

Interpretation as Abduction

Jerry R. Hobbs, Mark Stickel,
Paul Martin, and Douglas Edwards

Artificial Intelligence Center
SRI International

Abstract

An approach to abductive inference developed in the TACITUS project has resulted in a dramatic simplification of how the problem of interpreting texts is conceptualized. Its use in solving the local pragmatics problems of reference, compound nominals, syntactic ambiguity, and metonymy is described and illustrated. It also suggests an elegant and thorough integration of syntax, semantics, and pragmatics.

1 Introduction

Abductive inference is inference to the best explanation. The process of interpreting sentences in discourse can be viewed as the process of providing the best explanation of why the sentences would be true. In the TACITUS Project at SRI, we have developed a scheme for abductive inference that yields a significant simplification in the description of such interpretation processes and a significant extension of the range of phenomena that can be captured. It has been implemented in the TACITUS System (Stickel, 1982; Hobbs, 1986; Hobbs and Martin, 1987) and has been and is being used to solve a variety of interpretation problems in casualty reports, which are messages about breakdowns in machinery, as well as in other texts.¹

It is well-known that people understand discourse so well because they know so much. Accordingly, the aim of the TACITUS Project has been to investigate how knowledge is used in the interpretation of discourse. This has involved building a large knowledge base of commonsense and domain knowledge (see Hobbs et al., 1986), and developing procedures for using this knowledge for the interpretation of discourse. In the latter effort, we have concentrated on problems in local pragmatics, specifically, the problems of reference resolution, the interpretation of compound nominals, the resolution of some kinds of syntactic ambiguity, and metonymy resolution. Our approach to these problems is the focus of this paper.

In the framework we have developed, what the interpretation of a sentence is can be described very concisely:

¹Charniak (1986) and Norvig (1987) have also applied abductive inference techniques to discourse interpretation.

To interpret a sentence:

- (1) Derive the logical form of the sentence,
together with the constraints that predicates
impose on their arguments,
allowing for coercions,
Merging redundancies where possible,
Making assumptions where necessary.

By the first line we mean "derive in the logical sense, or prove from the predicate calculus axioms in the knowledge base, the logical form that has been produced by syntactic analysis and semantic translation of the sentence."

In a discourse situation, the speaker and hearer both have their sets of private beliefs, and there is a large overlapping set of mutual beliefs. An utterance stands with one foot in mutual belief and one foot in the speaker's private beliefs. It is a bid to extend the area of mutual belief to include some private beliefs of the speaker's. It is anchored referentially in mutual belief, and when we derive the logical form and the constraints, we are recognizing this referential anchor. This is the given information, the definite, the presupposed. Where it is necessary to make assumptions, the information comes from the speaker's private beliefs, and hence is the new information, the indefinite, the asserted. Merging redundancies is a way of getting a minimal, and hence a best, interpretation.²

In Section 2 of this paper, we justify the first clause of the above characterization by showing that solving local pragmatics problems is equivalent to proving the logical form plus the constraints. In Section 3, we justify the last two clauses by describing our scheme of abductive inference. In Section 4 we provide several examples. In Section 5 we describe briefly the type hierarchy that is essential for making abduction work. In Section 6 we discuss future directions.

²Interpreting indirect speech acts, such as "It's cold in here," meaning "Close the window," is not a counterexample to the principle that the minimal interpretation is the best interpretation, but rather can be seen as a matter of achieving the minimal interpretation coherent with the interests of the speaker.

2 Local Pragmatics

The four local pragmatics problems we have addressed can be illustrated by the following "sentence" from the casualty reports:

(2) Disengaged compressor after lube-oil alarm.

Identifying the compressor and the alarm are reference resolution problems. Determining the implicit relation between "lube-oil" and "alarm" is the problem of compound nominal interpretation. Deciding whether "after lube-oil alarm" modifies the compressor or the disengaging is a problem in syntactic ambiguity resolution. The preposition "after" requires an event or condition as its object and this forces us to coerce "lube-oil alarm" into "the sounding of the lube-oil alarm"; this is an example of metonymy resolution. We wish to show that solving the first three of these problems amounts to deriving the logical form of the sentence. Solving the fourth amounts to deriving the constraints predicates impose on their arguments, allowing for coercions. For each of these problems, our approach is to frame a logical expression whose derivation, or proof, constitutes an interpretation.

Reference: To resolve the reference of "compressor" in sentence (1), we need to prove (constructively) the following logical expression:

(3) $(\exists c)compressor(c)$

If, for example, we prove this expression by using axioms that say C_1 is a starting air compressor, and that a starting air compressor is a compressor, then we have resolved the reference of "compressor" to C_1 .

In general, we would expect definite noun phrases to refer to entities the hearer already knows about and can identify, and indefinite noun phrases to refer to new entities the speaker is introducing. However, in the casualty reports most noun phrases have no determiner. There are sentences, such as

Retained oil sample and filter for future analysis.

where "sample" is indefinite, or new information, and "filter" is definite, or already known to the hearer. In this case, we try to prove the existence of both the sample and the filter. When we fail to prove the existence of the sample, we know that it is new, and we simply assume its existence.

Elements in a sentence other than nominals can also function referentially. In

Alarm sounded.

