

# An Integrated Framework for Semantic and Pragmatic Interpretation<sup>1</sup>

Martha E. Pollack  
Artificial Intelligence Center  
SRI International  
333 Ravenswood Ave  
Menlo Park, CA 94025, USA

Fernando C.N. Pereira  
Cambridge Computer Science Research Centre  
SRI International  
Suite 23, Millers Yard, Mill Lane  
Cambridge CB2 1RQ, England

## Abstract

We report on a mechanism for semantic and pragmatic interpretation that has been designed to take advantage of the generally compositional nature of semantic analysis, without unduly constraining the order in which pragmatic decisions are made. To achieve this goal, we introduce the idea of a *conditional interpretation*: one that depends upon a set of assumptions about subsequent pragmatic processing. Conditional interpretations are constructed compositionally according to a set of declaratively specified interpretation rules. The mechanism can handle a wide range of pragmatic phenomena and their interactions.

## 1 Introduction

Compositional systems of semantic interpretation, while logically and computationally very attractive [6,20,26], seem unable to cope with the fact that the syntactic form of an utterance is not the only source of information about its meaning. Contextual information—information about the world and about the history of a discourse— influences not only an utterance's meaning, but even its preferred syntactic analysis [3,5,7,16]. Of course, context also influences the interpretation (or meaning in context) of the utterance, in which, for example, referring expressions have been resolved.

One possible solution is to move to an integrated system of semantic and pragmatic interpretation, defined recursively on syntactic analyses that are

neutral about those decisions that depend upon context. In this approach, a *least-commitment grammar* may be used to produce neutral representations that can be reconfigured later. Such a grammar might, for example, leave quantifiers in place [30], attach all prepositional phrases low and right [22], and bracket to the right all compound nominals.<sup>2</sup> These neutral analyses can then serve as input to a system that produces interpretations (and not meanings) in a nearly compositional manner, in that the interpretation of a phrase<sup>3</sup> is a function of the interpretations of its syntactic constituents together with its context of utterance.

This model of semantic interpretation assumes that contextual information is available whenever it is needed for deciding among alternative interpretations. However, this is often not the case: questions about the interpretation of some constituent of an utterance might be answerable only when information about the interpretation of syntactically distant constituents becomes available. Familiar examples of this can be found, for instance, in sentences with quantifier scoping ambiguities and in sentences that include intrasentential anaphora. The so-called donkey sentences [9] exhibit both these phenomena.

These difficulties do not necessitate a complete abandonment of compositionality. To take advantage of the generally compositional nature of semantic analysis without constraining unduly the order in which pragmatic decisions are made, we assign to phrases *conditional interpretations*,

<sup>1</sup>This research has been funded by DARPA under Contract N00039-84-C-0524, and by a gift from the Systems Development Foundation as part of a coordinated research effort with the Center for the Study of Language and Information, Stanford University. We would like to thank David Israel, Ray Perrault, and Stuart Shieber for their helpful discussions regarding this work.

<sup>2</sup>There are reasons to suspect that ultimately syntactic analysis should be incorporated into the same stage of processing as semantic and pragmatic analysis; in particular, it is difficult to develop syntactically neutral representations for certain constructions such as conjunction.

<sup>3</sup>For simplicity, we shall use the term "phrase" to refer both to an entire utterance and to a constituent of an utterance, distinguishing between the two only when needed.

which represent the dependence of a phrase's interpretation on assumptions about subsequent pragmatic processing. Conditional interpretations are built compositionally according to declaratively specified interpretation rules.

The interpretation mechanism we discuss here has been implemented in Prolog as part of the Candide system, a multimodal tool for knowledge acquisition. Incorporating both a graphical interface and a processor for English discourse, Candide allows a user of the Procedural Reasoning System (PRS) [10] to build and maintain procedural networks in a natural way. Procedural networks, an essential part of PRS's knowledge base, encode the information about procedures that is used by PRS for reasoning about and performing tasks in any given domain. The current version of Candide has been used to construct networks for malfunction procedures for NASA's space shuttle. Further details of the Candide system will be presented elsewhere [24].

## 2 Conditional Interpretations

In our approach to semantic and pragmatic interpretation, conditional interpretations separate the context-independent aspects of an interpretation from those that are context-dependent. Each conditional interpretation consists of a *sense* and a [possibly empty] set of *assumptions*. As a first approximation, one might think of the sense of a phrase as representing purely semantic information, that is, information that can be adduced solely from the linguistic content of the phrase, no matter in which context the phrase has been uttered. The assumptions then represent constraints relating the phrase's sense to its ultimate interpretation. A *complete* interpretation has an empty assumption set, indicating that all of its dependencies on context have been resolved.

The present version of the theory allows for two kinds of assumptions. A *bind assumption* introduces a new parameter in an interpretation and places constraints on the binding of the parameter to individuals in the context. A *restrict assumption* does not introduce a new parameter, but instead further restricts the way in which an existing parameter can be bound.

These concepts are illustrated by the following conditional interpretation of the sentence "The jet failed":

[[ "The jet failed" ] ] =

$$\langle \text{fail}(x), \{\text{bind}(x, \text{def}, \text{jet})\} \rangle \quad (1)$$

The first element of the interpretation is the sense *fail(x)*, while the second is the set of assumptions containing a single assumption whose informal reading is that *x* should be bound to something of the sort *jet* according to the constraints of definite reference.

## 3 The Interpretation Process

The process of semantic and pragmatic interpretation computes complete interpretations of sentences from least-commitment parse trees. Two types of rules govern the interpretation process: *semantic-interpretation rules* and *pragmatic-discharge rules*.

