

# From English to Logic: Context-Free Computation of 'Conventional' Logical Translation<sup>1</sup>

Lenhart K. Schubert

Department of Computing Science  
University of Alberta

Francis Jeffry Pelletier

Department of Philosophy  
University of Alberta  
Edmonton, Canada T6G 2H1

We describe an approach to parsing and logical translation that was inspired by Gazdar's work on context-free grammar for English. Each grammar rule consists of a syntactic part that specifies an acceptable fragment of a parse tree, and a semantic part that specifies how the logical formulas corresponding to the constituents of the fragment are to be combined to yield the formula for the fragment. However, we have sought to reformulate Gazdar's semantic rules so as to obtain more or less 'conventional' logical translations of English sentences, avoiding the interpretation of NPs as property sets and the use of intensional functors other than certain propositional operators. The reformulated semantic rules often turn out to be slightly simpler than Gazdar's. Moreover, by using a semantically ambiguous logical syntax for the preliminary translations, we can account for quantifier and coordinator scope ambiguities in syntactically unambiguous sentences without recourse to multiple semantic rules, and are able to separate the disambiguation process from the operation of the parser-translator. We have implemented simple recursive descent and left-corner parsers to demonstrate the practicality of our approach.

## 1. Introduction

Our ultimate objective is the design of a natural language understanding system whose syntactic, semantic and pragmatic capabilities are encoded in an easily comprehensible and extensible form. In addition, these encodings should be capable of supporting efficient algorithms for parsing and comprehension.

In our view, the achievement of the former objective calls for a careful *structural* separation of the subsystems that specify possible constituent structure (syntax), possible mappings from constituent structure to underlying logical form (part of semantics), and possible mappings from logical form to deeper, unambiguous representations as a function of discourse context and world knowledge (part of pragmatics and

inference). This sort of view is now widely held, as evidenced by a recent panel discussion on parsing issues (Robinson 1981). In the words of one of the panelists,

"I take it to be uncontroversial that, other things being equal, a homogenized system is less preferable on both practical and scientific grounds than one that naturally decomposes. Practically, such a system is easier to build and maintain, since the parts can be designed, developed, and understood to a certain extent in isolation... Scientifically, a decomposable system is much more likely to provide insight into the process of natural language comprehension, whether by machines or people." (Kaplan 1981)

The panelists also emphasized that structural decomposition by no means precludes interleaving or paral-

<sup>1</sup> Submitted August 1981; revised July 1982.

lelism of the *processes* that draw on the various kinds of linguistic and non-linguistic knowledge.

Note that we are making a distinction between the logical form that corresponds directly to surface structure on the one hand, and an unambiguous deeper representation on the other. Indeed, at the level of logical form our theory of logical translation admits ambiguities in all of the formal building blocks (terms, functions, predicates, connectives, and quantifiers), as well as in the scopes of quantifiers and coordinators. For example, logical-form translations may contain terms such as *Mary2* and *<the1 (little2 girl3)>*, ambiguous between various referents (e.g., *MARY5* and *MARY17*), and quasi-predicates such as *has3*, *good2*, *cold5*, and *recovers1*, ambiguous between various proper predicates (e.g., *has3: OWNS1*, *AFFLICTED-WITH, ...*; *good2: VIRTUOUS, GOOD-TASTING, ...*; *cold5: COLD1, EMOTIONLESS, ...*; and *recovers1: RE-COVERS1, REGAINS, ...*). In other words, we do not regard the logical form of a sentence as fully determining its meaning – not even its ‘literal’ meaning; rather, its meaning is determined by its logical form along with the context of its utterance. Thus “She is becoming cold” might convey on one occasion that Lady Godiva is beginning to feel cold, on another that Queen Victoria is becoming emotionless, and on a third that Mount St. Helens is cooling off; but the logical form does no more than specify the feminine gender of the referent and its property of “becoming cold (in some sense) at the time of utterance”. Our primary concern in this paper will be with the semantic rules that define immediate logical form, although we attempt to define this form in a way that minimizes the remaining gap to the deeper representation.

All the experience gained within AI and linguistics suggests that bridging this final gap will be very difficult. Some would take as their lesson that research efforts should concentrate on the last, pragmatic phase of comprehension, where ‘the real problems’ lie. We believe on the contrary that the only way to make the pragmatic problems tractable is to have a precise conception of the constituent structure and logical form of the natural language input, in terms of which the pragmatic operations can in turn be precisely formulated.

In AI research, the objectives of clarity and extensibility have often been sacrificed to immediate performance goals. One reason for this may have been the need to establish the credibility of a relatively young and controversial discipline. In any case, the state of linguistic theory until fairly recently left no real alternatives. The transformational grammars whose study dominated theoretical linguistics seemed a poor prospect even for the limited goal of describing natural language syntax, because of the subtlety of transformational rules and supplementary devices such as co-indexing procedures, filters and constraints on

movement, and the complexity of their interactions. Moreover, the prospects for writing efficient transformational parsers seemed poor, given that transformational grammars can in principle generate all recursively enumerable languages. But most importantly, generative grammarians developed syntactic theories more or less independently of any semantic considerations, offering no guidance to AI researchers whose primary objective was to compute ‘meaning representations’ for natural language utterances. Katz and Fodor’s markerese (Katz & Fodor 1963) was patently inadequate as a meaning representation language from an AI point of view, and Generative Semantics (Lakoff 1971) never did develop into a formal theory of the relation between surface form and meaning.