Alarm activated during routine start of compressor.

one can argue that the activation is the same as, or at least implicit in, the sounding. Hence, in addition to trying to derive expressions such as (3) for nominal reference, for possible non-nominal reference we try to prove similar expressions.

$(\exists \dots e, a, \dots) \dots \wedge activate'(e, a) \wedge \dots$ ³

That is, we wish to derive the existence, from background knowledge or the previous text, of some known or implied activation. Most, but certainly not all, information conveyed non-nominally is new, and hence will be assumed.

Compound Nominals: To resolve the reference of the noun phrase "lube-oil alarm", we need to find two entities o and a with the appropriate properties. The entity o must be lube oil, a must be an alarm, and there must be some implicit relation between them. Let us call that implicit relation nn . Then the expression that must be proved is

$(\exists o, a, nn)lube-oil(o) \wedge alarm(a) \wedge nn(o, a)$

In the proof, instantiating nn amounts to interpreting the implicit relation between the two nouns in the compound nominal. Compound nominal interpretation is thus just a special case of reference resolution.

Treating nn as a predicate variable in this way seems to indicate that the relation between the two nouns can be anything, and there are good reasons for believing this to be the case (e.g., Downing, 1977). In "lube-oil alarm", for example, the relation is

$\lambda x, y [y \text{ sounds if pressure of } x \text{ drops too low}]$

However, in our implementation we use a first-order simulation of this approach. The symbol nn is treated as a predicate constant, and the most common possible relations (see Levi, 1978) are encoded in axioms. The axiom

$(\forall x, y)part(y, x) \supset nn(x, y)$

allows interpretation of compound nominals of the form "<whole> <part>", such as "filter element". Axioms of the form

$(\forall x, y)sample(y, x) \supset nn(x, y)$

handle the very common case in which the head noun is a relational noun and the prenominal noun fills one of its roles, as in "oil sample". Complex relations such as the one in "lube-oil alarm" can sometimes be glossed as "for".

$(\forall x, y)for(y, x) \supset nn(x, y)$

Syntactic Ambiguity: Some of the most common types of syntactic ambiguity, including prepositional phrase and other attachment ambiguities and very compound nominal ambiguities, can be converted into constrained coreference problems (see Bear and Hobbs, 1988).

³See Hobbs (1985a) for explanation of this notation for events.

For example, in (2) the first argument of *after* is taken to be an existentially quantified variable which is equal to either the compressor or the alarm. The logical form would thus include

$$(\exists \dots e, c, y, a, \dots) \dots \wedge \text{after}(y, a) \wedge y \in \{c, e\} \\ \wedge \dots$$

That is, however *after*(*y*, *a*) is proved or assumed, *y* must be equal to either the compressor *c* or the disengaging *e*. This kind of ambiguity is often solved as a byproduct of the resolution of metonymy or of the merging of redundancies.

Metonymy: Predicates impose constraints on their arguments that are often violated. When they are violated, the arguments must be coerced into something related which satisfies the constraints. This is the process of metonymy resolution. Let us suppose, for example, that in sentence (2), the predicate *after* requires its arguments to be events:

$$\text{after}(e_1, e_2) : \text{event}(e_1) \wedge \text{event}(e_2)$$

To allow for coercions, the logical form of the sentence is altered by replacing the explicit arguments by "coercion variables" which satisfy the constraints and which are related somehow to the explicit arguments. Thus the altered logical form for (2) would include

$$(\exists \dots k_1, k_2, y, a, \text{rel}_1, \text{rel}_2, \dots) \dots \wedge \text{after}(k_1, k_2) \\ \wedge \text{event}(k_1) \wedge \text{rel}_1(k_1, y) \\ \wedge \text{event}(k_2) \wedge \text{rel}_2(k_2, a) \wedge \dots$$

As in the most general approach to compound nominal interpretation, this treatment is second-order, and suggests that any relation at all can hold between the implicit and explicit arguments. Nunberg (1978), among others, has in fact argued just this point. However, in our implementation, we are using a first-order simulation. The symbol *rel* is treated as a predicate constant, and there are a number of axioms that specify what the possible coercions are. Identity is one possible relation, since the explicit arguments could in fact satisfy the constraints.

$$(\forall x) \text{rel}(x, x)$$

In general, where this works, it will lead to the best interpretation. We can also coerce from a whole to a part and from an object to its function. Hence,

$$(\forall x, y) \text{part}(x, y) \supset \text{rel}(x, y)$$

$$(\forall x, e) \text{function}(e, x) \supset \text{rel}(e, x)$$

Putting it all together, we find that to solve all the local pragmatics problems posed by sentence (2), we must derive the following expression:

$$(\exists e, x, c, k_1, k_2, y, a, o) \text{Past}(e) \\ \wedge \text{disengage}'(e, x, c) \\ \wedge \text{compressor}(c) \wedge \text{after}(k_1, k_2) \\ \wedge \text{event}(k_1) \wedge \text{rel}(k_1, y) \wedge y \in \{c, e\} \\ \wedge \text{event}(k_2) \wedge \text{rel}(k_2, a) \wedge \text{alarm}(a) \\ \wedge \text{nn}(o, a) \wedge \text{lube-oil}(o)$$

But this is just the logical form of the sentence⁴ together with the constraints that predicates impose on their arguments, allowing for coercions. That is, it is the first half of our characterization (1) of what it is to interpret a sentence.