Semantic-interpretation rules specify the conditional interpretation of a phrase in terms of the conditional interpretations of its constituents. Compositionality is enforced by making semantic-interpretation rules sensitive only to the syntactic types of a phrase and its constituents, as well as to the types of assumptions in the conditional interpretations associated with the constituents; semantic-interpretation rules are *not* sensitive to the senses of the constituents.

Pragmatic-discharge rules change the conditional interpretation of a phrase by specifying how assumptions in the conditional interpretation may be eliminated with respect to the context of utterance. For example, one discharge rule applies to assumptions constraining a parameter to be bound as a definite reference. This rule allows an assumption of the form  $\text{bind}(v, \text{def}, T)$  to be discharged, provided that there is a unique contextually available entity of sort *T*. The effect of applying the definite discharge rule to an interpretation  $\langle S, A \rangle$  is twofold: the bind assumption operated upon is removed from the set of assumptions *A*; the sense *S* is changed to reflect the binding. For instance, if the rule were applied to the interpretation in (1), and if the context of utterance *C* contained a unique available entity *j* of sort *jet*, the resulting interpretation would be

$$\langle \text{fail}(j), \phi \rangle \quad (2)$$

As we shall see in the next section, assumption discharge will in general not only make use of but also change the discourse context. Therefore, discharge rules should be viewed as four-place relations. For example, the following would be an instance of the

discharge relation:

$$\text{discharge}(C, \{fail(x), \{bind(x, def, jet)\}\}, \\ \{fail(j), \phi\}, C'),$$

where  $C$  is the discourse context before the assumption is discharged, while  $C'$  is the resulting discourse context.

Semantic-interpretation rules are obligatory in that some semantic-interpretation rule associated with a given syntactic rule must be applied to any phrase analyzed by the syntactic rule. In contrast, the application of pragmatic-discharge rules is optional, although discharging a particular assumption too early or too late may lead to a dead end in the interpretation process. Applying the same discharge rule at different points in the interpretation process for some utterance may lead to alternative interpretations, as we shall illustrate with the examples in Sections 6 and 7.

Given a sentence and its syntactic analysis, the interpretation process applies semantic-interpretation and pragmatic-discharge rules, according to their applicability conditions, to construct the derivation of a complete interpretation of the sentence. In *Candide*, this process resembles a syntax-directed translation system [1]. Interpretation starts at the root node of the analysis tree. For each node of the tree, the interpretation process selects an appropriate semantic-interpretation rule and calls itself recursively for each of the node's daughters. Interpretations are constructed on return from the recursion, and pragmatic-discharge rules are optionally applied in a discharge cycle that follows each application to a node of a semantic-interpretation rule.

Lexical ambiguity, multiple semantic-interpretation rules for a given syntactic construction, optional application of discharge rules, and alternative ways of discharging a given assumption, are all sources of nondeterminism in the interpretation process, which need to be somehow controlled. In *Candide*, we adopted four simple control tactics: overall depth-first search, early discharge of assumptions, breadth-first search for alternative bindings of a discharged parameter, and bounds on assumption percolation wherever it can be shown that an assumption would not be dischargeable outside a certain syntactic domain. For lack of space, a fuller discussion of these heuristics will be conducted elsewhere [24].

## 4 The Discourse Context

Pragmatic-discharge rules need access to a discourse context that encodes information about relevant world knowledge and the discourse history. Although our framework for semantic and pragmatic interpretation can accommodate alternative representations of the discourse context, the specific discharge rules we have written and incorporated into the *Candide* system rely on a particular representation comprising four parts: *immediate context*, *local context*, *global context*, and a *knowledge base*.

During the analysis of a sentence, the immediate context contains detailed information about the entities referred to in that sentence; it is used primarily for resolving intrasentential anaphora. The local context generally contains detailed information about the immediately preceding sentence,<sup>4</sup> while the global context includes somewhat less detailed information about entities referred to throughout longer stretches of the discourse. We use the local context primarily for pronoun resolution, following the theory of centering introduced by Grosz *et al.*[12]. The global context is employed primarily for the resolution of definite anaphora, and is structured as a stack to make use of the theory of focusing. Each element of the global-context stack is itself a list of entries containing information about the entities referred to in a discourse segment [13]. We refer to the top element of the global context as the *intermediate context*.

Individual discharge rules used in processing a sentence can extend the immediate context for that sentence. For instance, the rule mentioned earlier that binds a parameter as a definite reference adds to the immediate context an entry for the entity to which the parameter is bound. When the assumption in (1) is discharged, resulting in the interpretation in (2), an entry for  $j$  must be added to the immediate context. The entry will include the sort of  $j$  (*jet*) and the surface position of that phrase in the sentence (subject).

The discourse context must also be updated after each sentence has been processed. In the simplest case, the update will be quite straightforward, as illustrated in Figure 1: the current immediate context will become the new local context, while a subset of the information encoded in the immediate context will be also added to the intermediate context (the topmost element of the global-context stack). The immediate context will

<sup>4</sup>This will not be true when a "pop" of the global context has just occurred [13].

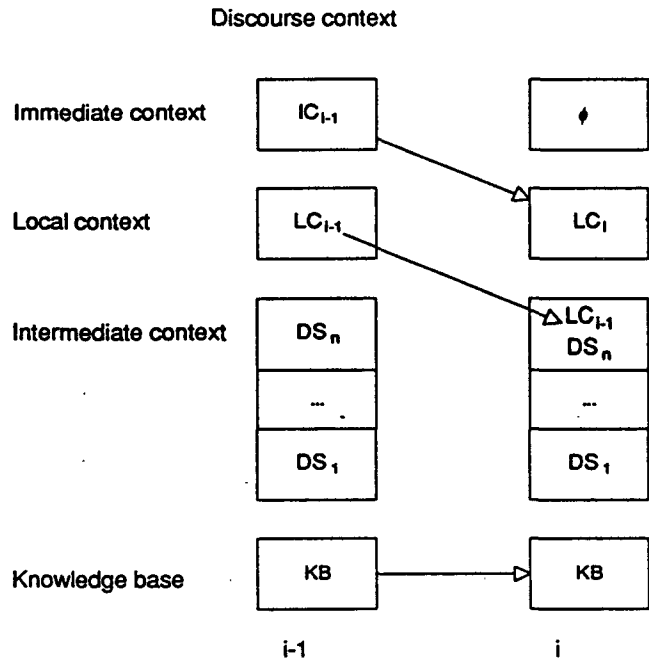


Figure 1: Updating the Discourse Context

be cleared in preparation for the next sentence. For the moment, we shall assume that the knowledge base is static, although it will ultimately have to be reorganizable dynamically so as to reflect a language user's current perspective.