Theoretical linguistics took an important new turn with the work of Montague on the logic of English and later expansions and variants of his theory (e.g., see Thomason 1974a, Partee 1976a, and Cresswell 1973). According to Montague grammar the correspondence between syntactic structure and logical form is much simpler than had generally been supposed: to each lexeme there corresponds a logical term or functor and to each rule of syntactic composition there corresponds a structurally analogous semantic rule of logical composition; this is the so-called *rule-to-rule* hypothesis [Bach 1976].<sup>2</sup> Furthermore, the translations of all constituents of a particular syntactic category are assigned formal meanings of the same set-theoretic type; for example, all NPs, be they names or definite or indefinite descriptions, are taken to denote property sets. Crucially, the formal semantics of the logical translations produced by the semantic rules of Montague grammar accords by and large with intuitions about entailment, synonymy, ambiguity and other semantic phenomena.

<sup>2</sup> Interestingly enough, this linguistic hypothesis was anticipated by Knuth’s work on the semantics of attribute grammars (Knuth 1968). Schwind (1978) has applied Knuth’s insights to the development of a formal basis for question answering systems, anticipating some of the work by Gazdar and others on which our own efforts are founded.

There is also some similarity between the rule-to-rule hypothesis and the rule-based approach to the interpretation of syntactic structures that emerged within AI during the 1960’s and early 70’s. The idea of pairing semantic rules with phrase structure rules was at the heart of DEACON (Craig et al. 1966), a system based on F. B. Thompson’s proposal to formalize English by limiting its subject matter to well-defined computer memory structures (Thompson 1966). However, DEACON’s semantic rules performed direct semantic evaluation of sorts (via computations over a data base) rather than constructing logical translations. The systems of Winograd (1972) and Woods (1977) constructed input translations prior to evaluation, using semantic rules associated with particular syntactic structures. However, these rules neither corresponded one-to-one to syntactic rules nor limited interpretive operations to composition of logical expressions; for example, they incorporated tests for selectional restrictions and other forms of inference, with unrestricted use of the computational power of LISP.

The chief limitation of Montague's grammar was that it treated only very small, syntactically (though not semantically) simple fragments of English, and efforts were soon under way to extend the fragments, in some cases by addition of a transformational component (Partee 1976b, Cooper & Parsons 1976). At the same time, however, linguists dissatisfied with transformational theory were beginning to develop non-transformational alternatives to traditional generative grammars (e.g., Peters & Ritchie 1969, Bresnan 1978, Lapointe 1977, Brame 1978, Langendoen 1979). A particularly promising theory that emerged from this development, and explicitly incorporates Montague's approach to semantics, is the phrase structure theory advanced by Gazdar and others (Gazdar 1980, 1981, Gazdar, Pullum & Sag 1980, Gazdar & Sag 1980, Sag 1980, Gazdar, Klein, Pullum & Sag, to appear). The theory covers a wide range of the syntactic phenomena that have exercised transformationalists from Chomsky onward, including subcategorization, coordination, passivization, and unbounded dependencies such as those occurring in topicalization, relative clause constructions and comparatives. Yet the grammar itself makes no use of transformations; it consists entirely of phrase structure rules, with a node-admissibility rather than generative interpretation. For example, the rule [(S) (NP) (VP)] states that a fragment with root S, left branch NP and right branch VP is an admissible fragment of a syntactic tree.<sup>3</sup> Such phrase structure rules are easy to understand and permit the use of efficient context-free parsing methods. Moreover, the grammar realizes the rule-to-rule hypothesis, pairing each syntactic rule with a Montague-like semantic rule that supplies the intensional logic translation of the constituent admitted by the syntactic rule.

It has long been assumed by transformationalists that linguistic generalizations cannot be adequately captured in a grammar devoid of transformations. Gazdar refutes the assumption by using metagrammatical devices to achieve descriptive elegance. These devices include *rule-schemata* (e.g., coordination schemata that yield the rules of coordinate structure for all coordinators and all syntactic categories), and *metarules* (e.g., a passive metarule that takes any transitive-VP rule as 'input' and generates a corresponding passive-VP rule as 'output' by deleting the

object NP from the input rule and appending an optional *by-PP*). Although metarules resemble transformational rules, they map rules into rules rather than trees into trees, leaving the grammar itself context-free. Another key innovation is the use of categories with 'gaps', such as NP/PP, denoting a NP from which a PP has been deleted (*not* necessarily at the top level). A simple metarule and a few rule schemata are used to introduce rules involving such derived categories, elegantly capturing unbounded dependencies.

The character of the syntactic theory will become clearer in Section 4, where we supply a sampling of grammatical rules (with our variants of the semantic rules), along with the basic metarule for passives and the coordination schemata. First, however, we would like to motivate our attempt to reformulate Gazdar's semantic rules so as to yield 'conventional' logical translations (Section 2), and to explain the syntactic and semantic idiosyncrasies of our target logic (Section 3).

By 'conventional' logics we mean first-order (and perhaps second-order) predicate logics, augmented with a lambda operator, necessity operator, propositional attitude operators and perhaps other non-extensional propositional operators, and with a Kripke-style possible-worlds semantics (Hughes & Cresswell 1968).<sup>4</sup> The logic employed by Montague in his first formal fragment of English comes rather close to what we have in mind (Montague 1970a), while the intensional logics of the later fragments introduce the unconventional features we hope to avoid (1970b,c). It is the treatment in these later fragments that is usually referred to by the term "Montague grammar". (For a detailed discussion of the distinction between conventional logics in the above sense and intensional logics, see Guenther 1978).

