoretical issues in reading comprehension: Perspectives from cognitive psychology, linguistics, artificial intelligence, and education. Hillsdale, N.J.: Erlbaum, in press.

Spiro, R. J., & Esposito, J. Superficial processing of explicit inferences in text (Tech. Rep. No. 60). Urbana, Ill.: Center for the Study of Reading, University of Illinois, 1977.

Spiro, R. J., & Smith, D. Distinguishing subtypes of poor comprehenders: Patterns of over-reliance on conceptual- vs. data-driven processes (Tech. Rep. No. 61). Urbana, Ill.. Center for the Study of Reading, University of Illinois, 1978.

Frederiksen, C. H. Representing logical and semantic structure of knowledge acquired from discourse. Cognitive Psychology, 1975, 7, 371-458.

Meyer, B. J. F. The organization of prose and its effects on memory. Amsterdam: North Holland, 1975.

Footnote

This research was supported by the National Institute of Education under Contract No. US-NIE-C-400-76-0116.

A Heuristic for Paradigms

Joseph E. Grimes
Cornell University and
Summer Institute of Linguistics

This paper helps clarify one of the pervasive problems of linguistic analysis: the interaction between the paradigmatic and syntagmatic dimensions of language. Paradigms are sets of alternatives: the speaker must decide on one member of the set to use, and the hearer must figure out which he used. In a syntagm or construction, an element chosen out of one paradigm is put together with elements chosen out of others. Thus far all grammars of all languages agree.

The problem comes when we put the grammar together. The choices available in one paradigm turn out often to be limited by those made in some other paradigm that is part of the same construction. Grammar is never as simple as a Cartesian product of paradigms.

Various forms of grammar have various means, none of them quite satisfying, to express these limitations. A common one is footnotes about irregularities; ad hoc features to trigger or block special rules when needed are also used.

Grammar ought to highlight the mutual constraints between paradigms and constructions, not downplay them. Halliday's systemic grammar has done well in this regard (Halliday 1961, Hudson 1971). It is already known to computational linguists through Winograd's work (1972). The heuristic, based on work by Lowe, Dooley, and myself (in press), is expressed within Halliday's framework here, though it is applicable within any other model of language as well.

In systemic terms a paradigm is known as a 'system'. A choice in one system can be the entry condition for another system, one part of a system can have different properties of combination from another part, and two or more systems can be activated together as the basis for a construction. The heuristic is intended to clarify something that is more often guessed at than proved: what element belongs to what system.

What I find, on looking at languages other than English, is that membership in a Hallidayan system is by no means obvious in all cases. This is true for two reasons: first, some elements have properties that permit us to assign them to more than one system, and second, some elements are artifacts of the mapping relation between systems and forms, rather than direct manifestations of choices within systems.

The Data

Table (1) gives some data which illustrate this general point by means of a limited example. It reports cooccurrences among a particularly complex subset of the prefixes to the verb in Huichol, a Uto-Aztecan language spoken in the Mexican Sierra Madre. A 1 in the table means that the prefix at the head of the column has been observed in the combination that the row reports. For this language there are exactly 15 observable combinations of these prefixes, each represented by one row in Table (1). The order in which the rows are written down makes no difference, nor does the order in which the columns appear. ka1- and ka2- are homophonous forms that occupy different positions in the prefix string and have different meanings. & stands for a high back unrounded vowel.

(1)

ka1	ke	p&	m&	ka2	ni
1	0	1	0	1	0
1	0	1	0	0	0
1	0	0	0	1	1
1	0	0	0	0	1
0	1	0	0	0	1
0	1	0	0	0	0
0	0	1	0	1	0
0	0	1	0	0	0
0	0	0	1	1	1
0	0	0	1	1	0
0	0	0	1	0	1
0	0	0	1	0	0
0	0	0	0	1	0
0	0	0	0	0	1
0	0	0	0	0	0

The simple fact that two forms cannot cooccur with each other is the most obvious basis for saying that those two

are members of a single system, that they are in opposition as alternatives in a paradigm, that the choice of one as over against the other has linguistic significance. In Table (1), for example, p&- does not occur in any combination where ni- occurs, and vice versa.

Noncooccurrence patterns

The patterns of noncooccurrence are derived from Table (1) by a simple algorithm:

```
For each column:
     Create a vector of as many 0's as
      there are columns
     For each row that has a 1 in the
      column in question:
          Unite that row with the
           vector.
     Complement the vector.
```

Each of the uncomplemented vectors represents the union of all the combinations which the form at the head of its column enters into. The 1's in its complement therefore identify the elements with which it cannot cooccur.

The Huichol data -- and this is true of other languages, possibly of all languages -- do not allow us to draw immediate conclusions about mutual exclusiveness or simple comembership in systems. The prefixes represented by the complement vectors of each form are

(2) ka1: ke, m&
ke: ka1, p&, m&, ka2
p&: ke, m&, ni
m&: ka1, ke, p&
ka2: ke
ni: p&

A form like p&- can be assigned to one system in opposition with ni-, and to another in opposition with ke- and m&-; but kali-, which could also go into a system with ke and m&-, cooccurs with p&- and therefore cannot represent an alternative to it. The logic of systems in grammar is more complex than independent commutation, with the Cartesian products that that implies, in which each form of one set cooccurs with every form of another.

Decomposition

The true interdependency of a systemic network can be captured in a cooccurrence graph by first decomposing Table (1). The most manageable decomposition strategy found so far is to start with the column that minimizes the number of 1's that would be removed from the table if all the rows that have 1's in that column were removed. We convert those rows into a component subgraph, then continue recursively on the table minus those rows until no rows are left, or until the zero row is left; then we also convert the zero row if there is one into a component subgraph. In the final step of the heuristic, the component subgraphs are united to give the complete cooccurrence graph. That graph of forms is the aim of the heuristic. It is not a systemic network diagram itself, but is rather a statement of a major constraint on the semantic systemic diagram that accounts for the forms.

Component subgraphs are formed by putting alternatives vertically in any order within square brackets, and connecting forms that cooccur in any order by horizontal lines. Absence of any form in a particular combination is represented by ---.

In Table (1) the two rows that contain 1's for ke- have a total of only three 1's in them; so those two rows are taken out for the first subgraph:

(3) ke —— [ni / ---]

This subgraph, like the two rows of Table (1) that it represents, says that ke- can occur with or without ni-

The full set of component subgraphs derived from Table (1) contains only simple alternatives and their Cartesian products:

(4) (a) ke —— [ni / ---]

(b) [ka1 / ---] —— p& —— [ka2 / ---]

(c) ka1 —— [ka2 / ---] —— ni

(d) m& —— ka2 —— [ni / ---]

(e) ka2

(f) m& —— [ni / ---]

(g) ni

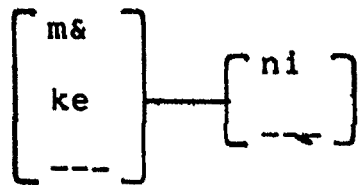
(h) ---

Union of component subgraphs

We unite these subgraphs by conflating what they have in common and symbolizing their differences as alternatives, by the distributive property. Four of the subgraphs, (a), (f), (g), and (h), can be combined without changing the picture of simple systems and

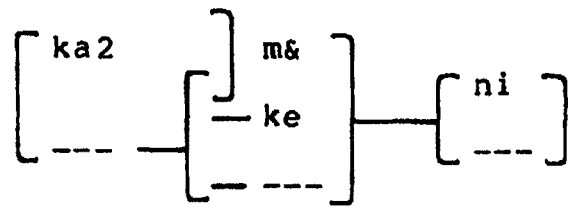
Cartesian products:

(5)



A restriction on Cartesian products appears, however, when we expand the composite diagram further. (d) has three out of four of its elements in common with elements already in the composite diagram (5). The fourth element, however, has nothing to do with ke- or its absence, but only with m&-. Here is where the discrepancies in noncooccurrence properties of different forms come into the picture, and here is where the Hallidayan device of linked brackets is needed in order to show up those discrepancies. The elements in (6) are reordered to disrupt the graphic shape given by (5) as little as possible:

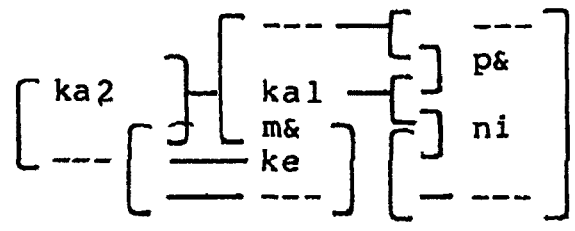
(6)



Cooccurrence graph

The complete cooccurrence graph is built up by continuing in the same way until all the component subgraphs are in it:

(7)



The use of two null symbols in a single set of alternatives does not mean that Huichol has two zero prefixes that contrast with each other, but rather that the graph is essentially nonplanar. Redundant nulls could be eliminated by crossing lines in an equivalent graph.

This diagram now shows all the constraints on cooccurrence that there are for these Huichol prefixes. It is not yet a systemic diagram, because systemic diagrams give differences in meaning and this one gives only cooccurrences of forms. The systemic diagram we come up with will, however, have to account for each of the constraints on cooccurrence given by this diagram.

Our scrutiny of cooccurrences and noncooccurrences has shown us what forms might be in opposition with each other in a semantic system and how those forms interlock. That is as far as our explicit heuristic take us; but it narrows the field for semantic investigation considerably.

Computational aspects

Before I go on to show the payoff in terms of systems of meaningful choices, let me sketch the computational aspects of the heuristic. For a small problem like the one in the example, of course, no computing is needed. But were we to take in all 42 verb prefixes of Huichol, and state how they combine with suffixes and different stem types as well, the heuristic would never get off the ground with pencil and paper. It is a good example of how a computationally simple process, actually a twist on concordance generation, can bring order into an area where a linguist is otherwise all too likely to shrug his shoulders and define oversimplified systems, then write interminable footnotes about why they don't quite combine as he says they do.

A linguist in the field needs a three-step computational aid. Step One is data entry: take in occurring combinations of forms, which could as well be function words or suffixes or any combination of closed class phenomena, and develop a table like Table (1). Step Two is union: read the table and develop a vector for each form that shows the union of all its combinations. Step Three is decomposition: segregate out from the table the subsets of its rows that facilitate making its component subgraphs.

These three steps are easy to implement. The fourth step of the heuristic, forming the cooccurrence graph by uniting the component subgraphs, is at least an order of magnitude more complex, and may not be feasible for a small field computer.

Systemic diagram

After the heuristic procedure is gone through, whether with pencil or by computer, the construction of a semantic hypothesis rich enough to account for all the patterns of cooccurrence can go ahead. This is a standard linguistic undertaking, and has two sides. The first is to investigate the reasons why one or another member of a noncooccurring set like the ones in (2) gets chosen. The reasons for choosing either member of a pair may not be the same in the context of one pattern of choices made in other systems as it is in other contexts. The second part of the semantic inquiry is to identify or combinations of forms whose presence is an artifact of the mapping between meaning and form, and not an assertion of a particular meaning.