In fact, the update function can be rather more complex. For example, if the current utterance is recognized to be the start of a subordinate discourse segment, a new, empty element can be pushed onto the global-context stack after the local context has been merged into the previous top element. We shall discuss the discourse-context update function further elsewhere [24].

## 5 A Simple Discourse

The following simple discourse will provide our first illustration of the interpretation mechanism and, in particular, the treatment of reference and coreference:

- The jet failed. (3)  
Close the manifold.

In the subsequent sections, we shall turn to more complex examples that provide further insight into the way in which pragmatic processes can interact

with one another affecting syntactic and semantic decisions.

The three semantic-interpretation rules given in Figure 2 are needed in the example. Recall that the interpretation process is driven by semantic-interpretation rules, which apply compositionally to phrases. Each such rule has three parts: an *applicability condition* (AC), a set of *selection functions* (SF), and a *conditional-interpretation function* (CIF). The applicability condition specifies the syntactic type of phrase to which the semantic interpretation rule applies; it is stated in terms of a predicate on trees.<sup>5</sup> The selection functions specify how to access the constituents of the phrase to which the rule is to be applied. Finally, the conditional-interpretation function defines the conditional interpretation of the phrase as a function of the conditional interpretations of its constituents. A conditional-interpretation function will often depend separately on the sense and assumptions of a conditional interpretation  $I$ , for which we use the notations  $I_S$  and  $I_A$ , respectively.

Figure 3 shows an annotated tree<sup>6</sup> represent-

<sup>5</sup>The meanings of the predicates on trees used in this paper should be clear from their names.

<sup>6</sup>Our analysis trees are closer to the functional struc-

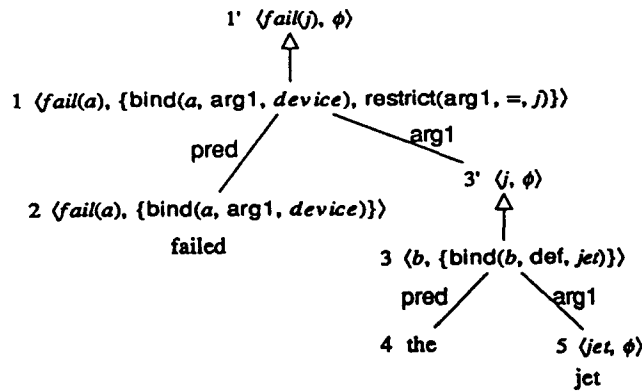


Figure 3: Interpretation of "The jet failed"

[iv-clause]:

- AC: intrans-verb-clause( $T$ )
- SF:  $pred(T) = V, arg1(T) = A$
- CIF:  $[T] = \langle [V]_S, [V]_A \cup [A]_A \cup \{restrict(arg1, =, [A]_S)\} \rangle$

[def-np]:

- AC: def-np( $T$ )
- SF:  $arg1(T) = N$
- CIF:  $[T] = \langle x, [N]_A \cup \{bind(x, def, [N]_S)\} \rangle$

[lex]:

- AC: lex-item( $T$ )
- SF:  $wordstem(T) = W$
- CIF:  $[T] = I_W$

Figure 2: Semantic-Interpretation Rules I

ing the derivation of a complete interpretation of the first sentence in (3). Conditional interpretations of constituents of the complete sentence are shown above the root nodes of the corresponding subtrees.

Semantic-interpretation rule [lex] applies to lexical subtrees (Nodes 2 and 5 in Figure 3<sup>7</sup>) associ-

tures of lexical-functional grammar [4] than to the usual surface constituent structures. The sample analyses have been extremely simplified for expository reasons; terminal nodes, in particular, appear in the trees simply as the corresponding word, but their actual representation, as required by interpretation rule [lex], has two branches: wordstem for the actual root form of the terminal, cat for its syntactic category. Finally, tree nodes relevant to the discussion are numbered for ease of reference.

<sup>7</sup>Node 4 is also lexical, but definite determiners contribute only to the interpretation of their mother noun

ating with each wordstem  $W$  conditional interpretations  $I_W$  according to the lexicon. The lexical entries relevant to the current discussion are:

- $I_{jet} = \langle jet, \phi \rangle$
- $I_{fail} = \langle fail(x), \{bind(x, arg1, device)\} \rangle$

In the conditional interpretation of a common noun, the sense is always a sort term. The assumption set may be empty, as it is for "jet" above, but for a relational noun it will contain bind assumptions for the relation's arguments, binding parameters occurring in the sort term.

The lexical entries for verbs and the structural rules that combine a verb with its subject and complements must refer, through assumptions, to the grammatical functions that provide the arguments of the predicates that represent the senses of verbs (roughly the *governable* grammatical functions of lexical-functional grammar [4,27]). Since we are not defending any particular theory of grammar in this paper, we shall skirt a theoretical and terminological minefield by naming the grammatical functions relevant to our purposes  $arg_i$  for  $i = 1, \dots, n$ , and calling them simply "arguments." Arguments are used as edge labels in our analyses, as well as in bind and restrict assumptions, and their intended interpretation should be clear from the examples we are discussing.