We should stress that it is semantics, not syntax, which is the crux of the distinction. We shall take certain liberties with conventional logical syntax, aligning it more nearly with the surface structure; but this will not lead to major departures from conventional semantics. For example, our syntax of terms allows syntactically unfamiliar formulas such as

[<all1 man2> mortal3].

<sup>3</sup> We use traditional category symbols in our exposition, occasionally followed by supplementary features, e.g., (V TRAN) for transitive verb. Gazdar actually assumes a two-bar  $\bar{X}$  system (e.g., see Bresnan 1976, Jackendoff 1977) that distinguishes between  $\bar{X}$ ,  $\bar{X}$  and X categories (e.g.,  $\bar{V}$ ,  $\bar{V}$ , and V, equivalent to the traditional S, VP and V respectively) and employs complex symbols whose first component specifies the 'number of bars' and whose second component supplies a feature bundle encoding syntactic category, subcategorization, and morphosyntactic and morphological information.

<sup>4</sup> We admit predicate modifiers and some second-order predicate constants into our logical vocabulary, and may ultimately want to employ a full-fledged second-order logic, in view of such sentences as "Every good general has at least some of Napoleon's qualities". On the other hand, we may pare down rather than expand the logical apparatus, opting for a logic that treats properties, propositions and other intensional entities as first-order individuals. This type of treatment, which avoids the unwanted identity of logically equivalent propositions, appears to be gaining currency (e.g., Fodor 1978, McCarthy 1979, Thomason 1980, Chierchia 1981). Some minor adjustments would be required in our rules of logical translation.

But the formula derives its interpretation from its stipulated logical equivalence to

$$(\text{all1 } x:[x \text{ man2}])[x \text{ mortal3}],$$

which may in turn become

$$\forall x[[x \text{ HUMAN}] \Rightarrow [x \text{ MORTAL}]],$$

after disambiguation.<sup>5</sup>

## 2. Intensional and 'Conventional' Translations

We should emphasize at the outset that our objective is not to impugn Montague grammar, but merely to make the point that the choice between intensional and conventional translations is as yet unclear. Given that the conventional approach appears to have certain advantages, it is worth finding out where it leads; but we are not irrevocably committed to this approach. Fortunately, the translation component of a parser for a Gazdar-style grammar is easily replaced.

Montague grammarians assume that natural languages closely resemble formal logical systems; more specifically, they postulate a strict homomorphism from the syntactic categories and rules of a natural language to the semantic categories and rules required for its formal interpretation. This postulate has led them to an analysis of the logical content of natural language sentences which differs in important respects from the sorts of analyses traditionally employed by philosophers of language (as well as linguists and AI researchers, when they have explicitly concerned themselves with logical content).

The most obvious difference is that intensional logic translations of natural language sentences conform closely with the surface structure of those sentences, except for some re-ordering of phrases, the introduction of brackets, variables and certain logical operators, and (perhaps) the reduction of idioms. For example, since the constituent structure of "John loves Mary" is

[John [loves Mary]],

the intensional logic translation likewise isolates a component translating the VP "loves Mary", composing this VP-translation with the translation of "John" to give the sentence formula. By contrast, a conventional translation will have the structure

[John loves Mary],

in which "John" and "Mary" combine with the verb at the *same* level of constituent structure.

In itself, this difference is not important. It only becomes important when syntactic composition is assumed to correspond to *function application* in the semantic domain. This is done in Montague grammar

<sup>5</sup> We consistently use infix form (with the predicate following its first argument) and square brackets for complete sentential formulas.

by resort to the Schoenfinkel-Church treatment of many-place functions as one-place functions (Schoenfinkel 1924, Church 1941). For example, the predicate "loves" in the above sentence is interpreted as a one-place function *that yields a one-place function* when applied to its argument (in this instance, when applied to the semantic value of "Mary", it yields the function that is the semantic value of "loves Mary"). The resultant function in turn yields a sentence value when applied to its argument (in this instance, when applied to the semantic value of "John", it yields the proposition expressed by "John loves Mary"). Thus, a dyadic predicator like "loves" is no longer interpreted as a set of pairs of individuals (at each possible world or index), but rather as a function into functions. Similarly a triadic predicator like "gives" is interpreted as a function into functions into functions.

Moreover, the arguments of these functions are not individuals, because NPs in general and names in particular are assumed to denote property sets (or truth functions over properties) rather than individuals. It is easy to see how the postulate of syntactic-semantic homomorphism leads to this further retreat from traditional semantics. Consider Gazdar's top-level rule of declarative sentence structure and meaning:

$$\langle 10, [(S) (NP) (VP)], (VP' NP'') \rangle.$$

The first element of this triple supplies the rule number (which we have set to 10 for consistency with the sample grammar of Section 4), the second the syntactic rule and the third the semantic rule. The semantic rule states that the intensional logic translation of the S-constituent is compounded of the VP-translation (as functor) and the NP-translation (as operand), where the latter is first to be prefixed with the intension operator  $\wedge$ . In general, a primed syntactic symbol denotes the logical translation of the corresponding constituent, and a double-primed symbol the logical translation prefixed with the intension operator (thus, NP'' stands for  $\wedge$ NP').