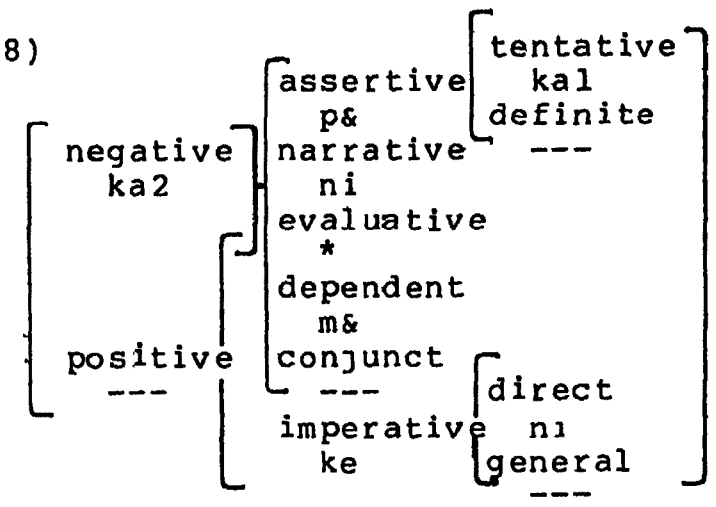
This arbitrariness in the mapping relation shows up in two places in the example. When p&- is present, kal- has either a tentative or a very strong negative meaning: kaalp&mie means 'he might not go' or 'he shall not go!' (the

meaning split is not too different from that of English terribly in terribly disfigured vs. terribly nice). With ni-, however, kal- has to be there when ka2-, the ordinary negative, is present, and may or may not be there when ka2- is absent. The requirement that kal- always go with ka2- in the presence of ni- eliminates the possibility of the two homophones ever being opposed to one another, with resulting confusion between negative and tentativn between negative and tentative meanings.

The other arbitrariness turns up on trying to relate m&- with ni-. m&- by itself is the sign of a dependent verb, and ni- by itself of an independent verb at a partice combination m&ni, however, has nothing to do with either of these meanings; it makes a statement of the speaker's opinion. I take it to be a morphologically complex expression of a separate term of the modal system.

Taking these discrepancies into account gives us a systemic diagram:

(8)



[narrative] implies kal
obligatory with [negative]
optional with [positive]
[evaluative] realized as m&+ni

It is by straightening out kal- and m&ni- that it becomes possible for us to give a systemic diagram plus a set of realization rules for it. The straightforward realization rules are written right into the diagram: for example, if you choose [negative], utter ka2- The more complex realizations are given at the bottom of the diagram.

The terms of the systemic diagram are labels for semantic choices that have been explained elsewhere and do not concern us now; they do not constitute explanations in themselves. Once the arbitrary mappings are defined in realization rules, the diagram embodies only one real restriction on Cartesian products of paradigms, in that Huichol has no special negative imperative form. (It uses the negative declarative p&ka2 in its place.) The completeness of the analysis is supported by the fact that the interconnected paradigms of (8) have exactly 14 paths through them, and that together with the optional rule for the realization of kal- with ni-, these yield exactly the 15 rows of Table (1) with which we began.

Bibliography

Grimes, Joseph E., Ivan Lowe, and Robert A. Dooley. in press. Closed systems with complex restrictions. Anthropological Linguistics.

Halliday, Michael A. K. 1961. Categories of the theory of grammar. Word 17:241-292.

Hudson, R. A. 1971. English complex sentences. Amsterdam: North-Holland Publishing Company.

Winograd, Terry. 1972. Understanding natural language. New York: Academic Press.

A Computational Account of Some Constraints on Language

Mitchell Marcus
MIT Artificial Intelligence Laboratory

In a series of papers over the last several years, Noam Chomsky has argued for several specific properties of language which he claims are universal to all human languages [Chomsky 73, 75, 76]. These properties, which form one of the cornerstones of his current linguistic theory, are embodied in a set of constraints on language, a set of restrictions on the operation of rules of grammar.

This paper will outline two arguments presented at length in [Marcus 77] demonstrating that important subcases of two of these constraints, the Subjacency Principle and the Specified Subject Constraint, fall out naturally from the structure of a grammar interpreter called PARSIFAL, whose structure is in turn based upon the hypothesis that a natural language parser needn't simulate a nondeterministic machine. This "Determinism Hypothesis" claims that natural language can be parsed by a computationally simple mechanism that uses neither backtracking nor pseudo-parallelism, and in which all grammatical structure created by the parser is "indelible" in that it must all be output as part of the structural analysis of the parser's input. Once built, no grammatical structure can be discarded or altered in the course of the parsing process.

In particular, this paper will show that the structure of the grammar interpreter constrains its operation in such a way that, by and large, grammar rules cannot parse sentences which violate either the Specified Subject Constraint or the Subjacency Principle. The component of the grammar interpreter upon which this result principally depends is motivated by the Determinism Hypothesis; this result thus provides indirect evidence for the hypothesis. This result also depends upon the use within a computational framework of the closely related notions of *annotated surface structure* and *trace theory*, which also derive from Chomsky's recent work.

(It should be noted that these constraints are far from universally accepted. They are currently the source of much controversy; for various critiques of Chomsky's position see [Postal 74; Bresnan 76]. However, what is presented below does not argue for these constraints, *per se*, but rather provides a different sort of explanation, based on a processing model, of why the sorts of sentences which these constraints forbid are bad. While the exact formulation of these constraints is controversial, the fact that some set of constraints is needed to account for this range of data is generally agreed upon by most generative grammarians. The account which I will present below *is* crucially linked to Chomsky's, however, in that trace theory is at the heart of this account.)

Because of space limitations, this paper deals only with those grammatical processes characterized by the competence rule "MOVE NP"; the constraints imposed by the grammar interpreter upon those processes characterized by the rule "MOVE WH-phrase" are discussed at length in [Marcus 77] where I show that the behavior characterized by Ross's Complex NP Constraint [Ross 67] itself follows directly from the structure of the grammar interpreter for rather different reasons than the behavior considered in this section. Also because of space limitations, I will not attempt to show that the two constraints I will deal with here *necessarily* follow from the grammar interpreter, but rather only that they *naturally* follow from the interpreter, in particular from a simple, natural formulation of a rule for passivization, which itself depends heavily upon the structure of the interpreter. Again, necessity is argued for in detail in [Marcus 77].

This paper will first outline the structure of the grammar interpreter, then present the PASSIVE rule, and then finally show how Chomsky's constraints "fall out" of the formulation of PASSIVE.

Before proceeding with the body of this paper, two other important properties of the parser should be mentioned which will not be discussed here. Both are discussed at length in [Marcus 77]; the first is sketched as well in [Marcus 78].

1) Simple rules of grammar can by written for this interpreter which elegantly capture the significant generalizations behind not only passivization, but also such constructions as *yes/no* questions, imperatives, and sentences with existential *there*. These rules are reminiscent of the sorts of rules proposed within the framework of the theory of generative grammar, despite the fact that the rules presented here must recover underlying structure given only the terminal string of the surface form of the sentence.

2)The grammar interpreter provides a simple explanation for the difficulty caused by "garden path" sentences, such as "The cotton clothing is made of grows in Mississippi." Rules can be written for this interpreter to

resolve local structural ambiguities which might seem to require nondeterministic parsing; the power of such rules, however, depends upon a parameter of the mechanism. Most structural ambiguities can be resolved, given an appropriate setting of this parameter, but those which typically cause garden paths cannot.

The Structure of PARSIFAL

PARSIFAL maintains two major data structures: a pushdown stack of incomplete constituents called *the active node stack*, and a small three-place *constituent buffer* which contains constituents which are complete, but whose higher level grammatical function is as yet uncertain.

Figure 1 below shows a snapshot of the parser's data structures taken while parsing the sentence "John should have scheduled the meeting.". Note that the active node stack in shown growing *downward*, so that the structure of the stack reflects the structure of the emerging parse tree. At the bottom of the stack is an auxiliary node labelled with the features *modal*, *past*, etc., which has as a daughter the modal "should". Above the bottom of the stack is an S node with an NP as a daughter, dominating the word "John". There are two words in the buffer, the verb "have" in the first buffer cell and the word "scheduled" in the second. The two words "the meeting" have not yet come to the attention of the parser. (The structures of form "(PARSE-AUX CPOOL)" and the like will be explained below.)

The Active Node Stack

S1 (S DECL MAJOR S) / (PARSE-AUX CPOOL)
NP : (John)
AUX1 (MODAL PAST VSPL AUX) / (BUILD-AUX)
MODAL : (should)

The Buffer

1 : WORD3 (*HAVE VERB TNSLESS AUXVERB PRES V+3S) : (have)
2 : WORD4 (*SCHEDULE COMP-OBJ VERB INF-OBJ V-3S ED=EN EN PART PAST ED) : (scheduled)

Yet unseen words: the meeting .

Figure 1 - PARSIFAL's two major data structures.

The constituent buffer is the heart of the grammar interpreter; it is the central feature that distinguishes this parser from all others. The words that make up the parser's input first come to its attention when they appear at the end of this buffer after morphological analysis. Triggered by the words at the beginning of the buffer, the parser may decide to create a new grammatical constituent, create a new node at the bottom of the active node stack, and then begin to attach the constituents in the buffer to it. After this new constituent is completed, the parser will then pop the new constituent from the active node stack; if the grammatical role of this larger structure is as yet undetermined, the parser will insert it into the first cell of the buffer. The parser is free to examine the constituents in the buffer, to act upon them, and to otherwise use the buffer as a workspace.

While the buffer allows the parser to examine some of the context surrounding a given constituent, it does not allow arbitrary look-ahead. The length of the buffer is strictly limited; in the version of the parser presented here, the buffer has only three cells. (The buffer must be extended to five cells to allow the parser to build NPs in a manner which is transparent to the "clause level" grammar rules which will be presented in this paper. This extended parser still has a window of only three cells, but the effective start of the buffer can be changed through an "attention shifting mechanism" whenever the parser is building an NP. In effect, this extended parser has two "logical" buffers of length three, one for NPs and another for clauses, with these two buffers implemented by allowing an overlap in one larger buffer. For details, see [Marcus 77].)

Note that each of the three cells in the buffer can hold a *grammatical constituent* of any type, where a constituent is any tree that the parser has constructed under a single root node. The size of the structure underneath the node is immaterial; both "that" and "that the big green cookie monster's toe got stubbed" are perfectly good constituents once the parser has constructed a subordinate clause from the latter phrase.

The constituent buffer and the active-node stack are acted upon by a grammar which is made up of pattern/action rules; this grammar can be viewed as an augmented form of Newell and Simon's production systems [Newell & Simon 72]. Each rule is made up of a pattern, which is matched against some subset of the constituents of the buffer and the accessible nodes in the active node stack (about which more will be said below), and an action, a sequence of operations which acts on these constituents. Each rule is assigned a numerical *priority*, which the grammar interpreter uses to arbitrate simultaneous matches.

The grammar as a whole is structured into *rule packets*, clumps of grammar rules which can be activated and deactivated as a group; the grammar interpreter only attempts to match rules in packets that have been activated by the grammar. Any grammar rule can activate a packet by associating that packet with the constituent at the bottom of the active node stack. As long as that node is at the bottom of the stack, the packets associated with it are active; when that node is pushed into the stack, the packets remain associated with it, but become active again only when that node reaches the bottom of the stack. For example, in figure 1 above, the packet BUILD-AUX is associated with the bottom of the stack, and is thus active, while the packet PARSE-AUX is associated with the S node above the auxiliary.

The grammar rules themselves are written in a language called PIDGIN, an English-like formal language that is translated into LISP by a simple grammar translator based on the notion of top-down operator precedence [Pratt 73]. This use of pseudo-English is similar to the use of pseudo-English in the grammar for Sager's STRING parser [Sager 73]. Figure 2 below gives a schematic overview of the organization of the grammar, and exhibits some of the rules that make up the packet PARSE-AUX.