The encoding of selectional restrictions is illustrated here in the conditional interpretation of the verb "fail," which is  $fail(x)$ , under the assumption that  $x$  must be bound as first argument of the verb to something of the sort *device*. This inter-

phrase, by rule [def-np], rather than being given a separate interpretation.

pretation effectively encodes the information that things that fail are devices.<sup>8</sup>

Because the local tree rooted at Node 3 represents a definite noun phrase, rule [def-np] applies to it in a straightforward fashion, yielding the conditional interpretation

$$\langle b, \{ \text{bind}(b, \text{def}, \text{jet}) \} \rangle \quad (4)$$

That is, “the jet” is interpreted as  $b$  under the assumption that  $b$  can be bound in accordance with the constraints of definite reference to an entity of sort *jet*.

As mentioned earlier, a pragmatic-discharge rule *may* be used whenever it is applicable to some conditional interpretation in context. In the current example, the rule for discharging the bind assumption is applicable to the conditional interpretation in (4), and it is actually used in the derivation to determine a referent for the definite noun phrase.

The process of resolving a definite reference is of course quite complex [5,11,28,29], and the rule that discharges assumptions to bind a parameter as a definite reference must reflect this complexity. For the moment, let us assume that there is only one entity of the correct sort available for definite reference (perhaps introduced in a preceding portion of the discourse): the jet identified as  $j$ . The pragmatic discharge rule can thus bind the parameter  $b$  to  $j$ , extend the immediate context accordingly, and delete the bind assumption from the list of assumptions in the current conditional interpretation. The resulting conditional interpretation of the string “the jet” is  $\langle j, \phi \rangle$ , shown in Figure 3 above Node 3’.

Finally, consider the interpretation of the whole sentence. Rule [v-clause] applies to the parse tree for the sentence, specifying that its sense is the sense of the predicate (pred) constituent, namely *fail*( $a$ ), and that its set of assumptions is the union of (i) the assumptions from its predicate constituent, (ii) the assumptions from its argument (arg1) constituent, and (iii) the new assumption  $\text{restrict}(\text{arg1}, =, j)$ , where  $j$  is the sense of the argument constituent. The restrict assumption, which arises from the sentence’s syntactic form, applies to whatever parameter is to be bound as the first argument of the sense of the sentence—in

<sup>8</sup>The conditional interpretation shown above Node 2 in the figure has a new parameter,  $a$ , substituted for the variable  $x$  of the lexical entry because parameters introduced through bind assumptions in distinct applications of semantic interpretation-rules in a derivation must be themselves distinct.

this case,  $a$ , as specified by the bind assumption inherited from the predicate constituent. The restrict assumption further constrains the binding of this parameter by requiring that it be equated with the entity  $j$ .

The interpretation process is completed after the two remaining assumptions are discharged, as indicated at the top of Figure 3.<sup>9</sup> They can be discharged successfully in parallel: binding  $a$  to  $j$  is legitimate because  $j$  is a jet, and *jet* is a subsort of *device*. Before the next sentence is processed the discourse context needs to be updated, as described earlier.

The second sentence of our example is “Close the manifold”; we shall be concerned primarily with the way in which the reference resolution problem is handled. The conditional interpretation for the definite noun phrase “the manifold” is

$$\langle c, \{ \text{bind}(c, \text{def}, \text{manifold}) \} \rangle \quad (5)$$

Discharging the bind assumption here requires the use not only of world knowledge — namely, that each jet is attached to one and only one manifold — but also of knowledge of the discourse history — namely that there is a single salient jet in context, the one identified as  $j$ . The latter information can be derived from the discourse context, while the former must be encoded in the knowledge base. This information is sufficient to resolve the reference in the sentence under consideration: “the manifold” refers to the manifold that is attached to  $j$ . Hence the interpretation we derive from (5) is

$$\langle m, \phi \rangle, \quad (6)$$

where  $m$  is the unique manifold attached to jet  $j$ . For use in constraining subsequent reference, the discourse context must be updated with the information that  $m$  has the restricted sort : *manifold* |  $\lambda x. \text{attached-to}(x, j)$ , where  $s|P$  is the subsort of  $s$  whose elements satisfy property  $P$ .

<sup>9</sup>In the *Candide* system as it currently exists, a bind assumption encoding a selectional restriction and a restrict assumption encoding the filler of an argument must be discharged as soon as the latter has been introduced; otherwise an erroneous interpretation might be derived if the restrict assumption is mistakenly applied at a higher clause node. A better scheme would encode sufficient information in these restrict assumptions to ensure that they could apply only to the appropriate clause.

## 6 Quantifier Scope

We shall now turn to the kind of interactions in pragmatic processing that challenge compositional systems. In this section we shall discuss an example of quantifier scope ambiguity; following that, we shall give an example of our analysis of donkey sentences, involving interactions between quantifier scoping and reference resolution.

The following sentence illustrates the quantifier scoping problem in its simplest form:

Every driver controls a jet. (7)

This sentence might be given either a wide-scope existential ( $\exists V$ ) interpretation, in which all the drivers control the same jet, or a narrow-scope existential ( $\forall \exists$ ) interpretation, in which each driver controls its own, possibly different, jet.