For example, if the NP dominates "John" and the VP dominates "loves Mary", then S' (the translation of S) is

$$((\text{loves}' \wedge \text{Mary}') \wedge \text{John}').$$

Similarly the translation of "Every boy loves Mary" comes out as

$$((\text{loves}' \wedge \text{Mary}') \wedge (\text{every}' \text{boy}')),$$

given suitable rules of NP and VP formation.<sup>6</sup> Note the uniform treatment of NPs in the logical formulas, i.e., (every' boy') is treated as being of the same semantic category as John', namely the (unique) seman-

<sup>6</sup> The exact function of the intension operator need not concern us here. Roughly speaking, it is used to bring meanings within the domain of discourse; e.g., while an NP' denotes a property set at each index, the corresponding  $\wedge$ NP' denotes the entire NP intension (mapping from indices to property sets) at each index.

tic category corresponding to the syntactic category NP. What is the set-theoretic type of that category? Since (every' boy') cannot be interpreted as denoting an individual (at least not without making the rules of semantic valuation for formulas depend on the structure of the terms they contain), neither can John'. The solution is to regard NPs as denoting sets of properties, where a property determines a set of individuals at each index, and VPs as sets of such property sets (or in functional terms, as truth functions over truth functions over properties). Thus John' does not denote an individual, but rather a set of properties, namely those which John has; (every' boy') denotes the set of properties shared by all boys, (a' boy') the set of all properties possessed by at least one boy, and so on. It is not hard to see that the interpretation of VPs as sets of property sets then leads to the appropriate truth conditions for sentences.<sup>7</sup>

With respect to our objective of building a comprehensible, expandable natural language understanding system, the simplicity of Gazdar's semantic rules and their one-to-one correspondence to phrase structure rules is extremely attractive; however, the semantics of the intensional logic translations, as sketched above, seems to us quite unnatural.

Admittedly naturalness is partly a matter of familiarity, and we are not about to fault Montague grammar for having novel features (as some writers do, e.g., Harman 1975). But Montague's semantics is at variance with pretheoretical intuitions as well as philosophical tradition, as Montague himself acknowledged (1970c:268). Intuitively, names denote individuals (when they denote anything real), not sets of properties of individuals; extensional transitive verbs express relations between pairs of individuals, not between pairs of property sets, and so on; and intuitively, quantified terms such as "everyone" and "no-one" simply *don't* bear the same sort of relationship to objects in the world as names, even though the evidence for placing them in the same syntactic category is overwhelming. Such objections would carry no weight if the sole purpose of formal semantics were to provide an explication of intuitions about truth and logical consequence, for in that area intensional logic is remarkably successful. But formal semantics should also do justice to our intuitions about the relationship between word and object, where those intuitions are clear – and intensional logic seems at odds with some of the clearest of those intuitions.<sup>8</sup>

<sup>7</sup> This was the approach in Montague (1970b) and is adopted in Gazdar (1981a). In another, less commonly adopted approach NPs are still interpreted as sets of properties but VPs are interpreted simply as properties, the truth condition for a sentence being that the property denoted by the VP be in the set of properties denoted by the NP (Montague 1970c, Cresswell 1973). In other words, the NP is thought of as predicating something about the VP, rather than the other way around.

There is also a computational objection to intensional logic translations. As indicated in our introductory remarks, a natural language understanding system must be able to make *inferences* that relate the natural language input to the system's stored knowledge and discourse model. A great deal of work in AI has focused on inference during language understanding and on the organization of the base of stored knowledge on which the comprehension process draws. Almost all of this work has employed more or less conventional logics for expressing the stored knowledge. (Even such idiosyncratic formalisms as Schank's conceptual dependency theory (Schank 1973) are much more akin to, say, first order modal logic than to any form of intensional logic – see Schubert 1976). How are intensional logic formulas to be connected up with stored knowledge of this conventional type?

One possible answer is that the stored knowledge should not be of the conventional type at all, but should itself be expressed in intensional logic. However, the history of automatic deduction suggests that higher-order logics are significantly harder to mechanize than lower-order logics. Developing efficient inference rules and strategies for intensional logics, with their arbitrarily complex types and their intension, extension and lambda abstraction operators in addition to the usual modal operators, promises to be very difficult indeed.

Another possible answer is that the intensional logic translations of input sentences should be post-processed to yield translations expressed in the lower-order, more conventional logic of the system's knowledge base. A difficulty with this answer is that discourse inferences need to be computed 'on the fly' to guide syntactic choices. For example, in the sentences "John saw the bird without binoculars" and "John saw the bird without tail feathers" the syntactic roles of the prepositional phrases (i.e., whether they modify "saw" or "the bird") can only be determined by inference. One could uncouple inference from parsing by computing all possible parses and choosing among the resultant translations, but this would be cumbersome and psychologically implausible at best.

As a final remark on the disadvantages of intensional translations, we note that Montague grammar relies heavily on *meaning postulates* to deliver simple consequences, such as

A boy smiles - There is a boy;

<sup>8</sup> Thomason reminds us that "...we should not forget the firmest and most irrefragable kind of data with which a semantic theory must cope. The theory must harmonize with the actual denotations taken by the expressions of natural languages,..." but confines his further remarks to sentence denotations, i.e., truth values (Thomason, 1974b:54).

(in this instance an extensionalizing postulate is required for “smiles” – see Montague 1970c:263). A conventional approach dispensing with postulates of this type would be preferable.

Having stated our misgivings about Montague grammar, we need to confront the evidence in its favour. Are there compelling reasons for regarding sentential constituents as more or less directly and uniformly interpretable? In support of the affirmative, one can point out the simplicity and elegance of this strategy from a logical point of view. More tellingly, one can cite its success record: it has made possible for the first time the formal characterization of non-trivial fragments of natural languages, with precisely defined syntactic-semantic mappings; and as one would hope, the formal semantics accounts for many cases of entailment, ambiguity, contradictoriness, and other semantic phenomena, including some of the subtlest arising from intensional locutions.