A few comments on the grammar notation itself are

in order. The general form of each grammar rule is:

{Rule <name> priority: <priority> in <packet>
<pattern> --> <action>}

Each pattern is of the form :

[<description of 1st buffer constituent>] [<2nd>] [<3rd>]

The symbol "=", used only in pattern descriptions, is to be read as "has the feature(s)". Features of the form "*<word>" mean "has the root <word>", e.g. "*have" means "has the root "have"". The tokens "1st", "2nd", "3rd" and "C" (or "c") refer to the constituents in the 1st, 2nd, and 3rd buffer positions and the current active node (i.e. the bottom of the stack), respectively. The PIDGIN code of the rule patterns should otherwise be fairly self-explanatory.

Priority	Pattern Description of: 1st	2nd	3rd	The Stack	Action
			PACKET1		
6:	[ ]	[ ]	[ ]		--> ACTION1
10:	[ ]			[ ]	--> ACTION2
10:	[ ]	[ ]	[ ]	[ ]	--> ACTION3
			PACKET2		
10:	[ ]	[ ]			--> ACTION4
15:	[ ]			[ ]	--> ACTION5

(a) - The structure of the grammar.

{RULE START-AUX PRIORITY: 10. IN PARSE-AUX
[=verb] -->
Create a new aux node.
Label C with the meet of the features of 1st and pres, past, future, tnsless.
Activate build-aux.}

{RULE TO-INFINITIVE PRIORITY: 10. IN PARSE-AUX
[=*to, auxverb] [=tnsless] -->
Label a new aux node inf.
Attach 1st to C as to.
Activate build-aux.}

(b) - Some grammar rules that initiate auxiliaries.

Figure 2

The parser (i.e. the grammar interpreter interpreting some grammar) operates by attaching constituents which are in the buffer to the constituent at the bottom of the stack: functionally, a constituent is in the stack when the parser is attempting to find its daughters, and in the buffer when the parser is attempting to find its mother. Once a constituent in the buffer has been attached, the grammar interpreter will automatically remove it from the buffer, filling in the gap by shifting to the left the constituents formerly to its right. When the parser has completed the constituent at the bottom of the stack, it pops that constituent from the active node stack; the constituent either remains attached to its parent, if it was attached to some larger constituent when it was created, or else it falls into the first cell of the constituent buffer, shifting the buffer to the right to create a gap (and causing an error if the buffer was already full). If the constituents in the buffer provide sufficient evidence that a constituent of a given type should be initiated, a new node of that type can be created and pushed onto the stack; this new node can also be attached to the node at the bottom of the stack before the stack is pushed, if the grammatical function of the new constituent is clear when it is created.

This structure is motivated by several properties which, as is argued in [Marcus 77], any "non-nondeterministic" grammar interpreter must embody. These principles, and their embodiment in PARSIFAL, are as follows:

1) *A deterministic parser must be at least partially data driven.* A grammar for PARSIFAL is made up of pattern/action rules which are triggered when constituents which fulfill specific descriptions appear in the buffer.

2) *A deterministic parser must be able to reflect expectations that follow from the partial structures built up during the parsing process.* Packets of rules can be activated and deactivated by grammar rules to reflect the properties of the constituents in the active node stack.

3) *A deterministic parser must have some sort of constrained look-ahead facility.* PARSIFAL's buffer provides this constrained look-ahead. Because the buffer can hold several constituents, a grammar rule can examine the context that follows the first constituent in the buffer before deciding what grammatical role it fills in a higher level structure. The key idea is that the size of the buffer can be sharply constrained if each location in the buffer can hold a single complete constituent, regardless of that constituent's size. *It must be stressed that this look-ahead ability must be constrained in some manner, as it is here by limiting the length of the buffer; otherwise the "determinism" claim is vacuous.*

The General Grammatical Framework - Traces

The form of the structures that the current grammar builds is based on the notion of *Annotated Surface Structure*. This term has been used in two different senses by Winograd [Winograd 71] and Chomsky [Chomsky 73]; the usage of the term here can be thought of as a synthesis of the two concepts. Following Winograd, this term will be used to refer to a notion of surface structure *annotated by the addition of a set of features* to each node in a parse tree. Following Chomsky, the term will be used to refer to a notion of surface structure annotated by the addition of an element called *trace* to indicate the "underlying position" of "shifted" NPs.

In current linguistic theory, a trace is essentially a "phonologically null" NP in the surface structure representation of a sentence that has no daughters but is "bound" to the NP that filled that position at some level of underlying structure. In a sense, a trace can be viewed as a "dummy" NP that serves as a placeholder for the NP that earlier filled that position; in the same sense, the trace's

binding can be viewed as simply a pointer to that NP. It should be stressed at the outset, however, that a trace is indistinguishable from a normal NP in terms of normal grammatical processes; a trace *is* an NP, even though it is an NP that dominates no lexical material.

There are several reasons for choosing a properly annotated surface structure as a primary output representation for syntactic analysis. While a deeper analysis is needed to recover the predicate/argument structure of a sentence (either in terms of Fillmore case relations [Fillmore 68] or Gruber/Jackendoff "thematic relations" [Gruber 65; Jackendoff 72]), phenomena such as focus, theme, pronominal reference, scope of quantification, and the like can be recovered only from the surface structure of a sentence. By means of proper annotation, it is possible to encode in the surface structure the "deep" syntactic information necessary to recover underlying predicate/argument relations, and thus to encode in the same formalism both deep syntactic relations and the surface order needed for pronominal reference and the other phenomena listed above.

Some examples of the use of trace are given in Figure 3 immediately below.

(1a) What did John give to Sue?
(1b) What did John give *t* to Sue?
|_____|
(1c) John gave *what* to Sue.

(2a) The meeting was scheduled for Wednesday.
(2b) The meeting was scheduled *t* for Wednesday.
|_____|
(2c) ∇ scheduled a *meeting* for Wednesday.

(3a) John was believed to be happy.
(3b) John was believed [$_S$ *t* to be happy].
|_____|

Figure 3 – Some examples of the use of trace.

One use of trace is to indicate the underlying position of the wh-head of a question or relative clause. Thus, the structure built by the parser for 3.1a would include the trace shown in 3.1b, with the trace's binding shown by the line under the sentence. The position of the trace indicates that 3.1a has an underlying structure analogous to the overt surface structure of 3.1c.

Another use of trace is to indicate the underlying position of the surface subject of a passivized clause. For example, 3.2a will be parsed into a structure that includes a trace as shown as 3.2b; this trace indicates that the subject of the passive has the underlying position shown in 3.2c. The symbol "∇" signifies the fact that the subject position of (2c) is filled by an NP that dominates no lexical structure. (Following Chomsky, I assume that a passive sentence in fact has *no underlying subject*, that an agentive "by NP" prepositional phrase originates as such in underlying structure.) The trace in (3b) indicates that the phrase "to be happy", which the brackets show is really an embedded clause, has an underlying subject which is identical with the surface subject of the matrix S, the clause that dominates the embedded complement. Note that what is conceptually the underlying subject of the embedded clause has been passivized into subject position of the matrix S, a phenomenon commonly called "raising". The analysis of this phenomenon assumed here derives from [Chomsky 73]; it is an alternative to the classic analysis which involves "raising" the subject of the embedded clause into object position of the matrix S before passivization (for details of this later analysis see [Postal 74])

The Passive Rule

In this section and the next, I will briefly sketch a solution to the phenomena of passivization and "raising" in the context of a grammar for PARSIFAL. This section will present the Passive rule; the next section will show how this rule, without alteration, handles the "raising" cases.

Let us begin with the parser in the state shown in figure 4 below, in the midst of parsing 3.2a above. The analysis process for the sentence prior to this point is essentially parallel to the analysis of any simple declarative with one exception: the rule PASSIVE-AUX in packet BUILD-AUX has decoded the passive morphology in the auxiliary and given the auxiliary the feature *passive* (although this feature is not visible in figure 4). At the point we begin our example, the packet SUBJ-VERB is active.

The Active Node Stack (1. deep)
S21 (S DECL MAJOR) / (SS-FINAL)
NP : (The meeting)
AUX : (was)
VP : ↓
C: VP17 (VP) / (SUBJ-VERB,
VERB : (scheduled)

The Buffer
1 PP14 (PP) : (for Wednesday)
2 WORD162 (*. FINALPUNC PUNC) : (.)

Figure 4 - Partial analysis of a passive sentence: after the verb has been attached.

The packet SUBJ-VERB contains, among other rules, the rule PASSIVE, shown in figure 5 below. The pattern of this rule is fulfilled if the auxiliary of the S node dominating the current active node (which will always be a VP node if packet SUBJ-VERB is active) has the feature *passive*, and the S node has not yet been labelled *np-preposed*. (The notation "** C" indicates that this rule matches against the two accessible nodes in the stack, not against the contents of the buffer.) The action of the rule PASSIVE simply creates a trace, sets the binding of the trace to the subject of the dominating S node, and then drops the new trace into the buffer.

```
{RULE PASSIVE IN SUBJ-VERB
[** c; the aux of the s above c is passive;
        the s above c is not np-preposed] -->
Label the s above c np-preposed.
Create a new np node labelled trace.
Set the binding of c to the np of the s above c.
Drop c.}
```

Figure 5 - Six lines of code captures np-preposing.

The state of the parser after this rule has been executed, with the parser previously in the state in figure 4 above, is shown in figure 6 below. S21 is now labelled with the feature *np-preposed*, and there is a trace, NP53, in the first buffer position. NP53, as a trace, has no daughters, but is bound to the subject of S21.

```
The Active Node Stack ( 1. deep)
  S21 (NP-PREPOSED S DECL MAJOR) / (SS-FINAL)
        NP : (The meeting)
        AUX : (was)
        VP : ↓
C:    VP17 (VP) / (SUBJ-VERB)
        VERB : (scheduled)

      The Buffer
1 :   NP53 (NP TRACE) : bound to: (The meeting)
2 :   PP14 (PP) : (for Wednesday)
3 :   WORD162 (*. FINALPUNC PUNC) : (.)
```

Figure 6 - After PASSIVE has been executed.

Now rules will run which will activate the two packets SS-VP and INF-COMP, given that the verb of VP17 is "schedule". These two packets contain rules for parsing simple objects of non-embedded Ss, and infinitive complements, respectively. Two such rules, each of which utilize an NP immediately following a verb, are given in figure 7 below. The rule OBJECTS, in packet SS-VP, picks up an NP after the verb and attaches it to the VP node as a simple object. The rule INF-S-START1, in packet INF-COMP, triggers when an NP is followed by "to" and a tenseless verb; it initiates an infinitive complement and attaches the NP as its subject. (An example of such a sentence is "We wanted John to give a seminar next week".) The rule INF-S-START1 must have a higher priority than OBJECTS because the pattern of OBJECTS is fulfilled by any situation that fulfills the pattern of INF-S-START1; if both rules are in active packets and match, the higher priority of INF-S-START1 will cause it to be run instead of OBJECTS.

```
{RULE OBJECTS PRIORITY: 10 IN SS-VP
[=np] -->
Attach 1st to c as np.}

{RULE INF-S-START1 PRIORITY: 5. IN INF-COMP
[=np] [=*to,auxverb] [=tnsless] -->
Label a new s node sec, inf-s.
Attach 1st to c as np.
Activate parse-aux.}
```

Figure 7 - Two rules which utilize an NP following a verb.