[tv-clause]:

AC: trans-verb-clause( $T$ )  
 SF:  $\text{pred}(T) = V, \text{arg1}(T) = A_1, \text{arg2} = A_2$   
 CIF:  $\llbracket T \rrbracket = \langle \llbracket V \rrbracket_S,$   
 $\llbracket V \rrbracket_A \cup \llbracket A_1 \rrbracket_A \cup \llbracket A_2 \rrbracket_A \cup$   
 $\{\text{restrict}(\text{arg1}, =, \llbracket A_1 \rrbracket_S),$   
 $\text{restrict}(\text{arg2}, =, \llbracket A_2 \rrbracket_S)\}$

[gen-quant]:

AC: gen-quant( $T$ )  
 SF:  $\text{pred}(T) = Q, \text{arg1}(T) = N$   
 CIF:  $\llbracket T \rrbracket = \langle x, \llbracket N \rrbracket_A \cup \{\text{bind}(x, \llbracket Q \rrbracket_S, \llbracket N \rrbracket_S)\}\rangle$

[indef-np]:

AC: indef-np( $T$ )  
 SF:  $\text{arg1}(T) = N$   
 CIF:  $\llbracket T \rrbracket = \langle x, \llbracket N \rrbracket_A \cup \{\text{bind}(x, \text{indef}, \llbracket N \rrbracket_S)\}\rangle$

Figure 4: Semantic-Interpretation Rules II

Interpreting (7) requires additional rules of semantic interpretation, shown in Figure 4, and the lexical entry

$$I_{\text{control}} = \langle \text{controls}(x, y), \{\text{bind}(x, \text{arg1}, \text{device}), \text{bind}(y, \text{arg2}, \text{device})\} \rangle$$

Derivations of the  $\exists V$  and the  $\forall \exists$  interpretations are shown in Figures 5 and 6, respectively.

In both derivations, the general noun phrase “every driver” is interpreted at Node 2 by rule [gen-quant] and the indefinite noun phrase “a jet”

is interpreted at Node 1 by rule [indef-np]. However, the two derivations differ as to where the indefinite-reference assumption is discharged. In Figure 5 the assumption is discharged immediately after its introduction. The resulting sense is a new entity  $j$  of sort *jet*. The same  $\exists V$  reading could also be derived by allowing the indefinite-reference assumption to percolate up to Node 3, but then discharging it before the generalized quantifier assumption. In either case, the immediate context is updated at the time of the discharge with an entry for the new entity  $j$ .

Somewhat more interesting is the derivation of the  $\forall \exists$  reading, shown in Figure 6. The indefinite-reference assumption is allowed to percolate to Node 3, where the generalized-quantifier assumption is discharged. This discharge applies a quantifier to its scope, but it also selects some subset of the outstanding indefinite-reference assumptions in the current conditional interpretation and discharges them, by existential quantification of the respective parameters, within the scope of the generalized quantifier. In our example, the rule converts the conditional interpretation

$$\langle \text{controls}(a, b), \{\text{bind}(a, \forall, \text{driver}), \text{bind}(b, \text{indef}, \text{jet})\} \rangle$$

into the completed interpretation

$$\langle \forall a: \text{driver} \exists b: \text{jet} \text{ controls}(a, b), \phi \rangle$$

## 7 A Donkey Sentence

We can now discuss the more complicated interactions between assumptions occurring in donkey sentences. Our example will be the sentence:

Every driver controlling a jet closes it. (8)

Clearly, this sentence has an interpretation in which, for every driver controlling a jet, the driver closes the jet. However, it is difficult to see how this interpretation can be derived compositionally. The well-recognized problem is that, in the intended reading, the indefinite noun phrase “a jet” has narrower scope than the determiner “every,” forcing its interpretation to be part of the sort term translating the nominal “driver controlling a jet.” But this means that the interpretation of the pronoun “it” will be outside the scope of the indefinite “a jet.”

Our solution to the problem of interpreting donkey sentences involves two new mechanisms: *capture* rules that allow the quantifier in a general

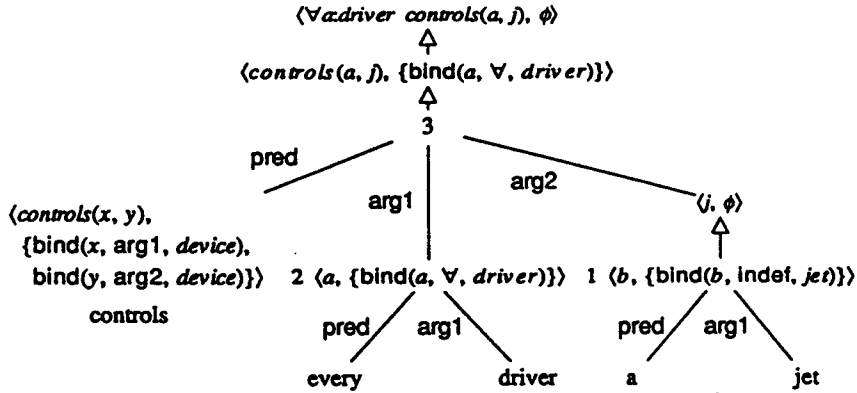


Figure 5:  $\exists\forall$  Interpretation

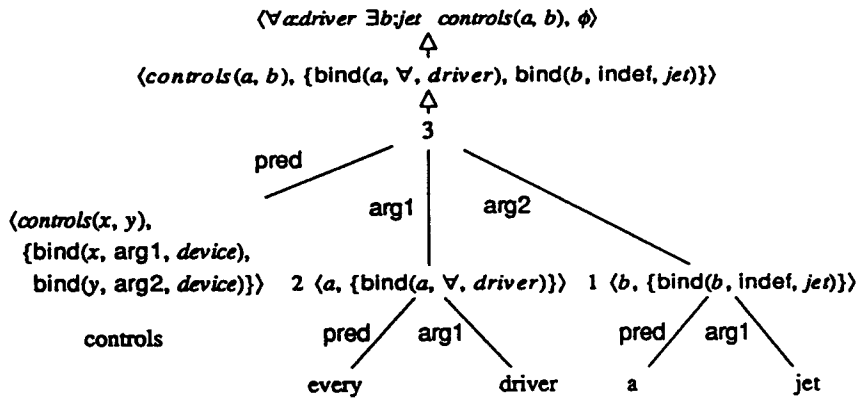


Figure 6:  $\forall\exists$  Interpretation

noun phrase to discharge in a particular way bind assumptions derived from singular noun phrases occurring in the general noun phrase, and a *pronoun resolution* rule that discharges a pronoun-introduced bind assumption by replacing the assumption's parameter with the parameter bound by the assumption for a possible antecedent of the pronoun.