Concerning the simplicity of the strategy, we note that the connection between language and the world could be just as simple as Montague grammar would have it, without being quite so direct. Suppose, for a moment, that people communicated in first-order logic. Then, to express that Al, Bill and Clyde were born and raised in New York, we would have to say, in effect, “Al was born in New York. Al was raised in New York. Bill was born in New York. ... Clyde was raised in New York.” The pressure to condense such redundant verbalizations would be great, and might well lead to ‘overlay’ verbalizations in which lists enumerating the non-repeated constituents were fitted into a common sentential matrix. In other words, it might lead to something like constituent coordination. But unlike simple constituents, coordinated constituents would not be meaningful in isolation; they would realize their meaning only upon expansion of the embedding overlay verbalization into a set of first-order formulas. Yet the connection between language and the world would remain simple, assuming that the *syntactic* relation between overlay verbalizations and their first-order translations were simple. It would be quite pointless to reconstrue the semantics of the enhanced language so as to align the denotations of names with the denotations of coordinated names, for example, as is done in Montague grammar. While formally simplifying the semantic mapping function, such a move would lead to complex and counterintuitive semantic types.

The success of Montague grammar in characterizing fragments of natural languages, with a proper account of logical relations such as entailment, is indeed strong evidence in its favour. The only way of challenging this success is to offer an equally simple, equally viable alternative. In part, this paper is intended as a move in that direction. While we do not explicitly

discuss logical relations between the translations of sentences, the kinds of translations produced by the sample grammar in Section 4 should at least provide some basis for discussion. To the extent that the translations are of a conventional type (or easily converted to conventional form), the entailment relations should be more or less self-evident.

There is one linguistic phenomenon, however, which deserves preliminary comment since it might be thought to provide conclusive evidence in favour of Montague grammar, or at least in favour of the intensional treatment of NPs. This concerns intensional verbs such as those in sentences (1) and (2), and perhaps (3):

- (1) John looks for a unicorn,
- (2) John imagines a unicorn,
- (3) John worships a unicorn.

These sentences admit non-referential readings with respect to the NP “a unicorn”, i.e., readings that do not entail the existence of a unicorn which is the referent of the NP. In intensional logic the nonreferential reading of the first sentence would simply be

((looks-for'  $\wedge$  (a' unicorn'))  $\wedge$  John').

The formal semantic analysis of this formula turns out just as required; that is, its value can be “true” or “false” (in a given possible world) irrespective of whether or not there are unicorns (in that world). The referential reading is a little more complicated, but presents no difficulties.

It is the non-referential reading which is troublesome for conventional logics. For the first sentence, there seems to be only one conventional translation, viz.,

$\exists x[[\text{John looks-for } x] \ \& \ [x \text{ unicorn}]]$ ,

and of course, this is the referential reading. There is no direct way of representing the non-referential reading, since the scope of a quantifier in conventional logics is always a sentence, never a term.

The only possible escape from the difficulty lies in translating intensional verbs as complex(non-atomic) logical expressions involving opaque sentential operators.<sup>9</sup> The extant literature on this subject supports the view that a satisfactory decomposition cannot be supplied in all cases (Montague 1970c, Bennett 1974, Partee 1974, Dowty 1978, 1979, Dowty, Wall & Peters 1981). A review of this literature would be out of place here; but we would like to indicate that the case against decomposition (and hence against conventional translations) is not closed, by offering the fol-

<sup>9</sup> With regard to our system-building objectives, such resort to lexical decomposition is no liability: the need for some use of lexical decomposition to obtain “canonical” representations that facilitate inference is widely acknowledged by AI researchers, and carried to extremes by some (e.g., Wilks 1974, Schank 1975).

lowing paraphrases of the three sample sentences. (Paraphrase (1)' is well-known, except perhaps for the particular form of adverbial (Quine 1960, Bennett 1974, Partee 1974), while (2)'-(3)'' are original). These could be formalized within a conventional logical framework allowing for non-truth-functional sentential operators:

- (1)' John *tries* to find a unicorn (by looking around),  
 (2)' John forms a mental description which *could* apply to a unicorn,  
 (3)' John *acts, thinks and feels as if* he worshipped a unicorn.  
 (3)'' John worships an entity which *he believes* to be a unicorn.

In each case the operator that is the key to the translation is italicized. Note that the original ambiguity of (1) and (2) has been preserved, but can now be construed as a quantifier scope ambiguity in the conventional fashion. In (3)' and (3)'' the embedded "worships" is to be taken in a veridical sense that entails the existence of the worshippee. It is important to understand that the translations corresponding to (3)' and (3)'' would not be obtained *directly* by applying the rules of the grammar to the original sentence; rather, they would be obtained by *amending* the direct translation, which is patently false for a hearer who interprets "worships" veridically and does not believe in unicorns. Thus we are presupposing a mechanism similar to that required to interpret metaphor on a Gricean account (Grice 1975). The notion of "acting, thinking and feeling as if..." may seem rather ad hoc, but appears to be applicable in a wide variety of cases where (arguably) non-intensional verbs of human action and attitude are used non-referentially, as perhaps in "John is communing with a spirit", "John is afraid of the boogie-man in the attic", or "John is tracking down a sasquatch". Formulation (3)'' represents a more radical alternative, since it supplies an acceptable interpretation of (3) only if the entity *actually* worshipped by John may be an 'imaginary unicorn'. But we may need to add imaginary entities to our 'ontological stable' in any event, since entities may be *explicitly* described as imaginary (fictitious, hypothetical, supposed) and yet be freely referred to in ordinary discourse. Also, sentences such as "John frequently dreams about a certain unicorn" (based on an example in Dowty, Wall and Peters 1981) seem to be untranslatable into any logic without recourse to imaginary entities. Our paraphrases of (3) have the important advantage of entailing that John has a *specific* unicorn in mind, as intuitively required (in contrast with (1) and (2)). This is not the case for the intensional logic translation of (3) analogous to that of (1), a fact that led Bennett to regard "worships" – correctly, we think – as extensional (Bennett 1974).