When parts of this expression cannot be derived, assumptions must be made, and these assumptions are taken to be the new information. The likelihood of different atoms in this expression being new information varies according to how the information is presented, linguistically. The main verb is more likely to convey new information than a definite noun phrase. Thus, we assign a cost to each of the atoms—the cost of assuming that atom. This cost is expressed in the same currency in which other factors involved in the "goodness" of an interpretation are expressed; among these factors are likely to be the length of the proofs used and the salience of the axioms they rely on. Since a definite noun phrase is generally used referentially, an interpretation that simply assumes the existence of the referent and thus fails to identify it should be an expensive one. It is therefore given a high assumability cost. For purposes of concreteness, let's call this \$10. Indefinite noun phrases are not usually used referentially, so they are given a low cost, say, \$1. Bare noun phrases are given an intermediate cost, say, \$5. Propositions presented non-nominally are usually new information, so they are given a low cost, say, \$3. One does not usually use selectional constraints to convey new information, so they are given the same cost as definite noun phrases. Coercion relations and the compound nominal relations are given a very high cost, say, \$20, since to assume them is to fail to solve the interpretation problem. If we superscript the atoms in the above logical form by their assumability costs, we get the following expression:

$$(\exists e, x, c, k_1, k_2, y, a, o) \text{Past}(e)^{\$3} \\ \wedge \text{disengage}'(e, x, c)^{\$3} \\ \wedge \text{compressor}(c)^{\$5} \wedge \text{after}(k_1, k_2)^{\$3} \\ \wedge \text{event}(k_1)^{\$10} \wedge \text{rel}(k_1, y)^{\$20} \wedge y \in \{c, e\} \\ \wedge \text{event}(k_2)^{\$10} \wedge \text{rel}(k_2, a)^{\$20} \wedge \text{alarm}(a)^{\$5} \\ \wedge \text{nn}(o, a)^{\$20} \wedge \text{lube-oil}(o)^{\$5}$$

While this example gives a rough idea of the relative assumability costs, the real costs must mesh well with the inference processes and thus must be determined experimentally. The use of numbers here and throughout the next section constitutes one possible regime with the needed properties. We are at present working, and with some optimism, on a semantics for the numbers and the procedures that operate on them. In the course of this work, we may modify the procedures to an extent, but we expect to retain their essential properties.

⁴For justification for this kind of logical form for sentences with quantifiers and intensional operators, see Hobbs(1983) and Hobbs (1985a).

3 Abduction

We now argue for the last half of the characterization (1) of interpretation.

Abduction is the process by which, from $(\forall x)p(x) \supset q(x)$ and $q(A)$, one concludes $p(A)$. One can think of $q(A)$ as the observable evidence, of $(\forall x)p(x) \supset q(x)$ as a general principle that could explain $q(A)$'s occurrence, and of $p(A)$ as the inferred, underlying cause of $q(A)$. Of course, this mode of inference is not valid; there may be many possible such $p(A)$'s. Therefore, other criteria are needed to choose among the possibilities. One obvious criterion is consistency of $p(A)$ with the rest of what one knows. Two other criteria are what Thagard (1978) has called *consilience* and *simplicity*. Roughly, *simplicity* is that $p(A)$ should be as small as possible, and *consilience* is that $q(A)$ should be as big as possible. We want to get more bang for the buck, where $q(A)$ is bang, and $p(A)$ is buck.

There is a property of natural language discourse, noticed by a number of linguists (e.g., Joos (1972), Wilks (1972)), that suggests a role for *simplicity* and *consilience* in its interpretation—its high degree of redundancy. Consider

Inspection of oil filter revealed metal *particles*.

An inspection is a looking at that *causes one to learn* a property relevant to the *function* of the inspected object. The *function* of a filter is to capture *particles* from a fluid. To reveal is to *cause one to learn*. If we assume the two causings to learn are identical, the two sets of particles are identical, and the two functions are identical, then we have explained the sentence in a minimal fashion. A small number of inferences and assumptions have explained a large number of syntactically independent propositions in the sentence. As a byproduct, we have moreover shown that the inspector is the one to whom the particles are revealed and that the particles are in the filter.

Another issue that arises in abduction is what might be called the "informativeness-correctness tradeoff". Most previous uses of abduction in AI from a theorem-proving perspective have been in diagnostic reasoning (e.g., Pople, 1973; Cox and Pietrzykowski, 1986), and they have assumed "most specific abduction". If we wish to explain chest pains, it is not sufficient to assume the cause is simply chest pains. We want something more specific, such as "pneumonia". We want the most specific possible explanation. In natural language processing, however, we often want the least specific assumption. If there is a mention of a fluid, we do not necessarily want to assume it is lube oil. Assuming simply the existence of a fluid may be the best we can do.⁵ However, if there is corroborating evidence, we may want to make a more specific assumption. In

Alarm sounded. Flow obstructed.

⁵Sometimes a cigar is just a cigar.

we know the alarm is for the lube oil pressure, and this provides evidence that the flow is not merely of a fluid but of lube oil. The more specific our assumptions are, the more informative our interpretation is. The less specific they are, the more likely they are to be correct.