Figure 7 shows a simplified derivation of an interpretation of sentence (8), with some of the less interesting assumptions discharged immediately after their introduction rather than being listed explicitly. Before discussing the main points of this example we need to explain our somewhat nonstandard representation of [reduced] relative clauses, as in the compound nominal “driver controlling a jet” (Node 2). A relative clause is represented as a main clause but has one of its argu-

ment positions filled by a nominal (the head noun modified by the relative clause) instead of a noun phrase. The discharge rule discussed in Section 5 that combines a verb argument with its filler then has two versions: one in which the filler sense is an entity, already described, and one in which the filler sense is a sort. In the latter case, the rule produces an interpretation whose sense is the filler sort restricted by the sense of the clause. In the foregoing example, the sort-filler discharge rule is applied to the interpretation

$\langle \text{controls}(x, b),$   
 $\{\text{bind}(x, \text{arg1}, \text{device}),$   
 $\text{restrict}(\text{arg1}, =, \text{driver}),$   
 $\text{bind}(b, \text{indef}, \text{jet})\} \rangle$



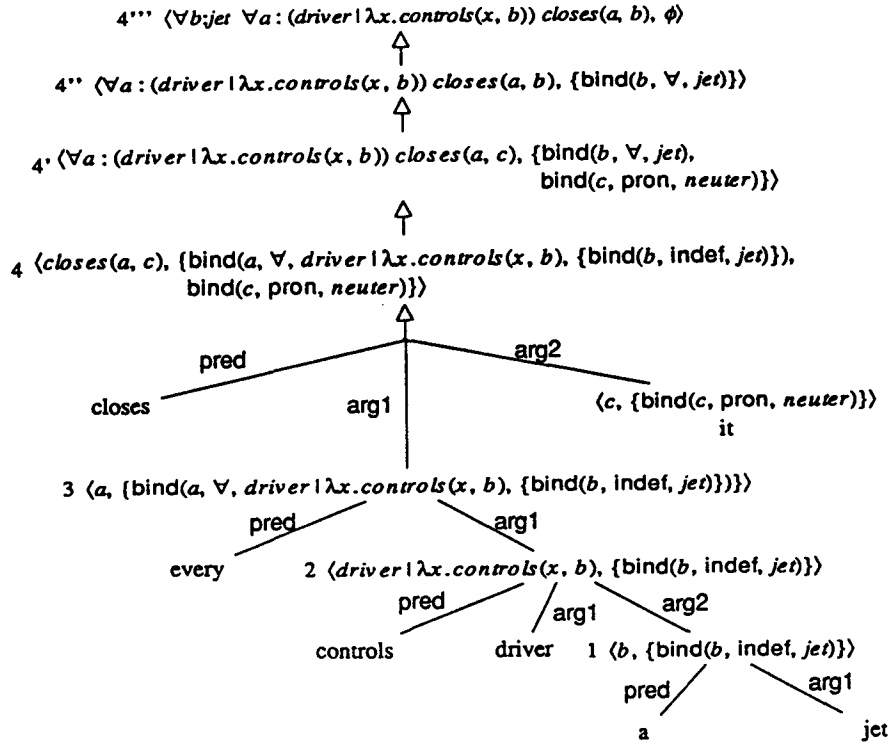


Figure 7: Interpretation of a Donkey Sentence

to produce the restricted sort

$$\langle driver \upharpoonright \lambda x.controls(x, b), \{bind(b, indef, jet)\} \rangle$$

After these preliminaries, we can go on to the main point of the example. The first observation to make is that the sentence has an alternative (albeit unlikely) interpretation in which “a jet” is taken to refer to a specific jet that every driver controls. This interpretation would be derived by discharging the corresponding indefinite-reference assumption at Nodes 1 or 2 in the derivation.<sup>10</sup> We shall assume that this is not done, and that the indefinite-reference assumption is therefore available at Node 3.

So far bind assumptions have been given as triples of a parameter, a binding criterion (derived from a determiner), and a sort restriction for the parameter. In fact, a fourth component of *depen-*

<sup>10</sup>A third interpretation is also possible, in which “a jet” is interpreted as a narrow-scope (nonreferential) existential, and “it” is interpreted as having an extrasentential referent. Limitations in Candide’s handling of nonreferential indefinites preclude this reading, but a somewhat different rule system will generate all three readings correctly [23].

*dependencies* is in general required, a set of other assumptions that the given assumption may depend on.<sup>11</sup> An assumption  $\alpha$  (the *dependent* assumption) *depends* on another assumption  $\beta$  (the *independent* assumption) whenever the parameter for  $\beta$  occurs in the sort constraint of  $\alpha$ . For the language fragment under discussion,  $\alpha$  would be the bind assumption for a complex noun phrase and  $\beta$  the bind assumption for a noun phrase within a prepositional phrase or relative clause in the complex noun phrase. For correct binding of quantified parameters, semantic interpretation and discharge rules must maintain the invariant that assumptions on which a given assumption depends can occur only in its set of dependencies. Consequently, whenever a dependent assumption  $\alpha$  is introduced any other assumption on which it depends must be moved into  $\alpha$ ’s dependencies, thereby becoming inaccessible to discharge rules. If  $\alpha$  is later discharged, the assumptions in its set of dependencies again become accessible to dis-

<sup>11</sup>In the examples so far this set has been empty and therefore omitted for the sake of clarity.

charge rules. Semantic interpretation must be modified to fit this analysis. For instance, rule [gen-quant], given earlier, should be instead

[gen-quant]:  
 AC: gen-quant( $T$ )  
 SF:  $\text{pred}(T) = Q, \text{arg1}(T) = N$   
 CIF:  $[T] = \langle x, \{\text{bind}(x, [Q]_S, [N]_S, [N]_A)\} \rangle$

In Figure 7, this rule is applied at Node 3.