In the light of these considerations, the conventional approach to logical translation seems well worth pursuing. The simplicity of the semantic rules to which we are led encourages us in this pursuit.

### 3. Syntactic and Semantic Preliminaries

The logical-form syntax provides for the formation of simple terms such as

John<sub>1</sub>, x,

quantified terms such as

<some<sub>1</sub> man<sub>2</sub>>, <the<sub>1</sub> (little<sub>2</sub> boy<sub>3</sub>)>,

simple predicate formulas such as

man<sub>2</sub>, loves<sub>3</sub>, P<sub>4</sub>,

compound predicate formulas such as

<loves<sub>2</sub> Mary<sub>3</sub>>, <<loves<sub>2</sub> Mary<sub>3</sub>> John<sub>1</sub>>,  
 [John<sub>1</sub> loves<sub>2</sub> Mary<sub>3</sub>],

modified predicate formulas such as

(bright<sub>3</sub> red<sub>4</sub>), (passionately<sub>2</sub> <loves<sub>3</sub> Mary<sub>4</sub>>),

and lambda abstracts such as

$\lambda x[x \text{ shaves}_2 x]$ ,  $\lambda y[y \text{ expects}_2 [y \text{ wins}_4]]$ .

Note the use of sharp angle brackets for quantified terms, square brackets or blunt angle brackets for compound predicate formulas, and round brackets for modified predicate formulas. (We explain the use of square brackets and blunt angle brackets below.) We also permit sentences (i.e., compound predicate formulas with all arguments in place) as operands of sentential operators, as in

[[John<sub>5</sub> loves<sub>6</sub> Mary<sub>7</sub>] possible<sub>3</sub>],  
 [Sue<sub>1</sub> believes<sub>2</sub> [John<sub>5</sub> loves<sub>5</sub> Mary<sub>6</sub>]],  
 [[John<sub>1</sub> feverish<sub>3</sub>] because<sub>4</sub>  
 [John<sub>1</sub> has<sub>5</sub> malaria<sub>6</sub>]].

For coordination of expressions of all types (quantifiers, terms, predicate formulas, modifiers, and sentential operators) we use sharp angle brackets and prefix form, as in

<or<sub>2</sub> many<sub>1</sub> few<sub>3</sub>>, <and<sub>2</sub> John<sub>1</sub> Bill<sub>3</sub>>,  
 <and<sub>4</sub> <hugs<sub>2</sub> Mary<sub>3</sub>> <kisses<sub>5</sub> Sue<sub>6</sub>>>.

The resemblance of coordinated expressions to quantified terms is intentional: in both cases the sharp angle brackets signal the presence of an unscoped operator (viz., the first element in brackets) to be scoped later on.

Finally, we may want to admit second-order predicates with first-order predicate arguments, as in

[Fido<sub>1</sub> little-for<sub>3</sub> dog<sub>5</sub>], [blue<sub>1</sub> colour<sub>4</sub>],  
 $[\lambda x[x \text{ kisses}_1 \text{ Mary}_2] \text{ is-fun}_3]$ ,

though it remains to be seen whether such second-order predications adequately capture the meaning of English sentences involving implicit comparatives and nominalization.

Fuller explanations of several of the above features follow. In outline, we first delve a little further into the syntax and semantics of predicate formulas; then we discuss the sources and significance of ambiguities in the formulas.

Atomic sentences are of the form

$$[t_n P t_1 \dots t_{n-1}], \text{ (equivalently, } \langle P t_1 \dots t_n \rangle),$$

where  $t_1, \dots, t_n$  are terms and  $P$  is a predicate constant, and the square brackets and blunt angle brackets distinguish infix and prefix syntax respectively. We regard this sentential form as equivalent to

$$[t_n \langle \dots \langle (P t_1) t_2 \rangle \dots t_{n-1} \rangle],$$

i.e., as obtained by applying an  $n$ -ary predicate *successively* to  $n$  terms. For example,

$$\begin{aligned} [\text{John loves Mary}] &= \langle \text{loves Mary John} \rangle \\ &= [\text{John } \langle \text{loves Mary} \rangle] = \langle \langle \text{loves Mary} \rangle \text{ John} \rangle. \end{aligned} \text{ }^{10}$$

As in Montague grammar, this predicate application syntax helps to keep the rules of translation simple: in most cases the translation of a phrase is just the composition of the translations of its top-level constituents. However, we saw earlier that a functional interpretation of predicate application leads to the interpretation of predicates as telescoped function-valued functions, whereas we wish to interpret predicates as  $n$ -ary relations (in each possible world) in the conventional way.