We therefore need a scheme of abductive inference with three features. First, it should be possible for goal expressions to be assumable; at varying costs. Second, there should be the possibility of making assumptions at various levels of specificity. Third, there should be a way of exploiting the natural redundancy of texts.

We have devised just such an abduction scheme.⁶ First, every conjunct in the logical form of the sentence is given an *assumability cost*, as described at the end of Section 2. Second, this cost is passed back to the antecedents in Horn clauses by assigning weights to them. Axioms are stated in the form

$$(4) P_1^{w_1} \wedge P_2^{w_2} \supset Q$$

This says that P_1 and P_2 imply Q , but also that if the cost of assuming Q is c , then the cost of assuming P_1 is w_1c , and the cost of assuming P_2 is w_2c . Third, factoring or synthesis is allowed. That is, goal wffs may be unified, in which case the resulting wff is given the smaller of the costs of the input wffs. This feature leads to minimality through the exploitation of redundancy.

Note that in (4), if $w_1 + w_2 < 1$, most specific abduction is favored—why assume Q when it is cheaper to assume P_1 and P_2 . If $w_1 + w_2 > 1$, least specific abduction is favored—why assume P_1 and P_2 when it is cheaper to assume Q . But in

$$P_1^6 \wedge P_2^6 \supset Q$$

if P_1 has already been derived, it is cheaper to assume P_2 than Q . P_1 has provided evidence for Q , and assuming the "remainder" P_2 of the necessary evidence for Q should be cheaper.

Factoring can also override least specific abduction. Suppose we have the axioms

$$P_1^6 \wedge P_2^6 \supset Q_1$$

$$P_2^6 \wedge P_3^6 \supset Q_2$$

and we wish to derive $Q_1 \wedge Q_2$, where each conjunct has an assumability cost of \$10. Then assuming $Q_1 \wedge Q_2$ will cost \$20, whereas assuming $P_1 \wedge P_2 \wedge P_3$ will cost only \$18, since the two instances of P_2 can be unified. Thus, the abduction scheme allows us to adopt the careful policy of favoring least specific abduction while also allowing us to exploit the redundancy of texts for more specific interpretations.

In the above examples we have used equal weights on the conjuncts in the antecedents. It is more reasonable,

⁶The abduction scheme is due to Mark Stickel, and it, or a variant of it, is described at greater length in Stickel (1988).

however, to assign the weights according to the “semantic contribution” each conjunct makes to the consequent. Consider, for example, the axiom

$$(\forall x)car(x)^8 \wedge no-top(x)^4 \supset convertible(x)$$

We have an intuitive sense that *car* contributes more to *convertible* than *no-top* does.⁷ In principle, the weights in (4) should be a function of the probabilities that instances of the concept *P*_i are instances of the concept *Q* in the corpus of interest. In practice, all we can do is assign weights by a rough, intuitive sense of semantic contribution, and refine them by successive approximation on a representative sample of the corpus.

One would think that since we are deriving the logical form of the sentence, rather than determining what can be inferred from the logical form of the sentence, we could not use superset information in processing the sentence. That is, since we are back-chaining from the propositions in the logical form, the fact that, say, lube oil is a fluid, which would be expressed as

$$(5) (\forall x)lube-oil(x) \supset fluid(x)$$

could not play a role in the analysis. Thus, in the text

Flow obstructed. Metal particles in lube oil filter.

we know from the first sentence that there is a fluid. We would like to identify it with the lube oil mentioned in the second sentence. In interpreting the second sentence, we must prove the expression

$$(\exists x)lube-oil(x)$$

If we had as an axiom

$$(\forall x)fluid(x) \supset lube-oil(x)$$

then we could establish the identity. But of course we don't have such an axiom, for it isn't true. There are lots of other kinds of fluids. There would seem to be no way to use superset information in our scheme.

Fortunately, however, there is a way. We can make use of this information by converting the axiom into a biconditional. In general, axioms of the form

$$species \supset genus$$

can be converted into a biconditional axiom of the form

$$genus \wedge differentiae \equiv species$$

⁷To prime this intuition, imagine two doors. Behind one is a car. Behind the other is something with no top. You pick a door. If there's a convertible behind it, you get to keep it. Which door would you pick?

Often, of course, as in the above example, we will not be able to prove the differentiae, and in many cases the differentiae can not even be spelled out. But in our abductive scheme, this does not matter. They can simply be assumed. In fact, we need not state them explicitly. We can simply introduce a predicate which stands for all the remaining properties. It will never be provable, but it will be assumable. Thus, we can rewrite (5) as

$$(\forall x)fluid(x) \wedge etc_1(x) \equiv lube-oil(x)$$

Then the fact that something is fluid can be used as evidence for its being lube oil. With the weights distributed according to semantic contribution, we can go to extremes and use an axiom like

$$(\forall x)mammal(x)^2 \wedge etc_2(x)^9 \supset elephant(x)$$

to allow us to use the fact that something is a mammal as (weak) evidence that it is an elephant.