Capture may occur whenever a generalized-quantifier assumption with a nonempty set of dependencies  $D$  is discharged. Any indefinite assumption in  $D$  may be captured by turning it into a universal-quantification assumption and putting it into the set of assumptions for the new conditional interpretation. In our example, the indefinite assumption for “a jet” is captured in the discharge of the universal assumption for “every driver...”, from Node 4 to Node 4' in the derivation. The resulting assumption is now universal. If this assumption were discharged immediately, there would be no way of discharging the pronoun assumption as an intrasentential anaphoric reference. Instead the pronoun resolution rule is applied to discharge the pronoun assumption, causing identification of the pronoun parameter  $c$  with the jet parameter  $b$ . The resulting conditional interpretation is 4''. Finally, the remaining assumption can be discharged by quantification leading to the complete interpretation at Node 4'''.

The example shows how assumptions allow interactions between reference and quantification to be left unresolved until all the necessary information becomes available. Early discharge of the assumption for “a driver” blocks the desired interpretation for the pronoun “it”; capture makes available the attributive use of “a driver” at an appropriate point for its identification with the direct object of “close.”

## 8 Related Research

Strictly compositional approaches to semantic interpretation, such as Montague grammar [19], have so far proved inadequate for dealing with interactions between meaning and context; reasons for this are noted in Section 1. Our approach can be thought of as a generalization of the compositional mechanism of Cooper storage [6], or of its computational analogue developed by Woods [30]. Alternative approaches that attempt to address these interactions include discourse-representation theory (DRT) [14,18] and Barwise's partial-valuation approach [2].

In DRT, the interpretation of a sentence is derived in a compositional manner from an intermediate representation called a discourse-representation structure (DRS). However, the rules that have been developed for constructing DRSs are not themselves compositional. According to the DRS-construction rules presented by Kamp [18], the DRS for a phrase is found only as a by-product of finding the DRS for the embedded discourse. In particular, DRS-construction rules apply only after the relative scope of noun phrases and anaphoric bindings have been determined. It is conceivable that our notion of conditional interpretation might be reexpressible in DRT terms, leading to a compositional system for DRS construction.

Barwise [2] uses the notion of partial valuation, that is, partial assignments of values to variables, to analyze the sorts of interactions exemplified by the donkey sentences. Similar comments apply to Webber's work [29]. In addition, none of the aforementioned accounts has been concerned with as wide a range of phenomena as is currently handled in *Candide*.<sup>12</sup> One of the motivations for our work has been to see how Barwise's direct-interpretation approach could be turned into a two-stage one in which phrases are first “compiled” into conditional interpretations, which are then “executed” by applying pragmatic-discharge rules.

Finally, several other computational systems developed recently are concerned with interactions between context and meaning, especially Pundit [8,21] and Tacitus [17,16]. Both these systems have emphasized solutions to such difficult pragmatic problems as reference resolution. In particular, the Pundit project has made notable progress on the question of resolving missing arguments, while the Tacitus group has done the same for questions involving the determination of implicit relations. In *Candide*, solutions to such pragmatic problems should be encoded in the procedures that discharge assumptions; in future versions of the system the discharge procedures might be improved applying some of the techniques developed in this other work. What neither Pundit nor Tacitus has been concerned with is the question of how to build interpretations composition-

<sup>12</sup>To date, we have included capabilities for processing reference and coreference (definite and indefinite noun phrases, pronouns, possessives, and proper nouns), quantifier scope, compound nominals, prepositional-phrase attachment, and certain types of underspecified relations (e.g., main-verb “have”). We shall report on these mechanisms elsewhere [24].

ally. Both systems first build partial interpretations of sentences, and then attempt to solve a collection of associated pragmatic problems. Pundit does the latter in an overly constrained way, with the result that it cannot handle systematically the sort of interactions exemplified by the donkey sentences. Tacitus, on the other hand, casts all the pragmatic problems as theorems to be proved; the result is an underconstrained control strategy. We believe that the generally compositional approach developed in Candide enables us to avoid both these extremes.

## 9 Further Work

We have developed a mechanism of semantic and pragmatic interpretation that relaxes the constraints of compositional semantics just enough to allow pragmatic information to play its necessary role in the derivation of sentence interpretations. Central to the mechanism are conditional interpretations, which allow us to separate constraints on interpretation that depend only on syntactic structure, represented by the sense component of the conditional interpretation, from those that depend on pragmatic choices, represented by the assumption component. The interpretation process is carried out by a combination of semantic-interpretation rules, which build conditional interpretations of phrases on the basis of lexical and syntactic information, and pragmatic-discharge rules, which satisfy assumptions on the basis of discourse and domain information. While the system we have implemented deals with a variety of semantic and pragmatic phenomena, of which only a few were discussed in this paper, it can only be seen as a first limited instantiation of a system architecture that requires much further work. We shall mention now a few of the directions that might be pursued in developing the architecture further.

At the most theoretical level, it is interesting to note the formal similarity of our interpretation rules to rules in "deductive" models of programming language semantics [25]. It is also interesting to consider the connection between conditional interpretations and the relational theory of meaning from situation semantics [3]. These two similarities might be fruitful in developing a semantic justification for our formal interpretation rules in terms of constraints on interpretation relations.