We can satisfy this requirement by interpreting predicate application not as function application, but rather as *leftmost section* of the associated relation at the value of the given argument. For example, let  $V$  denote the semantic valuation function (with a particular interpretation and possible world understood) and let

$$\begin{aligned} V(P) &= \{ \langle a, b, c \rangle, \langle a, b, d \rangle, \langle e, f, g \rangle \}, \\ V(x) &= a, V(y) = b, \text{ and } V(z) = d, \end{aligned}$$

where  $P$  is a triadic predicate symbol,  $x, y,$  and  $z$  are individual constants or variables, and  $a, b, \dots, g$  are elements of the individual domain  $D$ . Then

$$\begin{aligned} V(\langle P x \rangle) &= \{ \langle b, c \rangle, \langle b, d \rangle \}, \\ V(\langle P x y \rangle) &= V(\langle \langle P x \rangle y \rangle) = \{ \langle c \rangle, \langle d \rangle \}, \text{ and} \\ V(\langle [z P x y] \rangle) &= V(\langle \langle \langle P x \rangle y \rangle z \rangle) = \{ \langle \rangle \}. \end{aligned}$$

We use the convention  $\{ \langle \rangle \} = \text{true}, \{ \} = \text{false}$ .

Lambda abstraction can be defined compatibly by

$$V_I(\lambda x \phi) = \{ \{d\} \times V_{I(x:d)}(\phi) \mid d \in D \},$$

where  $I$  is an interpretation,  $I(x:d)$  is an interpretation identical to  $I$  except that  $x$  denotes  $d$ , and  $\times$  denotes Cartesian product (and a particular possible world is

<sup>10</sup> We provide the double syntax for purely cosmetic reasons. In our use of the notation, expressions delimited by square brackets will generally be complete open or closed sentences, while expressions delimited by blunt angle brackets will be 'incomplete sentences', i.e., predicates with one or more arguments missing (and denoting a relation with adicity = number of missing arguments).

understood). It can be verified that the usual lambda-conversion identities hold, i.e.,

$$\langle \lambda x \langle P \dots x \dots \rangle t \rangle = \langle P \dots t \dots \rangle, \text{ and}$$

$$P = \lambda x \langle P x \rangle = \lambda x \lambda y \langle P x y \rangle = \dots,$$

where  $P$  is a predicate of any adicity (including null, if we use  $\{ \langle \rangle \} \times A = A$  for any set  $A$ ).

As far as modified predicate formulas such as (bright3 red4) are concerned, we can interpret the modifiers as functions from  $n$ -ary relations to  $n$ -ary relations (perhaps with  $n$  restricted to 1).

We now turn to a consideration of the potential sources of ambiguity in the formulas. One source of ambiguity noted in the Introduction lies in the primitive logical symbols themselves, which may correspond ambiguously to various proper logical symbols. The ambiguous symbols are obtained by the translator via the first stage of a two-stage lexicon (and with the aid of morphological analysis, not discussed here). This first stage merely distinguishes the formal logical roles of a lexeme, supplying a distinct (but in general still ambiguous) symbol or compound expression for each role, along with syntactic information. For example, the entry for "recover" might distinguish (i) a predicate role with preliminary translation "recovers-from" and the syntactic information that this is a  $V$  admissible in the rule that expands a  $VP$  as a  $V$  optionally followed by a  $(PP \text{ from})$ ; (this information is supplied via the appropriate rule number); and (ii) a predicate role with preliminary translation "recovers" and the syntactic information that this is a  $V$  admissible in the rule that expands a  $VP$  as a  $V$  followed by an  $NP$ .

Having obtained a preliminary translation of a lexeme in keeping with its apparent syntactic role, the translator affixes an index to it which has not yet been used in the current sentence (or if the translation is a compound expression, it affixes the same index to all of its primitive symbols). In this way indexed preliminary translations such as *Mary1, good2, and recovers3* are obtained. For example, the verb translation selected for "recovers" in the sentence context "John recovers the sofa" would be *recovers2, recovers-from2* being ruled out by the presence of the  $NP$  complement. The second stage of the lexicon supplies alternative final translations of the first-stage symbols, which in the case of "recovers" might be *RE-COVERS, REGAINS, and so on*. Naturally, the processors that choose among these final symbols would have to draw on knowledge stored in the propositional data base and in the representation of the discourse context.

A second source of ambiguity lies in quantified terms. The sentence

Someone loves every man



illustrates a quantifier scope ambiguity arising from a syntactically unambiguous construction. Its logical-form translation is

[<some1 one2> loves3 <every4 man5>],

wherein the relative scopes of the quantifiers some1 and every4 are ambiguous. Quantified terms are intended to be 'extracted' in the postprocessing phase to positions left-adjacent to sentential formulas (which may already be prefixed with other quantifiers). A new variable is introduced into each extracted quantifier expression, the angle brackets are changed to round brackets, and the new variable is substituted for all occurrences of the extracted term. (Thus the level of extraction must be 'high' enough to encompass all of these occurrences.) In the above formula, quantifier extraction reveals the implicit ambiguity, yielding either

(some1 x:[x one2])(every4 y:[y man5])[x loves3 y]

or

(every4 y:[y man5])(some1 x:[x one2])[x loves3 y],

depending on the order of extraction.

Assuming that some1 and every4 correspond to the standard existential and universal quantifiers, these translations could be further processed to yield

$\exists x[[x \text{ one2}] \ \& \ \forall y[[y \text{ man5}] \Rightarrow [x \text{ loves3 } y]]]$  and  
 $\forall y[[y \text{ man5}] \Rightarrow \exists x[[x \text{ one2}] \ \& \ [x \text{ loves3 } y]]]$ .