In principle, one should try to prove the entire logical form of the sentence and the constraints at once. In this global strategy, any heuristic ordering of the individual problems is done by the theorem prover. From a practical point of view, however, the global strategy generally takes longer, sometimes significantly so, since it presents the theorem-prover with a longer expression to be proved. We have experimented both with this strategy and with a bottom-up strategy in which, for example, we try to identify the lube oil before trying to identify the lube oil alarm. The latter is quicker since it presents the theorem-prover with problems in a piecemeal fashion, but the former frequently results in better interpretations since it is better able to exploit redundancies. The analysis of the sentence in Section 4.2 below, for example, requires either the global strategy or very careful axiomatization. The bottom-up strategy, with only a view of a small local region of the sentence, cannot recognize and capitalize on redundancies among distant elements in the sentence. Ideally, we would like to have detailed control over the proof process to allow a number of different factors to interact in determining the allocation of deductive resources. Among such factors would be word order, lexical form, syntactic structure, topic-comment structure, and, in speech, pitch accent.⁸

4 Examples

4.1 Distinguishing the Given and New

We will examine two difficult definite reference problems in which the given and the new information are intertwined and must be separated. In the first, new and old information about the same entity are encoded in a single noun phrase.

⁸Pereira and Pollack's CANDIDE system (1988) is specifically designed to aid investigation of the question of the most effective order of interpretation.

There was adequate lube oil.

We know about the lube oil already, and there is a corresponding axiom in the knowledge base.

$lube-oil(O)$

Its adequacy is new information, however. It is what the sentence is telling us.

The logical form of the sentence is, roughly,

$(\exists o)lube-oil(o) \wedge adequate(o)$

This is the expression that must be derived. The proof of the existence of the lube oil is immediate. It is thus old information. The adequacy can't be proved, and is hence assumed as new information.

The second example is from Clark (1975), and illustrates what happens when the given and new information are combined into a single lexical item.

John walked into the room.

The chandelier shone brightly.

What chandelier is being referred to?

Let us suppose we have in our knowledge base the fact that rooms have lights.

(6) $(\forall r)room(r) \supset (\exists l)light(l) \wedge in(l,r)$

Suppose we also have the fact that lights with numerous fixtures are chandeliers.

(7) $(\forall l)light(l) \wedge has-fixtures(l) \supset chandelier(l)$

The first sentence has given us the existence of a room— $room(R)$. To solve the definite reference problem in the second sentence, we must prove the existence of a chandelier. Back-chaining on axiom (7), we see we need to prove the existence of a light with fixtures. Back-chaining from $light(l)$ in axiom (6), we see we need to prove the existence of a room. We have this in $room(R)$. To complete the derivation, we assume the light l has fixtures. The light is thus given by the room mentioned in the previous sentence, while the fact that it has fixtures is new information.

4.2 Exploiting Redundancy

We next show the use of the abduction scheme in solving internal coreference problems. Two problems raised by the sentence

The plain was reduced by erosion to its present level.

are determining what was eroding and determining what "it" refers to. Suppose our knowledge base consists of the following axioms:

$(\forall p, l, s)decrease(p, l, s) \wedge vertical(s)$
 $\wedge etc_3(p, l, s) \equiv (\exists e_1)reduce'(e_1, p, l)$

or e_1 is a reduction of p to l if and only if p decreases to l on some vertical scale s (plus some other conditions).

$(\forall p)landform(p) \wedge flat(p) \wedge etc_4(p) \equiv plain(p)$

or p is a plain if and only if p is a flat landform (plus some other conditions).

$(\forall e, y, l, s)at'(e, y, l) \wedge on(l, s) \wedge vertical(s)$
 $\wedge flat(y) \wedge etc_5(e, y, l, s) \equiv level'(e, l, y)$

or e is the condition of l 's being the level of y if and only if e is the condition of y 's being at l on some vertical scale s and y is flat (plus some other conditions).

$(\forall x, l, s)decrease(x, l, s) \wedge landform(x)$
 $\wedge altitude(s) \wedge etc_6(y, l, s) \equiv (\exists e)erode'(e, x)$

or e is an eroding of x if and only if x is a landform that decreases to some point l on the altitude scale s (plus some other conditions).

$(\forall s)vertical(s) \wedge etc_7(p) \equiv altitude(s)$

or s is the altitude scale if and only if s is vertical (plus some other conditions).

Now the analysis. The logical form of the sentence is roughly

$(\exists e_1, p, l, x, e_2, y)reduce'(e_1, p, l) \wedge plain(p)$
 $\wedge erode'(e_1, x) \wedge present(e_2) \wedge level'(e_2, l, y)$

Our characterization of interpretation says that we must derive this expression from the axioms or from assumptions. Back-chaining on $reduce'(e_1, p, l)$ yields

$decrease(p, l, s_1) \wedge vertical(s_1) \wedge etc_3(p, l, s_1)$

Back-chaining on $erode'(e_1, x)$ yields

$decrease(x, l_2, s_2) \wedge landform(x) \wedge altitude(s_2)$
 $\wedge etc_6(x, l_2, s_2)$

and back-chaining on $altitude(s_2)$ in turn yields

$vertical(s_2) \wedge etc_7(s_2)$

We unify the goals $decrease(p, l, s_1)$ and $decrease(x, l_2, s_2)$, and thereby identify the object of the erosion with the plain. The goals $vertical(s_1)$ and $vertical(s_2)$ also unify, telling us the reduction was on the altitude scale. Back-chaining on $plain(p)$ yields