While there is not space to continue the example here in detail, note that the rule OBJECTS will trigger with the parser in the state shown in figure 6 above, and will attach NP53 as the object of the verb "schedule. OBJECTS is thus totally indifferent both to the fact that NP53 was not a regular NP, but rather a trace, and the fact that NP53 did not originate in the input string, but was placed into the buffer by grammatical processes. Whether or not this rule is executed, is absolutely unaffected by differences between an active sentence and its passive form; the analysis process for either is identical as of this point in the parsing process. Thus, the analysis process will be exactly parallel in both cases after the PASSIVE rule has been executed. (I remind the reader that the analysis of passive assumed above, following Chomsky, does *not* assume a process of "agent deletion", "subject postposing" or the like.)

Passives in Embedded Complements - "Raising"

The reader may have wondered why PASSIVE drops the trace it creates into the buffer rather than immediately attaching the new trace to the VP node. As we will see below, such a formulation of PASSIVE also correctly analyzes passives like 3.3a above which involve "raising", but with no additional complexity added to the grammar, correctly capturing an important generalization about English. To show the range of the generalization, the example which we will investigate in this section, sentence (1) in figure 8 below, is yet a level more complex than 3.3a above; its analysis is shown schematically in 8.2. In this example there are two traces: the first, the subject of the embedded clause, is bound to the subject of the major clause, the second, the object of the embedded S, is bound to the first trace, and is thus ultimately bound to the subject of the higher S as well. Thus the underlying position of the NP "the meeting" can be viewed as being the object position of the embedded S, as shown in 8.3.

(1)The meeting was believed to have been scheduled for Wednesday.
(2)The meeting was believed [$_S$ *t* to have been scheduled. *t* for Wednesday]
(3) ∇ believed [$_S$ ∇ to have scheduled *the meeting* for Wednesday].

Figure 8 - This example shows simple passive and raising.

We begin our example, once again, right after "believed" has been attached to VP20, the current active node, as shown in figure 9 below. Note that the AUX node has been labelled *passive*, although this feature is not shown here.

The Active Node Stack (1. deep)
S22 (S DECL MAJOR) / (SS-FINAL)
NP : (The meeting)
AUX : (was)
VP : ↓
C: VP20 (VP) / (SUBJ-VERB)
VERB : (believed)

The Buffer
1 : WORD166 (*TO PREP AUXVERB) : (to)
2 : WORD167 (*HAVE VERB TNSLESS AUXVERB PRES ...) : (have)

Figure 9 - After the verb has been attached.

The packet SUBJ-VERB is now active; the PASSIVE rule, contained in this packet now matches and is executed. This rule, as stated above, creates a trace, binds it to the subject of the current clause, and drops the trace into the first cell in the buffer. The resulting state is shown in figure 10 below.

The Active Node Stack (1. deep)
S22 (NP-PREPOSED S DECL MAJOR) / (SS-FINAL)
NP : (The meeting)
AUX : (was)
VP : ↓
C: VP20 (VP) / (SUBJ-VERB)
VERB : (believed)

The Buffer
1 : NP55 (NP TRACE) : bound to: (The meeting)
2 : WORD166 (*TO PREP AUXVERB) : (to)
3 : WORD167 (*HAVE VERB TNSLESS AUXVERB PRES ...) : (have)

Yet unseen words: been scheduled for Wednesday

Figure 10 - After PASSIVE has been executed.

Again, rules will now be executed which will activate the packet SS-VP (which contains the rule OBJECTS) and, since "believe" takes infinitive complements, the packet INF-COMP (which contains INF-S-START1), among others. (These rules will also deactivate the packet SUBJ-VERB.) Now the patterns of OBJECTS and INF-S-START1 will both match, and INF-S-START1, shown above in figure 7, will be executed by the interpreter since it has the higher priority. (Note once again that a trace is a perfectly normal NP from the point view of the pattern matching process.) This rule now creates a new S node labelled infinitive and attaches the trace NP55 to the new infinitive as its subject. The resulting state is shown in figure 11 below.

The Active Node Stack (2. deep)
S22 (NP-PREPOSED S DECL MAJOR) / (SS-FINAL)
NP : (The meeting)
AUX : (was)
VP : ↓
VP20 (VP) / (SS-VP THAT-COMP INF-COMP)
VERB : (believed)
C: S23 (SEC INF-S S) / (PARSE-AUX)
NP : bound to: (The meeting)

The Buffer
1 : WORD166 (*TO PREP AUXVERB) : (to)
2 : WORD167 (*HAVE VERB TNSLESS AUXVERB PRES ...) : (have)

Yet unseen words: been scheduled for Wednesday

Figure 11 - After INF-S-START1 has been executed.

We are now well on our way to the desired analysis. An embedded infinitive has been initiated, and a trace bound to the subject of the dominating S has been attached as its subject, although no rule has explicitly "lowered" the trace from one clause into the other.

The parser will now proceed exactly as in the previous example. It will build the auxiliary, attach it, and attach the verb "scheduled" to a new VP node. Once again PASSIVE will match and be executed, creating a trace, binding it to the subject of the clause (in this case itself a trace), and dropping the new trace into the buffer. Again the rule OBJECTS will attach the trace NP57 as the object of VP21, and the parse will then be completed by grammatical processes which will not be discussed here. An edited form of the tree structure which results is shown in figure 12 below. A trace is indicated in this tree by giving the terminal string of its ultimate binding in parentheses.

(NP-PREPOSED S DECL MAJOR)
NP: (MODIBLE NP DEF DET NP)
The meeting
AUX: (PASSIVE PAST V13S AUX)
was
VP: (VP)
VERB: believed
NP: (NP COMP)
S: (NP-PREPOSED SEC INF-S S)
NP: (NP TRACE) (bound* to: The meeting)
AUX: (PASSIVE PERF INF AUX)
to have been
VP: (VP)
VERB: scheduled
NP: (NP TRACE) (bound* to: The meeting)
PP: (PP)
PREP: for
NP: (NP TIME DOW)
Wednesday

Figure 12 - The final tree structure.

This example demonstrates that the simple formulation of the PASSIVE rule presented above, interacting with other simply formulated grammatical rules

for parsing objects and initiating embedded infinitives, allows a trace to be attached either as the object of a verb or as the subject of an embedded infinitive, whichever is the appropriate analysis for a given grammatical situation. Because the PASSIVE rule is formulated in such a way that it drops the trace it creates into the buffer, later rules, already formulated to trigger on an NP in the buffer, will analyze sentences with NP-preposing exactly the same as those without a preposed subject. Thus, we see that the availability of the buffer mechanism is crucial to capturing this generalization; such a generalization can only be stated by a parser with a mechanism much like the buffer used here.

The Grammar Interpreter and Chomsky's Constraints

Before turning now to a sketch of a computational account of Chomsky's constraints, there are several important limitations of this work which must be enumerated.

First of all, while two of Chomsky's constraints seem to fall out of the grammar interpreter, there seems to be no apparent account of a third, the Propositional Island Constraint, in terms of this mechanism.

Second, Chomsky's formulation of these constraints is intended to apply to all rules of grammar, both syntactic rules (i.e. transformations) and those rules of semantic interpretation which Chomsky calls "rules of construal", a set of shallow semantic rules which govern anaphoric processes [Chomsky 77]. The discussion here will only touch on purely syntactic phenomena; the question of how rules of semantic interpretation can be meshed with the framework presented in this document has yet to be investigated.

Third, the arguments presented below deal only with English, and in fact depend strongly upon several facts about English syntax, most crucially upon the fact that English is subject-initial. Whether these arguments can be successfully extended to other language types is an open question, and to this extent this work must be considered exploratory.

And finally, I will not show that these constraints must be true *without exception*; as we will see, there are various situations in which the constraints imposed by the grammar interpreter can be circumvented. Most of these situations, though, will be shown to demand much more complex grammar formulations than those typically needed in the grammar so far constructed. This is quite in keeping with the suggestion made by Chomsky [Chomsky 77] that the constraints are not necessarily without exception, but rather that exceptions will be "highly marked" and therefore will count heavily against any grammar that includes them.

The Specified Subject Constraint

The Specified Subject Constraint (SSC), stated informally, says that no rule may involve two constituents that are Dominated by different cyclic nodes unless the lower of the two is the subject of an S or NP. Thus, no rule may involve constituents X and Y in the structure shown in figure 13 below, if α and β are cyclic nodes and Z is the subject of α, Z distinct from X.

$[_{\beta}...Y...[_{\alpha}\ Z...X...]...Y...]$

Figure 13 - SSC:
No rule can involve X and Y in this structure.

The SSC explains why the surface subject position of verbs like "seems" and "is certain" which have no underlying subject can be filled only by the subject and not the object of the embedded S: The rule "MOVE NP" is free to shift any NP into the empty subject position, but is constrained by the SSC so that the object of the embedded S cannot be moved out of that clause. This explains why (a) in figure 14 below, but not 14b, can be derived from 14c; the derivation of 14b from 14c would violate the SSC.

(a) John seems to like Mary.
(b)*Mary seems John to like.
(c) ∇ seems [$_S$ John to like Mary]

Figure 14 - Some examples illustrating the SSC.

essence, then, the Specified Subject Constraint constrains the rule "MOVE NP" in such a way that only the subject of a clause can be moved out of that clause into a position in a higher S. Thus, if a trace in an annotated surface structure is bound to an NP Dominated by a higher S, that trace must fill the subject position of the lower clause.

In the remainder of this section I will show that the grammar interpreter constrains grammatical processes in such a way that annotated surface structures constructed by the grammar interpreter will have this same property, given the formulation of the PASSIVE rule presented above. In terms of the parsing process, this means that if a trace is "lowered" from one clause to another as a result of a "MOVE NP"-type operation during the parsing process, then it will be attached as the subject of the second clause. To be more precise, if a trace is attached so that it is Dominated by some S node S1, and the trace is bound to an NP Dominated by some other S node S2, then that trace will necessarily be attached so that it fills the subject position of S1. This is depicted in figure 15 below.

The Active Node Stack

```
                .....
          S2 ... / ...
                ...
                NP2
                ...
C:        S1 ... / ...
                NP: NP1 (NP TRACE) : bound to NP2
```

Figure 15 - NP1 must be attached as the subject of S1 since it is bound to an NP Dominated by a higher S.

Looking back at the complex passive example involving "raising" presented above, we see that the parsing process results in a structure exactly like that shown above. The original point of the example, of course, was that the rather simple PASSIVE rule handles this case without the need for some mechanism to explicitly lower the NP. The PASSIVE rule captures this generalization by

dropping the trace it creates into the buffer (after appropriately binding the trace), thus allowing other rules written to handle normal NPs (e.g. OBJECTS and INF-S-START1) to correctly place the trace.

This statement of PASSIVE does more, however, than simply capture a generalization about a specific construction. As I will argue in detail below, the behavior specified by both the Specified Subject Constraint and Subjacency follows almost immediately from this formulation. In [Marcus 77], I argue that this formulation of PASSIVE is the only simple, non-*ad hoc*, formulation of this rule possible, and that all other rules characterized by the competence rule "MOVE NP" must operate similarly; here, however, I will only show that these constraints follow naturally from this formulation of PASSIVE, leaving the question of necessity aside. I will also assume one additional constraint below, the *Left-to-Right Constraint*, which will be briefly motivated later in this paper as a natural condition on the formulation of a grammar for this mechanism.

> The Left-to-Right Constraint: the constituents in the buffer are (almost always) attached to higher level constituents in left-to-right order, i.e. the first constituent in the buffer is (almost always) attached before the second constituent.

I will now show that a trace created by PASSIVE which is bound to an NP in one clause can only serve as the subject of a clause dominated by that first clause.