The applicability of discharge rules depends in many cases on the compatibility of expected and

supplied sorts for relation arguments. In general, these sorts may be parameterized by assumption parameters, and some semantic interpretation problems not considered here suggest that higher-order parameterized types, instead of first-order sorts, may be needed. A suitable notion of type subsumption for such higher-order parameterized types [15] would be useful. More generally, the whole architecture would benefit from a semantically grounded treatment of parameters and parameterized objects.

Other pragmatic processes associated with discharge rules, such as those for reference resolution, also must be able to reason with parameterized objects—for example in checking the uniqueness of a dependent object relative to arbitrary parameter assignments. Ultimately, the proper treatment of singular noun phrases in context will require a closer connection between assumptions and [parameterized] fragments of the discourse context.

## References

- [1] A. V. Aho, R. Sethi, and J. D. Ullman. *Compilers: Principles, Techniques and Tools*. Addison-Wesley, Reading, Massachusetts, 1985.
- [2] J. Barwise. Noun phrases, generalized quantifiers and anaphora. In P. Gärdenfors, editor, *Generalized Quantifiers: Linguistic and Logical Approaches*, pages 1–29, D. Reidel, Dordrecht, Netherlands, 1987.
- [3] J. Barwise and J. Perry. *Situations and Attitudes*. MIT Press, Cambridge, Massachusetts, 1983.
- [4] J. Bresnan and R. Kaplan. Lexical-functional grammar: a formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281, MIT Press, Cambridge, Massachusetts, 1982.
- [5] D. Carter. *Interpreting Anaphors in Natural Language Texts*. Ellis Horwood, Chichester, England, 1987.
- [6] R. Cooper. *Quantification and Syntactic Theory*. D. Reidel, Dordrecht, Netherlands, 1983.
- [7] S. Crain and M. Steedman. On not being led up the garden path: the use of context by the psychological syntax processor. In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Parsing: Psychological, Computational, and Theoretical Perspectives*, pages 320–358, Cambridge University Press, Cambridge, England, 1985.
- [8] D. A. Dahl, M. S. Palmer, and R. J. Passonneau. Nominalizations in Pundit. In *Proceedings of the*

- 24th Annual Meeting of the Association for Computational Linguistics, pages 131-139, Stanford, California, 1987.
- [9] P. Geach. *Reference and Generality*. Cornell University Press, Ithaca, New York, 1962.
- [10] M. P. Georgeff and A. L. Lansky. Procedural knowledge. *Proceedings of the IEEE, Special Issue on Knowledge Representation*, 1383-1398, 1986.
- [11] B. J. Grosz. *The Representation and Use of Focus in Dialogue Understanding*. Technical Report 151, SRI International, Menlo Park, Ca., 1977.
- [12] B. J. Grosz, A. K. Joshi, and S. Weinstein. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 44-50, Cambridge, Massachusetts, 1983.
- [13] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175-204, 1986.
- [14] F. Guenther, H. Lehmann, and W. Schonfeld. A theory for the representation of knowledge. *IBM Journal of Research and Development*, 30(1):39-56, January 1986.
- [15] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. In *Proceedings of the Second Symposium on Logic in Computer Science*, Cornell University, IEEE, Ithaca, New York, 1987.
- [16] J. R. Hobbs. Interpretation as abduction. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, New York, 1988.
- [17] J. R. Hobbs. Overview of the TACITUS project. *Computational Linguistics*, 12(3), 1986.
- [18] H. Kamp. A theory of truth and semantic interpretation. In J. A. G. Groenendijk, T. M. V. Janssen, and M. B. J. Stokhof, editors, *Formal Methods in the Study of Language*, pages 277-322, Mathematisch Centrum, Amsterdam, Netherlands, 1981.
- [19] R. Montague. The proper treatment of quantification in ordinary English. In R. H. Thomason, editor, *Formal Philosophy*, Yale University Press, 1973.
- [20] R. C. Moore. Problems in logical form. In *Proceedings of the 19th Meeting of the Association for Computational Linguistics*, pages 117-124, Stanford, California, 1981.
- [21] M. S. Palmer, D. A. Dahl, R. J. Schiffman, L. Hirschman, M. Linebarger, and J. Dowding. Recovering implicit information. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, pages 10-19, New York, 1986.
- [22] F. C. Pereira. *Logic for Natural Language Analysis*. Technical Report 275, SRI International, Menlo Park, California, 1983.
- [23] F. C. Pereira. Towards a deductive theory of sentence interpretation. Unpublished manuscript.
- [24] F. C. Pereira and M. E. Pollack. A compositional, declarative system for semantic and pragmatic interpretation. In preparation.
- [25] G. D. Plotkin. *A Structural Approach to Operational Semantics*. Lecture notes DAIMI FN-19, Aarhus University, Aarhus, Denmark, September 1981.
- [26] L. K. Schubert and F. J. Pelletier. From English to logic: context-free computation of 'conventional' logical translation. *Computational Linguistics*, 8(1):26-44, 1982.
- [27] P. Sells. *Lectures on Contemporary Syntactic Theories*. Volume 3 of *CSLI Lecture Notes*, Center for the Study of Language and Information, Stanford University, Stanford, California, 1985. Distributed by University of Chicago Press.
- [28] C. L. Sidner. Focusing in the comprehension of definite anaphora. In *Computational Models of Discourse*, MIT Press, Cambridge, Massachusetts, 1983.
- [29] B. L. Webber. So what can we talk about now? In *Computational Models of Discourse*, MIT Press, Cambridge, Ma., 1983.
- [30] W. A. Woods. Semantics and quantification in natural language question answering. In M. Yovits, editor, *Advances in Computers*, Vol. 17, pages 2-64, Academic Press, New York, New York, 1978.