$landform(p) \wedge flat(p) \wedge etc_4(p)$

and $landform(x)$ unifies with $landform(p)$, reinforcing our identification of the object of the erosion with the plain. Back-chaining on $level'(e_2, l, y)$ yields

$$at'(e_2, y, l) \wedge on(l, s_3) \wedge vertical(s_3) \wedge flat(y) \\ \wedge etc_5(p)$$

and $vertical(s_3)$ and $vertical(s_2)$ unify, as do $flat(y)$ and $flat(p)$, thereby identifying "it", or y , as the plain p . We have not written out the axioms for this, but note also that "present" implies the existence of a change of level, or a change in the location of "it" on a vertical scale, and a decrease of a plain is a change of the plain's location on a vertical scale. Unifying these would provide reinforcement for our identification of "it" with the plain. Now assuming the most specific atoms we have derived including all the "et cetera" conditions, we arrive at an interpretation that is minimal and that solves the internal coreference problems as a byproduct.

4.3 A Thorough Integration of Syntax, Semantics, and Pragmatics

By combining the idea of interpretation as abduction with the older idea of parsing as deduction (Kowalski, 1980, pp. 52-53; Pereira and Warren, 1983), it becomes possible to integrate syntax, semantics, and pragmatics in a very thorough and elegant way.⁹ Below is a simple grammar written in Prolog style, but incorporating calls to local pragmatics. The syntax portion is represented in standard Prolog manner, with nonterminals treated as predicates and having as two of its arguments the beginning and end points of the phrase spanned by the nonterminal. The one modification we would have to make to the abduction scheme is to allow conjuncts in the antecedents to take costs directly as well as weights. Constraints on the application of phrase structure rules have been omitted, but could be incorporated in the usual way.

$$\begin{aligned} &(\forall i, j, k, x, p, args, req, e, c, rel) np(i, j, x) \\ &\quad \wedge vp(j, k, p, args, req) \wedge p'(e, c)^{\$3} \wedge rel(c, x)^{\$20} \\ &\quad \wedge subst(req, cons(c, args))^{\$10} \supset s(i, k, e) \\ &(\forall i, j, k, e, p, args, req, e_1, c, rel) s(i, j, e) \\ &\quad \wedge pp(j, k, p, args, req) \wedge p'(e_1, c)^{\$3} \wedge rel(c, e)^{\$20} \\ &\quad \wedge subst(req, cons(c, args))^{\$10} \supset s(i, k, e \& e_1) \\ &(\forall i, j, k, w, x, c, rel) v(i, j, w) \wedge np(j, k, x) \\ &\quad \wedge rel(c, x)^{\$20} \\ &\quad \supset vp(i, k, \lambda z[w(z, c)], \langle c \rangle, Req(w)) \\ &(\forall i, j, k, x) det(i, j, \text{"the"}) \wedge cn(j, k, x, p) \\ &\quad \wedge p(x)^{\$10} \supset np(i, k, x) \\ &(\forall i, j, k, x) det(i, j, \text{"a"}) \wedge cn(j, k, x, p) \wedge p(x)^{\$1} \\ &\quad \supset np(i, k, x) \\ &(\forall i, j, k, w, x, y, p, nn) n(i, j, w) \wedge cn(j, k, x, p) \\ &\quad \wedge w(y)^{\$5} \wedge nn(y, x)^{\$20} \supset cn(i, k, x, p) \\ &(\forall i, j, k, x, p_1, p_2, args, req, c, rel) cn(i, j, x, p_1) \\ &\quad \wedge pp(j, k, p_2, args, req) \end{aligned}$$

⁹This idea is due to Stuart Shieber.

$$\begin{aligned} &\wedge subst(req, cons(c, args))^{\$10} \wedge rel(c, x)^{\$20} \\ &\quad \supset cn(i, k, x, \lambda z[p_1(z) \wedge p_2(z)]) \\ &(\forall i, j, w) n(i, j, w) \supset (\exists x) cn(i, j, x, w) \\ &(\forall i, j, k, w, x, c, rel) prep(i, j, w) \wedge np(j, k, x) \\ &\quad \wedge rel(c, x)^{\$20} \\ &\quad \supset pp(i, k, \lambda z[w(c, z)], \langle c \rangle, Req(w)) \end{aligned}$$

For example, the first axiom says that there is a sentence from point i to point k asserting eventuality e if there is a noun phrase from i to j referring to x and a verb phrase from j to k denoting predicate p with arguments $args$ and having an associated requirement req , and there is (or, for \$3, can be assumed to be) an eventuality e of p 's being true of c , where c is related to or coercible from x (with an assumability cost of \$20), and the requirement req associated with p can be proved or, for \$10, assumed to hold of the arguments of p . The symbol $e \& e_1$ denotes the conjunction of eventualities e and e_1 (See Hobbs (1985b), p. 35.) The third argument of predicates corresponding to terminal nodes such as n and det is the word itself, which then becomes the name of the predicate. The function Req returns the requirements associated with a predicate, and $subst$ takes care of substituting the right arguments into the requirements. $\langle c \rangle$ is the list consisting of the single element c , and $cons$ is the LISP function $cons$. The relations rel and nn are treated here as predicate variables, but they could be treated as predicate constants, in which case we would not have quantified over them.