Given the formulation of PASSIVE, a trace can be "lowered" into one clause from another only by the indirect route of dropping it into the buffer before the subordinate clause node is created, which is exactly how the PASSIVE rule operates. This means that the ordering of the operations is crucially: 1) create a trace and drop it into the buffer, 2) create a subordinate S node, 3) attach the trace to the newly created S node. The key point is that at the time that the subordinate clause node is created and becomes the current active node, the trace must be sitting in the buffer, filling one of the three buffer positions. Thus, the parser will be in the state shown in figure 16 below, with the trace, in fact, most likely in the first buffer position.

The Active Node Stack

.....

C: S123 (S SEC ...) / ...

The Buffer

...

NP123 (NP TRACE) : bound to NP in S above S123

...

Figure 16 - Parser state after embedded S created.

Now, given the L-to-R Constraint, a trace which is in the buffer at the time that an embedded S node is first created must be one of the first several constituents attached to the S node or its daughter nodes. From the structure of English, we know that the leftmost three constituents of an embedded S node, ignoring topicalized constituents, must either be

COMP NP AUX
or
NP AUX [$_{VP}$ VERB ...].

(The COMP node will dominate flags like "that" or "for" that mark the beginning of a complement clause.) But then, if a trace, itself an NP, is one of the first several constituents attached to an embedded clause, the only position it can fill will be the subject of the clause, exactly the empirical consequence of Chomsky's Specified Subject Constraint in such cases as explained above.

The L-to-R Constraint

Let us now return to the motivation for the L-to-R Constraint. Again, I will not attempt to prove that this constraint must be true, but merely to show why it is plausible.

Empirically, the Left-to-Right Constraint seems to hold for the most part; for the grammar of English discussed in this paper, and, it would seem, for any grammar of English that attempts to capture the same range of generalizations as this grammar, the constituents in the buffer are utilized in left-to-right order, with a small range of exceptions. This usage is clearly not enforced by the grammar interpreter as presently implemented; it is quite possible to write a set of grammar rules that specifically ignores a constituent in the buffer until some arbitrary point in the clause, though such a set of rules would be highly *ad hoc*. However, there rarely seems to be a need to remove other than the first constituent in the buffer.

The one exception to the L-to-R Constraint seems to be that a constituent C_i may be attached before the constituent to its left, C_{i-1}, if C_i does not appear in surface structure in its underlying position (or, if one prefers, in its unmarked position) and if its removal from the buffer reestablishes the unmarked order of the remaining constituents, as in the case of the AUX-INVERSION rule discussed earlier in this paper. To capture this notion, the L-to-R Constraint can be restated as follows: All constituents must be attached to higher level constituents according to the left-to-right order of constituents in the unmarked case of that constituent's structure.

This reformulation is interesting in that it would be a natural consequence of the operation of the grammar interpreter if packets were associated with the phrase structure rules of an explicit "base component", and these rules were used as templates to build up the structure assigned by the grammar interpreter. A packet of grammar rules would then be explicitly associated with each symbol on the right hand side of each phrase structure rule. A constituent of a given type would then be constructed by activating the packets associated with each node type of the appropriate phrase structure rule in left-to-right order. Since these base rules would reflect the unmarked l-to-r order of constituents, the constraint suggested here would then simply fall out of the interpreter mechanism.

Subjacency

Before turning to the Subjacency Principle, a few auxiliary technical terms need to be defined: If we can

trace a path up the tree from a given node X to a given node Y, then we say *X is dominated by Y*, or equivalently, *Y dominates X*. If Y dominates X, and no other nodes intervene (i.e. X is a daughter of Y), then *Y immediately (or directly) dominates X*. [Akmajian & Heny 75]. One non-standard definition will prove useful: I will say that if Y dominates X, and Y is a cyclic node, i.e. an S or NP node, and there is no other cyclic node Z such that Y dominates Z and Z dominates X (i.e. there is no intervening cyclic node Z between Y and X) then *Y Dominates X*.

The principle of Subjacency, informally stated, says that no rule can involve constituents that are separated by more than one cyclic node. Let us say that a node X *is subjacent to* a node Y if there is at most one cyclic node, i.e. at most one NP or S node, between the cyclic node that Dominates Y and the node X. Given this definition, the Subjacency principle says that no rule can involve constituents that are not subjacent.

The Subjacency principle implies that movement rules are constrained so that they can move a constituent only into positions that the constituent was subjacent to, i.e. only within the clause (or NP) in which it originates, or into the clause (or NP) that Dominates that clause (...). This means that if α, β, and ϵ in figure 17 are cyclic nodes, no rule can move a constituent from position X to either of the positions Y, where $[_{\alpha}...X...]$ is distinct from $[_{\alpha}X]$.

$[_{\epsilon}...Y...[_{\beta}...[_{\alpha}...X...]...]...Y...]$

Figure 17 - Subjacency:
No rule can involve X and Y in this structure.

Subjacency implies that if a constituent is to be "lifted" up more than one level in constituent structure, this operation must be done by repeated operations. Thus, to use one of Chomsky's examples, the sentence given in figure 18a, with a deep structure analogous to 18b, must be derived as follows (assuming that "is certain", like "seems", has no subject in underlying structure): The deep structure must first undergo a movement operation that results in a structure analogous to 18c, and then another movement operation that results in 18d, each of these movements leaving a trace as shown. That 18c is in fact an intermediate structure is supported by the existence of sentences such as 18e, which purportedly result when the ∇ in the matrix S is replaced by the lexical item "it", and the embedded S is tensed rather than infinitival. The structure given in 18f is ruled out as a possible annotated surface structure, because the single trace could only be left if the NP "John" was moved in one fell swoop from its underlying position to its position in surface structure, which would violate Subjacency.

(a) John seems to be certain to win.
(b) ∇ seems $[_S$ ∇ to be certain $[_S$ John to win]]
(c) ∇ seems $[_S$ John to be certain $[_S$ *t* to win]]
(d) John seems $[_S$ *t* to be certain $[_S$ *t* to win]]
(e) It seems that John is certain to win.
(f) John seems $[_S$ ∇ to be certain $[_S$ *t* to win]]

Figure 18 - An example demonstrating Subjacency.

Having stated Subjacency in terms of the abstract competence theory of generative grammar, I now will show that a parsing correlate of Subjacency follows from the structure of the grammar interpreter. Specifically, I will show that there are only limited cases in which a trace generated by a "MOVE-NP" process can be "lowered" more than one clause, i.e. that a trace created and bound while any given S is current must almost always be attached either to that S or to an S which is Dominated by that S.

Let us begin by examining what it would mean to lower a trace more than one clause. Given that a trace can only be "lowered" by dropping it into the buffer and then creating a subordinate S node, as discussed above, lowering a trace more than one clause necessarily implies the following sequence of events, depicted in figure 19 below: First, a trace NP1 must (a) be created with some S node, S1, as the current S, (b) bound to some NP Dominated by that S and then (c) dropped into the buffer. By definition, it will be inserted into the first cell in the buffer. (This is shown in figure 19a) Then a second S, S2, must be created, supplanting S1 as the current S, and then yet a third S, S3, must be created, becoming the current S. During all these steps, the trace NP1 remains sitting in the buffer. Finally NP1 is attached under S3 (fig. 19b). By the Specified Subject Constraint, NP1 must then attach to S3 as its subject.

The Active Node Stack
.....
C: S1 ... / ...

The Buffer
1st: NP1 (NP TRACE) : bound to NP Dominated by S1
...

(a) NP1 is dropped into the buffer while S1 is the current S.

The Active Node Stack
.....
S1 ... / ...
S2 ... / ...
C: S3 ... / ...
NP1 (NP TRACE) : bound to NP Dominated by S

(b) - After S2 and S3 are created, NP1 is attached to S3 as its subject (by the SSC).

Figure 19 -- Lowering a trace more than 1 clause

But this sequence of events is highly unlikely. The essence of the argument is this:

Nothing in the buffer can change between the time that S2 is created and S3 is created if NP1 remains in the buffer. NP1, like any other node that is dropped from the active node stack into the buffer, is inserted into the first buffer position. But then, by the L-to-R Constraint, nothing to the right of NP1 can be attached to a higher level constituent until NP1 is attached. (One can show that it is most unlikely that any constituents will enter to the left of NP1 after it is dropped into the buffer, but I will suppress this detail here; the full argument is included in [Marcus 77].)

But if the contents of the buffer do not change between the creation of S2 and S3, then what can possibly motivate the creation of both S2 and S3? The contents of the buffer must necessarily provide clear evidence that both of these clauses are present, since, by the Determinism Hypothesis, the parser must be correct if it initiates a constituent. Thus, the same three constituents in the buffer must provide convincing evidence not only for the creation of S2 but also for S3. Furthermore, if NP1 is to become the subject of S3, and if S2 Dominates S3, then it would seem that the constituents that follow NP1 in the buffer must also be constituents of S3, since S3 must be completed before it is dropped from the active node stack and constituents can then be attached to S2. But then S2 must be created *entirely* on the basis of evidence provided by the constituents of another clause (unless S3 has less than three constituents). Thus, it would seem that the contents of the buffer cannot provide evidence for the presence of both clauses unless the presence of S3, by itself, is enough to provide confirming evidence for the presence of S2. This would be the case only if there were, say, a clausal construction that could only appear (perhaps in a particular environment) as the initial constituent of a higher clause. In this case, if there are such constructions, a violation of Subjacency should be possible.

With the one exception just mentioned, there is no motivation for creating two clauses in such a situation, and thus the initiation of only one such clause can be motivated. But if only one clause is initiated before NP1 is attached, then NP1 must be attached to this clause, and this clause is necessarily subjacent to the clause which Dominates the NP to which it is bound. Thus, the grammar interpreter will behave as if it enforces the Subjacency Constraint.

As a concluding point, it is worthy of note that while the grammar interpreter appears to behave exactly as if it were constrained by the Subjacency principle, it is in fact constrained by a version of the Clausemate Constraint! (The Clausemate Constraint, long tacitly assumed by linguists but first explicitly stated, I believe, by Postal [Postal 64], states that a transformation can only involve constituents that are Dominated by the same cyclic node. This constraint is at the heart of Postal's attack on the constraints that are discussed above and his argument for a "raising" analysis.) The grammar interpreter, as was stated above, limits grammar rules from examining any node in the active node stack higher than the current cyclic node, which is to say that it can only examine clausemates. The trick is that a trace is created and bound while it is a "clausemate" of the NP to which it is bound in that the current cyclic node at that time is the node to which that NP is attached. The trace is then dropped into the buffer and another S node is created, thereby destroying the clausemate relationship. The trace is then attached to this new S node. Thus, in a sense, the trace *is* lowered from one clause to another. The crucial point is that while this lowering goes on as a result of the operation of the grammar interpreter, it is only implicitly lowered in that 1) the trace was never *attached* to the higher S and 2) it is *not* dropped into the buffer because of any realization that it must be "lowered"; *in fact it may end up attached as a clausemate* of the NP to which it is bound - as the passive examples presented earlier make clear. The trace is simply dropped into the buffer because its grammatical function is not clear, and the creation of the second S follows from other independently motivated grammatical processes. From the point of view of this processing theory, we can have our cake and eat it too; to the extent that it makes sense to map results from the realm of processing into the realm of competence, in a sense *both* the clausemate/"raising" and the Subjacency positions are correct.

Evidence for the Determinism Hypothesis

In closing, I would like to show that the properties of the grammar interpreter crucial to capturing the behavior of Chomsky's constraints were originally motivated by the Determinism Hypothesis, and thus, to some extent, the Determinism Hypothesis explains Chomsky's constraints.