In this approach, $s(0, n, e)$ can be read as saying there is an *interpretable* sentence from point 0 to point n (asserting e). Syntax is captured in predicates like np , vp , and s . Compositional semantics is encoded in, for example, the way the predicate p' is applied to its arguments in the first axiom, and in the lambda expression in the third argument of vp in the third axiom. Local pragmatics is captured by virtue of the fact that in order to prove $s(0, n, e)$, one must derive the logical form of the sentence together with the constraints predicates impose on their arguments, allowing for metonymy.

Implementations of different orders of interpretation, or different sorts of interaction among syntax, compositional semantics, and local pragmatics, can then be seen as different orders of search for a proof of $s(0, n, e)$. In a syntax-first order of interpretation, one would try first to prove all the "syntactic" atoms, such as $np(i, j, x)$, before any of the "local pragmatic" atoms, such as $p'(e, c)$. Verb-driven interpretation would first try to prove $vp(j, k, p, args, req)$ by proving $v(i, j, w)$ and then using the information in the requirements associated with the verb to drive the search for the arguments of the verb, by deriving $subst(req, cons(c, args))$ before trying to prove the various np atoms. But more fluid orders of interpretation are obviously possible. This formulation allows one to prove those things first which are easiest to prove. It is also easy to see how processing could occur in parallel.

It is moreover possible to deal with ill-formed or unclear input in this framework, by having axioms such as this revision of our first axiom above.

$$\begin{aligned} & (\forall i, j, k, x, p, args, req, e, c, rel) np(i, j, x)^4 \\ & \quad \wedge vp(j, k, p, args, req)^5 \wedge p'(e, c)^{23} \\ & \quad \wedge rel(c, x)^{20} \wedge subst(req, cons(c, args))^{10} \\ & \quad \supset s(i, k, e) \end{aligned}$$

This says that a verb phrase provides more evidence for a sentence than a noun phrase does, but either one can constitute a sentence if the string of words is otherwise interpretable.

It is likely that this approach could be extended to speech recognition by using Prolog-style rules to decompose morphemes into their phonemes and weighting them according to their acoustic prominence.

5 Controlling Abduction: Type Hierarchy

The first example on which we tested the new abductive scheme was the sentence

There was adequate lube oil.

The system got the correct interpretation, that the lube oil was the lube oil in the lube oil system of the air compressor, and it assumed that that lube oil was adequate. But it also got another interpretation. There is a mention in the knowledge base of the adequacy of the lube oil pressure, so it identified that adequacy with the adequacy mentioned in the sentence. It then assumed that the pressure was lube oil.

It is clear what went wrong here. Pressure is a magnitude whereas lube oil is a material, and magnitudes can't be materials. In principle, abduction requires a check for the consistency of what is assumed, and our knowledge base should have contained axioms from which it could be inferred that a magnitude is not a material. In practice, unconstrained consistency checking is undecidable and, at best, may take a long time. Nevertheless, one can, through the use of a type hierarchy, eliminate a very large number of possible assumptions that are likely to result in an inconsistency. We have consequently implemented a module which specifies the types that various predicate-argument positions can take on, and the likely disjointness relations among types. This is a way of exploiting the specificity of the English lexicon for computational purposes. This addition led to a speed-up of two orders of magnitude.

There is a problem, however. In an ontologically promiscuous notation, there is no commitment in a primed proposition to truth or existence in the real world. Thus, *lube-oil'(e, o)* does not say that *o* is lube oil or even that it exists; rather it says that *e* is the eventuality of *o*'s being lube oil. This eventuality may or may not exist in the real

world. If it does, then we would express this as *Exists(e)*, and from that we could derive from axioms the existence of *o* and the fact that it is lube oil. But *e*'s existential status could be something different. For example, *e* could be nonexistent, expressed as *not(e)* in the notation, and in English as "The eventuality *e* of *o*'s being lube oil does not exist," or as "*o* is not lube oil." Or *e* may exist only in someone's beliefs. While the axiom

$$(\forall x) pressure(x) \supset \neg lube-oil(x)$$

is certainly true, the axiom

$$(\forall e_1, x) pressure'(e_1, x) \supset \neg(\exists e_2) lube-oil'(e_2, x)$$

would not be true. The fact that a variable occupies the second argument position of the predicate *lube-oil'* does not mean it is lube oil. We cannot properly restrict that argument position to be lube oil, or fluid, or even a material, for that would rule out perfectly true sentences like "Truth is not lube oil."

Generally, when one uses a type hierarchy, one assumes the types to be disjoint sets with cleanly defined boundaries, and one assumes that predicates take arguments of only certain types. There are a lot of problems with this idea. In any case, in our work, we are not buying into this notion that the universe is typed. Rather we are using the type hierarchy strictly as a heuristic, as a set of guesses not about what could or could not be but about what it would or would not occur to someone to say. When two types are declared to be disjoint, we are saying that they are certainly disjoint in the real world, and that they are very probably disjoint everywhere except in certain bizarre modal contexts. This means, however, that we risk failing on certain rare examples. We could not, for example, deal with the sentence, "It then assumed that the pressure was lube oil."

6 Future Directions

Deduction is explosive, and since the abduction scheme augments deduction with the assumptions, it is even more explosive. We are currently engaged in an empirical investigation of the behavior of this abductive scheme on a very large knowledge base performing sophisticated processing. In addition to type checking, we have introduced two other techniques that are necessary for controlling the explosion—unwinding recursive axioms and making use of syntactic noncoreference information. We expect our investigation to continue to yield techniques for controlling the abduction process.

We are also looking toward extending the interpretation processes to cover lexical ambiguity, quantifier scope ambiguity and metaphor interpretation problems as well. We will also be investigating the integration proposed in Section 4.3 and an approach that integrates all of this with the recognition of discourse structure and the recognition of relations between utterances and the hearer's interests.

Acknowledgements

The authors have profited from discussions with Todd Davies, John Lowrance, Stuart Shieber, and Mabry Tyson about this work. The research was funded by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-85-C-0013.

References

- [1] Bear, John, and Jerry R. Hobbs, 1988. "Localizing the Expression of Ambiguity", *Proceedings, Second Conference on Applied Natural Language Processing*, Austin, Texas, February, 1988.
- [2] Charniak, Eugene, 1986. "A Neat Theory of Marker Passing", *Proceedings, AAAI-86, Fifth National Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, pp. 584-588.
- [3] Clark, Herbert, 1975. "Bridging". In R. Schank and B. Nash-Webber (Eds.), *Theoretical Issues in Natural Language Processing*, pp. 169-174. Cambridge, Massachusetts.
- [4] Cox, P. T., and T. Pietrzykowski, 1986. "Causes for Events: Their Computation and Applications", *Proceedings, CADE-8*.
- [5] Downing, Pamela, 1977. "On the Creation and Use of English Compound Nouns", *Language*, vol. 53, no. 4, pp. 810-842.
- [6] Hobbs, Jerry R., 1983. "An Improper Treatment of Quantification in Ordinary English", *Proceedings of the 21st Annual Meeting, Association for Computational Linguistics*, pp. 57-63. Cambridge, Massachusetts, June 1983.
- [7] Hobbs, Jerry R. 1985a. "Ontological promiscuity." *Proceedings, 23rd Annual Meeting of the Association for Computational Linguistics*, pp. 61-69.
- [8] Hobbs, Jerry R., 1985b, "The Logical Notation: Ontological Promiscuity", manuscript.
- [9] Hobbs, Jerry (1986) "Overview of the TACITUS Project", *CL*, Vol. 12, No. 3.
- [10] Hobbs, Jerry R., William Croft, Todd Davies, Douglas Edwards, and Kenneth Laws, 1986. "Commonsense Metaphysics and Lexical Semantics", *Proceedings, 24th Annual Meeting of the Association for Computational Linguistics*, New York, June 1986., pp. 231-240.
- [11] Hobbs, Jerry R., and Paul Martin 1987. "Local Pragmatics". *Proceedings, International Joint Conference on Artificial Intelligence*, pp. 520-523. Milano, Italy, August 1987.
- [12] Joss, Martin, 1972. "Semantic Axiom Number One", *Language*, pp. 257-265.
- [13] Kowalski, Robert, 1980. *The Logic of Problem Solving*, North Holland, New York.
- [14] Levi, Judith, 1978. *The Syntax and Semantics of Complex Nominals*, Academic Press, New York.
- [15] Norvig, Peter, 1987. "Inference in Text Understanding", *Proceedings, AAAI-87, Sixth National Conference on Artificial Intelligence*, Seattle, Washington, July 1987.
- [16] Nunberg, Geoffery, 1978. "The Pragmatics of Reference", Ph. D. thesis, City University of New York, New York.
- [17] Pereira, Fernando C. N., and Martha E. Pollack, 1988. "An Integrated Framework for Semantic and Pragmatic Interpretation", to appear in *Proceedings, 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York, June 1988.
- [18] Pereira, Fernando C. N., and David H. D. Warren, 1983. "Parsing as Deduction", *Proceedings of the 21st Annual Meeting, Association for Computational Linguistics*, pp. 137-144. Cambridge, Massachusetts, June 1983.
- [19] Pople, Harry E., Jr., 1973, "On the Mechanization of Abductive Logic", *Proceedings, Third International Joint Conference on Artificial Intelligence*, pp. 147-152, Stanford, California, August 1973.
- [20] Stickel, Mark E., 1982. "A Nonclausal Connection-Graph Theorem-Proving Program", *Proceedings, AAAI-82 National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, pp. 229-233.
- [21] Stickel, Mark E., 1988. "A Prolog-like Inference System for Computing Minimum-Cost Abductive Explanations in Natural-Language Interpretation", forthcoming.
- [22] Thagard, Paul R., 1978. "The Best Explanation: Criteria for Theory Choice", *The Journal of Philosophy*, pp. 76-92.
- [23] Wilks, Yorick, 1972. *Grammar, Meaning, and the Machine Analysis of Language*, Routledge and Kegan Paul, London.