The strongest form of such an argument, of course, would be to show that (a) either (i) the grammar interpreter accounts for *all* of Chomsky's constraints in a manner which is conclusively universal or (ii) the constraints that it will not account for are wrong and that (b) the properties of the grammar interpreter which were crucial for this proof were *forced* by the Determinism Hypothesis. If such an argument could be made, it would show that the Determinism Hypothesis provides a natural processing account of the linguistic data characterized by Chomsky's constraints, giving strong confirmation to the Determinism Hypothesis.

I have shown none of the above, and thus my claims must be proportionately more modest. I have argued only that important sub-cases of Chomsky's constraints follow from the grammar interpreter, and while I can show that the Determinism Hypothesis strongly *motivates* the mechanisms from which these arguments follow, I cannot show necessity. The extent to which this argument provides evidence for the Determinism Hypothesis must thus be left to the reader; no objective measure exists for such matters.

The ability to drop a trace into the buffer is at the heart of the arguments presented here for Subjacency and the SSC as consequences of the functioning of the grammar interpreter; this is the central operation upon which the above arguments are based. But the buffer itself, and the fact that a constituent can be dropped into the buffer if its grammatical function is uncertain, are directly motivated by the Determinism Hypothesis. Given this, it is fair to claim that if Chomsky's constraints follow from the operation of the grammar interpreter, then they are strongly linked to the Determinism Hypothesis. If Chomsky's constraints are in fact true, then the arguments presented in this paper provide solid evidence in support of the Determinism Hypothesis.

Acknowledgments

This paper summarizes one result presented in my Ph.D. thesis; I would like to express my gratitude to the many people who contributed to the technical content of that work: Jon Allen, my thesis advisor, to whom I owe a special debt of thanks, Ira Goldstein, Seymour Papert, Bill

Martin, Bob Moore, Chuck Rieger, Mike Genesereth, Gerry Sussman, Mike Brady, Craig Thiersch, Beth Levin, Candy Bullwinkle, Kurt VanLehn, Dave McDonald, and Chuck Rich.

This paper describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defence under Office of Naval Research Contract N00014-75-C-0643.

BIBLIOGRAPHY

Akmajian, A. and F. Heny [1975] *An Introduction to the Principles of Transformational Syntax*, MIT Press, Cambridge, Mass.

Bresnan, J. W. [1976] "Evidence for a Theory of Unbounded Transformations", *Linguistic Analysis* 2:353.

Chomsky, N. [1973] "Conditions on Transformations", in S. Anderson and P. Kiparsky, eds., *A Festschrift for Morris Halle*, Holt, Rinehart and Winston, N.Y.

Chomsky, N. [1975] *Reflections on Language*, Pantheon, N.Y.

Chomsky, N. [1976] "Conditions on Rules of Grammar", *Linguistic Analysis* 2:303.

Chomsky, N. [1977] "On Wh-Movement", in A. Akmajian, P. Culicover, and T. Wasow, eds., *Formal Syntax*, Academic Press, N.Y.

Fillmore, C. J. [1968] "The Case for Case" in *Universals in Linguistic Theory*, E. Bach and R. T. Harms, eds., Holt, Rinehart, and Winston, N.Y.

Gruber, J. S. [1965] *Studies in Lexical Relations*, unpublished Ph.D. thesis, MIT.

Jackendoff, R. S. [1972] *Semantic Interpretation in Generative Grammar*, MIT Press, Cambridge, Mass.

Marcus, M. P. [1977] *A Theory of Syntactic Recognition for Natural Language*, unpublished Ph.D. thesis, MIT.

Marcus, M. P. [1978] "Capturing Linguistic Generalizations in a Parser for English", in the proceedings of *The 2nd National Conference of the Canadian Society for Computational Studies of Intelligence*, Toronto, Canada.

Newell, A. and H.A. Simon [1972] *Human Problem Solving*, Prentice-Hall, Englewood Cliffs, N.J.

Postal, P. M. [1974] *On Raising*, MIT Press, Cambridge, Mass.

Pratt, V. R. [1973] "Top-Down Operator Precedence", in the proceedings of *The SIGACT/SIGPLAN Symposium on Principles of Programming Languages*, Boston, Mass.

Sager, N. [1973] "The String Parser for Scientific Literature", in [Rustin 73].

Winograd, T. [1971] *Procedures as a Representation for Data in a Computer Program for Understanding Natural Language*, Project MAC-TR 84, MIT, Cambridge, Mass.

Woods, W. A. [1970] "Transition Network Grammars for Natural Language Analysis", *Communications of the ACM* 13:591.

REMARKS ON PROCESSING, CONSTRAINTS, AND THE LEXICON*

Thomas Wasow
Stanford University

Linguists have long recognized the desirability of embedding a theory of grammar within a theory of linguistic performance (see, eg., Chomsky (1965;10-15)). It has been widely assumed by transformationalists that an adequate model of a language user would include as one component some sort of generative grammar. Yet transformational grammarians have devoted relatively little energy to the problem that Bresnan (in press) calls "the grammatical realization problem": "How *would* a reasonable model of language use incorporate a transformational grammar?" When this question has been raised, little support could be adduced for the hypothesis that the operations of transformational grammar play a part in speakers' or hearers' processing of sentences (see Fodor, *et al* (1974; chapter 5)). Instead of concerning themselves with questions of processing, transformationalists have concentrated their efforts (at least in the last decade or so) on the problem of constraining the power of their theory. The goal of much recent research has been to construct as restrictive a theory of grammar as possible, within the bounds set by the known diversity of human languages (see, e.g., Ross (1967), Chomsky (1973), Bresnan (1976), Emonds (1976), and Culicover and Wexler (1977) for examples of this type of research).

Computational linguists, on the other hand, have not explicitly concerned themselves very much with the problem of constraints (but see Woods (1973; 124-5) for an exception). Rather, their goal has been to find effective procedures for the parsing and processing of natural language. While this is implicitly a restriction to recursive languages, the computational literature has dealt more with questions of processing than with how to limit the class of available grammars or languages.

In previous papers (Osherson and Wasow (1976), Wasow (in press a, 1978)) I have argued for the legitimacy of the quest for constraints as a research strategy. I have argued that a theory that places limits on the class of possible languages makes significant empirical claims about human mental capacities, and can contribute to a solution to "the fundamental empirical problem of linguistics" (as Chomsky has called it) of how children are able to learn languages with such facility. I have tried to show that such psychological claims can be made, without making any assumptions about what role grammars play in performance. In short, I have argued that a theory of grammar can make significant contributions to psychology, independent of the answer to the grammatical realization problem.

Recent work by Joan Bresnan (in press) takes a very different position: she has suggested that transformationalists ought to pay more attention to the grammatical realization problem, and that considerations of processing suggest radical modifications in the theory of transformational grammar. Further, she argues that there is ample grammatical evidence for these modifications. In this paper I will suggest some extensions of her proposals, and will explore some of their empirical consequences. Further, I will argue that her framework makes it possible to impose rather restrictive constraints on grammatical theory. Thus, I will argue that the grammatical realization problem and the problem of constraining transformational theory, while logically independent, are both addressed by Bresnan's proposals. If I am correct in this, then Bresnan's "realistic transformational grammar" represents a major convergence of the concerns of transformational and computational linguists.

My presentation will consist of three parts. First, I will briefly sketch Bresnan's framework. Second, I will suggest some extensions of her proposals and point out some consequences of these extensions. Third, I will propose how her framework can be constrained, and indicate certain desirable consequences of my proposals.

The primary innovation of Bresnan's framework is that it eliminates a large class of transformations in favor of an enriched conception of the lexicon. The grammar that results is one that Bresnan claims is far more realistic from a processing point of view than other versions of transformational grammar. She points out striking similarities between her proposals and recent computational and psycholinguistic work by Kaplan and Wanner, and she argues that Augmented Transition Networks can provide at least a partial answer to the grammatical realization problem within her framework.

I now sketch very roughly what Bresnan's "realistic" transformational grammar is like. Rules like passive, dative, and raising rules, which are "structure-preserving" (in the sense that their outputs are structurally identical to independently required base-generated structures) and "local" (in the sense that the elements affected are always in the immediate environment of some governing lexical item, usually a verb), are eliminated from the transformational component and relegated to the lexicon. Lexical entries include, among other things, (strict) subcategorization frames and more abstract representations which Bresnan calls "functional structures" or "predicate argument structures". Subcategorization frames give the syntactic environments in which the lexical item may appear; these are expressed in terms of a basic set of grammatical relations, including "subject" and "object". These notions, while universal, are instantiated differently in different languages; for example, Bresnan takes essentially the structural definitions of "subject" and "object" proposed by Chomsky (1965; 71) as language-specific characterizations of these notions for English. Functional structures give a more abstract representation of the elements mentioned in the subcategorization frame, indicating what their "logical" relationships are. Thus, the

functional structure corresponds very roughly to the deep structure in the standard theory of transformational grammar; and the subcategorization frame corresponds even more roughly to the surface structure.

What the standard theory did with local structure-preserving transformations Bresnan can do in either of two ways. Relationships like active/passive are handled by positing two separate lexical entries for active and passive verb forms. The productivity of this relationship can be accounted for by means of a lexical redundancy rule, which would say, in effect, that corresponding to the typical transitive verb there is an intransitive verb which looks morphologically like the perfect form of the transitive, and whose subject plays the same logical role (i.e., in the functional structure) as the object of the transitive verb. Bresnan's other way of replacing local structure-preserving rules is illustrated most clearly with the raising rules. Raising to object position, for example, is used to capture the fact that the NP which is syntactically the object of one clause is logically not an argument of that clause at all, but a subject of the subordinate clause. Bresnan expresses this simply in terms of the relationship between the subcategorization frame and the functional structure; that is, the object of the main clause plays no role in the functional structure of that clause, but is "passed down" to play a role in the next clause down. In the interests of brevity I will not illustrate Bresnan's framework here. Rather, I will refer the interested reader to her paper, and go on to indicate my reasons for seeking to modify her proposals.

My primary motivation comes from some earlier work of mine (Wasow (1977)), which argued against the elimination of local, structure-preserving transformations. My argument was based on the observation that there are two similar but distinct classes of linguistic relationships whose differences can be expressed rather naturally as the differences between transformational rules and lexical redundancy rules. The clearest example of this is the English passive. It has often been suggested that some passive participles are adjectives and others verbs; I pointed out that adjectival passives and verbal passives differed in certain systematic ways. My central claim was that the surface subject of adjectival passives was always the deep direct object of the corresponding verb. For example, a passive participle which is demonstrably adjectival (e.g., because it is prefixed with *un-* or immediately follows *seem*) may not have as its surface subject the "logical" subject of a lower clause, the indirect object, or a chunk of an idiom: **John is unknown to be a communist*; **John seemed told the story*; **Advantage seemed taken of John*. A verbal passive, in contrast, could have as its subject any NP which could immediately follow the corresponding active verb: *John is known to be a communist*; *John was told the story*; *Advantage was taken of John*. This, I claimed, would follow from the hypothesis that adjectival passives are formed by a lexical redundancy rule, whereas verbal passives are transformationally derived, if lexical redundancy rules are "relational", in the sense that they are formulated in terms of grammatical relations such as subject and object, whereas transformations are "structural", i.e., they are operations on phrase structure tree.

It is evident that my earlier position is inconsistent with Bresnan's recent proposals. My extensions of her ideas, developed in collaboration with Ron Kaplan, are in part an attempt to capture within her framework the distinction my earlier paper sought to explicate in terms of the lexicon/transformation contrast. They are also motivated by the very interesting comments of Anderson (1977). Anderson suggests that I was mistaken in claiming that the operative factor in formulating rules like the adjectival passive rule was the deep grammatical relation of the surface subject. Rather, he argues, it is thematic relations like "theme", "agent", "goal" and "source" (see Gruber (1965) and Jackendoff (1972)) which are crucial[1]. Assuming Anderson to be correct, an obvious modification of Bresnan's system suggests itself, which would permit the distinctions of my earlier paper to be captured. Let us suppose that the functional structure in lexical entries is a specification of which thematic relations should be assigned to the elements mentioned in the subcategorization frame. Then we may distinguish two types of lexical rules: those that make reference to thematic relations and those that do not. The former would correspond to rules that my earlier paper called lexical, and the latter to those that I called transformations. This is the extension of Bresnan's framework that I wish to propose. I will illustrate by formulating the two passive rules and the dative rule and applying them to a fragment of the lexicon of English.

My formalism is based on the assumption that the grammatical relations are given language-wide definitions in structural terms (at least in English) along the lines indicated by Bresnan, and that a verb's subcategorization frame merely indicates which relations it has, and what grammatical categories those relations are assigned to. (Thus, I differ from Bresnan in this respect, for she assumed that grammatical relations would be limited to NP's). I will adopt the following abbreviations: "SS" = (surface) subject; SO = (surface) object; "SO2" = (surface) second object; "1" = theme; "2" = agent; "3" = goal; "4" = complement. The rule forming verbal passive participles from the corresponding active lexical entries can now be formulated[2] quite simply as SS←SO. This is to be interpreted as follows: eliminate "SS" wherever it appears in the entry for the active verb (eliminating also any assignment it may have to a thematic relation) and change all occurrences of "SO" to "SS"[3]. The adjectival passive rule will differ from this in that it has an additional condition on it: if SO=1, then SS←SO. This condition insures that the SO is "local", in the sense that it bears a thematic relation to the verb. The dative rule[4] also has a "localness" condition: if SO2=1, then SO←SO2. Let me illustrate these rules with a simple example, namely the verb *sell*. The basic lexical entry I posit for this verb includes the following information: SS=NP, SO=NP, SO2=NP; SS=2, SO=3, SO2=1. This, I claim, is among the information that must be included in a representation of *sell* in such uses as *They sold John two cars*. Applying the verbal passive rule to this entry, we get the following: SS=NP, SO2=NP; SS=3, SO2=1. This verb appears in examples like *John was sold two cars*. Since the original entry for *sell* did not meet the condition SO=1, the adjectival passive rule is not applicable; correspondingly, forms like **John was unsold two cars* are impossible. The condition for application of dative, SO2=1 is met, so we can derive an entry in which SS=NP, SO=NP; SS=2, SO=1. This corresponds to examples like *They sold two cars*. Notice that this last entry does satisfy the condition on the adjectival passive rule, so we can derive the following entry for an adjectival passive participle for *sell*: SS=NP; SS=1. This corresponds to examples like *Two cars were unsold*.

Let us now turn to some more complex examples. Specifically, I now want to look at several different verbs which share the same strict subcategorization frame, namely, SS=NP, SO=NP, SO2=VP. The verbs in question differ from one another along two dimensions, namely, the assignment of thematic relations, and control properties. What I mean by this latter phrase is quite simple: the understood subject of the VP in the SO2 position will be the SS in some cases and the SO in others. I will represent this in the functional structure by assigning a thematic relation not simply to SO2, but to SO2(SS) or SO2(SO), depending on the control properties[5]. My assignments of thematic relations are intended to reflect certain intuitions about the semantic roles of the various elements, but I cannot, in general, provide empirical arguments

for my assignments, other than the fact that they give me the right results. I do have an operational criterion for deciding whether to call the SO a 1 or a 3: when the verb in question could appear in a double object construction (i.e., immediately followed by two NP's). I called the SO a 3; otherwise, I called it a 1. Thus, in what follows, the assignments are correlated with the fact that *promise* and *tell* have double object forms (*I promised/told him nothing*), but *persuade* and *believe* do not (**I persuaded/believed him nothing*).

Consider first *persuade*. The functional structure for this verb in examples like *They persuaded John to leave* would be SS=2, SO=1, SO2(SO)=4. The passive rule yields an entry whose functional structure is SS=1, SO2(SS)=4. Since SO=1 in the original entry, this passive may be either verbal or adjectival. Hence, we can get both *John was persuaded to leave* and *John seemed persuaded to leave*. On the other hand, the condition for application of dative is not met, and, accordingly, we cannot get **They persuaded to leave*. Transformational studies going back to Rosenbaum (1967) have pointed out numerous differences between the behavior of *persuade* and that of *believe*. The standard analysis of these differences has involved the claim that the surface object of *believe* was raised from the subject position of the complement. The system proposed here can mimic that analysis by assigning to *believe* a functional structure in which the SO bears no thematic relation[6]: SS=2 , SO2(SO)=1. These are the assignments for examples like *I believe John to be at home*. The verbal passive rule will apply, yielding the functional structure SO2(SS)=1, for examples like *John is believed to be at home*. Since neither the condition on the adjectival passive rule nor that on the dative rule is met, we can predict the non-occurrence of examples like **John seems believed to be at home* and **I believe to be at home*. The next verb I wish to consider is *tell*, which standard transformational accounts would not distinguish in any relevant way from *persuade*. For reasons noted above, I assign *tell* the functional structure SS=2, SO=3, SO2(SO)=1, as in examples like *We told John to bring the beer*. Applying the verbal passive rule we get SS=3, SO2(SS)=1, covering examples like *John was told to bring the beer*. The condition on the adjectival passive rule is not satisfied, so we cannot derive **John seemed told to bring the beer*. Notice now that the condition for applying the dative rule is met Applying the rule results in the following functional structure: SS=2, SO()=1; this structure is ill-formed, since there is no controller. Accordingly, examples like **We told to bring the beer* are impossible. Finally, consider *promise* in examples like *I promised John to mow the lawn*. *Promise* is exactly like *tell*, except that the controller is the subject, not the object, i.e., the functional structure is SS=2, SO=3, SO2(SS)=1. If we try to apply either passive rule, we will get the following functional structure: SS=3, SO2()=1. This is ill-formed for the same reason that the dative of *tell* was, namely, lack of a controller. The corresponding examples are also impossible: **John was promised to mow the lawn* or **John seemed promised to mow the lawn*. Dative, however, can apply, yielding an entry whose functional structure is SS=2, SO(SS)=1. This corresponds to examples like *I promised to mow the lawn*.

I hope that this fragment of the lexicon suffices to show that my proposed modification of Bresnan's system permits an elegant and natural account of a number of syntactic distinctions, including some which have not been discussed in the literature, to my knowledge. One nice feature that I would like to emphasize is that my proposals provide a rather straightforward account of Visser's (1973; 2118) observation: "A passive transform is only possible when the complement relates to the immediately preceding (pro)noun." In my terminology, passive will be impossible when the active has a complement controlled by the SS, as in the case of *promise*; for passivization will always lead to an uncontrolled complement. Thus, to take another standard example of Visser's generalization, we can account for the distinction between *strike* and *regard* much as we accounted for the difference between *promise* and *tell*. Both will have the following subcategorization frame: SS=NP, SO=NP, SO2=AP. Their functional structures will include the assignments SS=2 and SO=1; they will differ in that *regard* will have SO2(SO)=4, while *strike* has SO2(SS)=4. These assignments are for examples like *John regards/strikes Mary as pompous*. If we apply passive to *regard* we get SS=1, SO2(SS)=4, as in *Mary is regarded as pompous*. Applying passive to *strike* we get SS=1, SO2()=4, which is ill-formed, as is **Mary is struck as pompous*. Notice, incidentally, that this example illustrates that, in the system I advocate here, constituents other than VP's can serve as predicates and be subject to control.

This concludes my suggestions for modifying Bresnan's framework. I hope I have succeeded in indicating how a grammar which makes extensive use of the lexicon in place of syntactic transformations can handle an array of syntactic facts in a satisfying manner. Next, I wish to argue that a system of the sort outlined here can be effectively constrained in reasonable and interesting ways. Intuitively, it seems quite plausible that such a system would be easy to constrain, for by drastically reducing the role of transformations, it opens the way for reductions in the power of transformations. A number of candidate constraints on transformations come to mind. For example, within Bresnan's framework one might plausibly argue that no transformation can create new grammatical relations (e.g., there will be no "subject-creating" transformations, like passive or raising to subject), or that no transformation can change the words in the sentence morphologically (e.g., there will be no nominalization, agreement, or case-marking transformations--cf. Brame (1978)). Various ways in which lexical rules might be constrained also come to mind: most immediately, it seems to me that many of the "laws" of relational grammar proposed by Postal and Perlmutter in recent years could be translated straightforwardly into the kind of framework discussed here. In this paper, however, I would like to consider the consequences of a constraint on transformations modeled on the Freezing Principle of Culicover and Wexler (1977). My proposal depends on distinguishing two classes of transformations: root transformations (Emonds (1976)), and what I will call unbounded rules. Root transformations are rules like English subject-auxiliary inversion in questions, which apply only to main clauses; unbounded rules are transformations (e.g., *wh*-movement) which involve a crucial variable, i.e., they move something over a variable or they delete something under identity with something on the other side of a variable[7] (see the contributions by Chomsky, Bach, Bresnan, and Partee in Culicover, *et al* (1977) for discussion of whether unbounded rules are truly unbounded). The constraint I wish to propose, which I will call **the interaction constraint** is the following: once a rule of one of these classes has applied to a given structure, no further rule of the same type may apply to that structure. More specifically, when a transformation applies, the smallest constituent containing all of the affected elements becomes **frozen**, in the sense that no further transformations of the same type may analyze it. This means, in effect, that there will be no interactions among root transformations, nor among unbounded transformations (though a root transformation may interact with an unbounded rule, as in the case of English *wh*-questions). I believe that there are several desirable consequences of prohibiting such interactions.

First of all, let me mention a somewhat conjectural reason for advocating the interaction constraint. As noted above, a very similar proposal emerged from the learnability studies of Wexler, Culicover, and Hamburger; they were able to prove

that a class of grammars in which nodes were frozen under similar conditions was learnable by a fairly simple learning device. Hence, it seems plausible to conjecture that the interaction constraint might be useful in devising a learnability proof for some version of Bresnan's theory. In any event, it seems that the interaction constraint would make the language-learner's task easier by limiting the extent to which surface structures could deviate from base forms (see Coker & Crain (in preparation)).

Second, there is empirical support for the interaction constraint. Emonds (1972: 38-40) shows that only one root preposing transformation can apply per sentence. Since the smallest structure containing initial position in a root sentence is the whole sentence, Emonds's observation is an immediate consequence of the interaction constraint. Similarly, many of the ways in which unbounded transformations are prohibited from interacting are familiar. For example, the fact that elements in relative clauses are inaccessible to unbounded transformations has been extensively discussed in the literature (e.g., Ross (1967), Chomsky (1973), to cite only two accounts). This fact follows from the interaction constraint, since an unbounded transformation is involved in the formation of relative clauses. Hence, examples like *Who do you know a man who saw?* or **John is taller than I know a man who is* are excluded by the interaction constraint. The fact that comparative clauses and embedded questions are also "islands" has been less widely discussed in the literature, but is also a consequence of the interaction constraint. Thus, such examples as **Who is John louder than Mary persuaded to be?* or **Who does John wonder when Bill will see?* are excluded because they involve *wh*-movement extracting material from clauses in which *wh*-movement or comparative deletion has taken place. Likewise, comparative clauses are impervious to further applications of comparative deletion: *John was kind to more people than he liked Bill more than I liked* (where this would mean, if grammatical, that the number of people John was kind to exceeded the number of people liked better by Bill than by me). In short, the interaction constraint seems to make the right predictions about a substantial array of data.

Finally, I would like to suggest that the interaction constraint serves not only to restrict the class of grammars made available by linguistic theory, but also to limit the class of languages generable by the available grammars (see Wasow (in press a) for discussion of this distinction). I will not attempt any formal demonstration of this conclusion here, but will sketch briefly why I believe it to be the case. Peters and Ritchie (1973) prove that the language generated by a transformational grammar is recursive if it is possible, on the basis of a surface string, to effectively compute a maximum size of a deep structure from which that string could be derived. The interaction constraint, together with the standard condition on recoverability of deletions (see Peters and Ritchie (1973)), limit the extent to which deletions may shrink a structure. To show why this is the case, it will be useful to invent some terminology: let us call A a parent of B if B can be derived from A by a single application of one transformation. A parent's parent will be called a **grandparent**, and so on. Now consider a string of length n. Because of the recoverability condition, its parent cannot be longer than $2n$ (measuring length in terms of number of terminal symbols). Likewise, its grandparent cannot be longer than $4n$. However, if the grandparent were the full $4n$ long, then the parent would be frozen by the interaction constraint, and the original string would be underivable. In fact, each (length n) half of the parent must have a parent of length no more than $2n-1$, if we are to avoid blocking the derivation by the interaction constraint. Thus, the maximum size of a grandparent is $4n-2$. By similar reasoning it is not hard to see that the maximum size of any ancestor $m+1$ generations removed is $2^m(2n-m)$. Since this number becomes zero when $m=2n$, there is an effective upper bound on the size of any ancestor. Hence, the interaction constraint, together with the standard condition on recoverability of deletions, limits the class of languages which can be generated to a subclass of the recursive sets[8]. This provides yet another point of convergence with computational concerns, since, as noted above, a language must be recursive in order to be effectively processed.

I have sketched a version of transformational grammar which seems to hold considerable promise. There are a number of problems with this approach which I am aware of and undoubtedly many more I am blissfully ignorant of. What I have presented here was intended, more than anything else, as an indication of a program of research, and I have hence felt free to ignore many important issues. The primary point I wish to make is that the study of language appears to have progressed to a point where the concerns of the transformationalist and the concerns of the computational linguist need not conflict, and indeed may be addressed by a single theory.

* I wish to express my gratitude to Adrian Akmajian, Joan Bresnan, and especially Ron Kaplan for very stimulating discussions of some of the material in this paper. They are, of course, absolved of any responsibility for its shortcomings. I am also very grateful to the Xerox Corporation for making its resources human and electronic, available to me in the preparation of this paper. Some of the research reported on here was begun under a Summer Stipend from the National Endowment for the Humanities.

Footnotes

1. No rigorous definition of these notions has ever been offered in the literature, and certain problems with the way they have been used have been pointed out (e.g., Hust and Brame (1976)). I do not wish to commit myself to all of the claims which have been made in the literature about these notions, and my notation below is intended to reflect this. I do, however, believe that those who have discussed thematic relations are onto something important.

2. Obviously, there is more to forming passives than this; for example, I ignore morphology.

3. Those familiar with Postal and Perlmutter's version of relational grammar will recognize the resemblance of last sentence to the Relational Annihilation Law. Notice by the way, that my passive rules say nothing about the *by* phrase. I am assuming, with Bresnan (in press), that there is an independent rule assigning agent status to the objects of some *by* phrases. This rule would operate not only in passives, but also in examples like *The symphony was by Beethoven.*

4. Notice that I am formulating the dative rule "backwards", that is, with the double object construction as the input. My rule says nothing about the prepositions *to* and *for* because I assume that the functional role of their objects will be covered by separate rules, as is the case with *by*. Examples like *John's call was to Mary* and *This present is for you* lend credence to my assumption.

5. This is to be understood as saying that the SO2 will be treated as a predicate, with its own assignments of thematic relations, and with the element in paretheses treated as if it were the SS of that predicate.

6. Jane Robinson has suggested to me that it might be more appropriate semantically to treat the subject of believe as a 3. This would be perfectly compatible with my analysis.

7. My treatment here ignores anaphora rules like VP deletion and sluicing. I am assuming that these rules are not transformations, but a separate category of rules, subject to their own unique conditions (see Wasow (in press b) for discussion).

8. As given, my argument does not take into account root transformations or specified deletions (see Wasow (in press a)). It is quite trivial, however, to extend the argument to cover these cases.

References

Anderson, S. (1977) "Comments on the Paper by Wasow", in Culicover, *et al* (1977).

Brame, M. (1978) "The Base Hypothesis and the Spelling Prohibition". *Linguistic Analysis* 4.1.

Bresnan, J. (1976) "On the Form and Functioning of Transformations". *Linguistic Inquiry* 7.1.

Bresnan, J. (in press) "A Realistic Transformational Grammar", in M. Halle, J. Bresnan, and G. Miller (eds), *Linguistic Theory and Psychological Reality.* MIT Press, Cambridge, Massachusetts.

Chomsky, N. (1965) *Aspects of the Theory of Syntax.* MIT Press, Cambridge, Massachusetts.

Chomsky, N. (1973) "Conditions on Transformations", in S. Anderson and P. Kiparsky (eds), *A Festschrift for Morris Halle.* Holt, Rinehart, and Winston, New York.

Coker, P. and S. Crair (in preparation) "Linguistic Processing: The Grammatical Basis of Sentence Interpretation". Claremont Graduate School, Claremont, California.

Culicover, P. and K. Wexler (1977) "Some Syntactic Implications of a Theory of Language Learnability", in Culicover, *et al* (1977).

Culicover, P., T. Wasow, and A. Akmajian, eds (1977) *Formal Syntax.* Academic Press, New York.

Emonds, J. (1972) "A Reformulation of Certain Syntactic Transformations", in S. Peters (ed), *Goals of Linguistic Theory.* Prentice-Hall, Englewood Cliffs, N.J.

Emonds, J. (1976) *A Transformational Approach to English Syntax: Root, Structure-Preserving and Local Transformations.* Academic Press, New York.

Fodor, J. A., T. Bever, and M. Garrett (1974) *The Psychology of Language.* McGraw-Hill, New York.

Gruber, J. (1965) *Studies in Lexical Relations.* MIT dissertation.

Hust, J. and M. Brame (1976) "Jackendoff on Interpretive Semantics". *Linguistic Analysis* 2.3.

Jackendoff, R. (1972) *Semantic Interpretation in Generative Grammar* MIT Press, Cambridge, Massachusetts.

Osherson, D. and T. Wasow (1976) "Task Specificity and Species Specificity in the Study of Language: A Methodological Note". *Cognition* 4.

Peters, S. and R. Ritchie (1973) "On the Generative Power of Transformational Grammars". *Information Sciences* 6.

Rosenbaum, P. (1967) *The Grammar of English Predicate Complement Constructions.* MIT Press, Cambridge, Massachusetts.

Ross, J. (1967) *Constraints on Variables in Syntax.* MIT dissertation.

Wasow, T. (1977) "Transformations and the Lexicon", in Culicover, *et al* (1977).

Wasow, T. (1978) "Some Thoughts on Mental Representation and Transformational Grammar" Paper delivered at MIT Sloan Foundation Workshop on Mental Representation.

Wasow, T (in press a) "On Constraining the Class of Transformational Languages". *Synthese.*

Wasow, T. (in press b) *Anaphora in Generative Grammar.* Story-Scientia, Ghent.

Woods, W. (1973) "An Experimental Parsing System for Transition Network Grammar", in R. Rustin (ed), *Natural Language Processing.* Algorithmics Press, New York.

List of questions suggested for consideration in each session

Session 1 Language Representation and Psychology

1 How psychologically accurate are different formalisms? (e-g KRL, conceptual dependency diagrams, SCRIPTS, LNR, semantic networks/spreading activation, etc.)

How might we find out such information?
How important is psychological accuracy?
What aspects and consequences of various formalisms lead to implausible cognitive models?

2. How would highly parallel hardware affect representations and processing?

What evidence is there for the action of parallel agents (in Minsky's sense) in language understanding?
How would such a model of cognition affect models of language comprehension?

3 How general are various formalisms?

Are they really ad hoc solutions to relatively narrow domains (e g stories, newspaper articles, data base question-answering, isolated sentences, etc)?
Which could be most successfully generalized?
What problems are still unsolved by any formalism?

Session 2 Language Representation and Reference

It is the hypothesis of this session that entities which can be referred to provide prime evidence for the underlying representation necessary for extended passages of language (narrative text or discourse).

1. What functions do descriptions serve (e.g inferential as well as referential)?

How much inference is necessary to resolve reference?
Are items which can be referenced "naturally" already appropriately organized or "indexed?"
How close is representation to surface structure?
Does representation depend on factors like attention or visualization on the part of a listener?

2. What things can be referred to anaphorically

What things can not?
When (under what circumstances) can they be referenced?
What intervening items can confuse reference or make it impossible?
How are appropriate referring expressions constructed/understood?
Are there differences in the answers for reference by pronoun and definite noun phrases?

3. Is the initial hypothesis above valid?

What other methods can be used to find out about underlying passage representation?

Session 3. Discourse: Speech Acts and Dialogue

1. What sorts of models are necessary? (e.g. self models, other-models, model of "contract," model of topic, etc.).

2. What should be included in a model? (e.g. beliefs, goals, current topic content, current topic constituents, etc.).

3. How domain dependent must the models be?
What signals are used to cue model information?
How are these signals understood?

4. How much information is lost in transcripts of dialogues (i.e. without intonation, body language, etc.)?
Do we use different techniques in writing as substitutes?

5. What makes discourse coherent?
How could we characterize and model what is communicated in a coherent discourse?
What mechanisms are used to relate utterances in a discourse?
What relationships are there between production and comprehension, and how are the models of these processes used?

6. What extra meaning can be conveyed at the phrasal level?
How much depends on being able to "read between the lines" in a dialogue?

Session 4. Language and Preception

1. How are language and perception related?
How closely?
Are natural language primitives related to a priori perceptual entities?
How might we find out?
Are parts of speech perceptually based?

2. What is the function of visual imagery in the understanding of language?
How important is it?

3. Is perceptual experience represented in memory like linguistic experience (e.g. stories)?
If not, how are representations linked or combined?

4. Do all schemata arise from the sensory/motor world?
To what extent should computational linguistics mimic human development?
What are the possibilities for a system to learn language by experience?

Session 5. Inference Mechanisms in Natural Language

1. How can we effectively use multiple descriptions of entities?
Should we?

2. How can presupposition be represented and used in understand generating indirect replies to questions, etc.

3. How is inference controlled?

4. What is the role of deduction in language processing.
What is its relation to inference?

Session 6 Computational Models as a Vehicle for Theoretical Linguistics

1 What can theoretical linguistics learn from computational models that is not accessible by traditional means?

2 What aspects of linguistics have not been fully comprehended or appreciated by computational linguists

What current directions in linguistics are most promising for computational modelling?

3 Is linguistics ripe for a paradigm shift?
Are linguists ready?

4 What problems are most appropriate for each discipline?

How might cooperation and coordination be improved?

5 What are the current views in each field on syntax, sematics and pragmatics?

Why is there widespread disagreement, especially about the role of syntax?