However, we may not implement this last conversion step, since it cannot be carried out for all quantifiers. For example, as Cresswell remarks, "most A's are B's" cannot be rendered as "for most x, either x is not an A or x is a B" (Cresswell 1973: 137). (Consider, for instance, A = dog and B = beagle; then the last statement is true merely because most things are not dogs – irrespective of whether or not most dogs are in fact beagles.) It appears from recent work by Goebel (to appear) that standard mechanical inference methods can readily be extended to deal with formulas with restricted quantifiers.

A third source of ambiguity lies in coordinated expressions. For example, the logical form of the sentence "Every man loves Peggy or Sue" is

[<every1 man2> loves3 <or5 Peggy4 Sue6>],

which is open to the readings

(every1 x:[x man2])[ [x loves3 Peggy4] or5  
 [x loves3 Sue6]]

and

[(every1 x:[x man2])[x loves3 Peggy4]  
 or5 (every1 x:[x man2])[x loves3 Sue6]].

The postprocessing steps required to scope coordinators are similar to those for quantifiers and are illustrated in Section 4.<sup>11</sup>

An important constraint on the disambiguation of the basic symbols as well as quantified terms and coordinated expressions is that identical expressions (i.e.,

expressions with identical constituent structure, including indices) must be identically disambiguated. For example, "John shaves himself" and "John shaves John" translate respectively into

[John1  $\lambda x[x \text{ shaves2 } x]$ ] = [John1 shaves2 John1],  
 and

[John1 shaves2 John3].

The stated constraint ensures that both occurrences of John1 in the first formula will ultimately be replaced by the same unambiguous constant. Similarly "Someone shaves himself" and "Someone shaves someone" translate initially into

[<some1 one2> shaves3 <some1 one2>] and  
 [<some1 one2> shaves3 <some4 one5>]

respectively, and these translations become

(some1 x:[x one2])[x shaves3 x] and

(some1 x:[x one2])(some4 y:[y one5])[x shaves3 y]

respectively after quantifier extraction. Note that the two occurrences of <some1 one2> in the first formula are extracted in unison and replaced by a common variable. Indexing will be seen to play a similar role in the distribution of coordinators that coordinate non-sentential constituents.

By allowing the above types of ambiguities in the logical form translations, we are able to separate the problem of disambiguation from the problems of parsing and translation. This is an important advantage, since disambiguation depends upon pragmatic factors. For example, "John admires John" may refer to two distinct individuals or just to one (perhaps whimsically), depending on such factors as whether more than one individual named John has been mentioned in the current context. Examples involving ambiguities in nouns, verbs, determiners, etc., are easily supplied. Similarly, the determination of relative quantifier scopes involves pragmatic considerations in addition to level of syntactic embedding and surface order. This is true both for explicit quantifier scope ambiguities such as in the sentence "Someone loves every man", and for scope ambiguities introduced by decomposition, such as the decomposition of "seeks" into

$\lambda y \lambda x[x \text{ tries } [x \text{ finds } y]]$ ,

as a result of which a sentence like

John seeks a unicorn

admits the alternative translations

$\exists x[[x \text{ unicorn}] \ \& \ [\text{John tries } [\text{John finds } x]]]$ , and  
 $[\text{John tries } \exists x[[x \text{ unicorn}] \ \& \ [\text{John finds } x]]]$ ,

neglecting indices. It is simpler to produce a single output which can then be subjected to pragmatic post-

<sup>11</sup> If first-order predicates are to be allowed as arguments of second-order predicates, then quantifier and coordinator scoping of the following types must also be allowed:  $[P \dots \langle Q \ R \rangle \dots] \rightarrow \lambda x(Q \ y:[y \ R])[x \ P \dots y \dots]$ ,  $\langle C \ P \ R \rangle \rightarrow \lambda x[x \ P] \ C \ [x \ R]$ .

processing to determine likely quantifier scopes, than to generate all possible orderings and then to make a pragmatic choice among them. Much the same can be said about scoping of coordinators.

We also note that a grammar designed to generate all possible unambiguous translations of English phrases and sentences would have to supply multiple semantic rules for certain syntactic rules. For example, no one semantic rule can translate a quantifier-noun combination (rule 3 in Section 4) so as to deliver both readings of "Someone loves every man" upon combination of the verb translation with the translations of the NPs. Our use of an ambiguous logical form preserves the rule-to-rule hypothesis.

#### 4. Sample Grammar

Our syntactic rules do not depart significantly from Gazdar's. The semantic rules formally resemble Gazdar's as well, but of course produce conventionally interpretable translations of the type described in the preceding section. As in Gazdar's semantic rules, constituent translations are denoted by primed category symbols such as NP' and V'. The semantic rules show how to assemble such translations (along with the occasional variable and lambda operator) to form the translations of larger constituents. The translations of individual lexemes are obtained as described above.

- <1, [(NP) (PN)], PN'>, PN(1) = {John, Mary, New York, ...}
  - (a) Mary
  - (b) with PN' = Mary6, NP' becomes Mary6.
- <2, [(AN) (ADJP) (N)], (ADJP' N')>, N(2) = {boy, game, noise, ...}
  - (a) little boy
  - (b) with ADJP' = little2, N' = boy3, AN' becomes (little2 boy3);
  - (c) "little" is taken as a predicate modifier.<sup>12</sup>
- <3, [(NP) (Q) (AN)], <Q' AN'>>, Q(3) = {a, the, all, many, ...}
  - (a) the little boy
  - (b) with Q' = the1, AN' = (little2 boy3), NP' -> <the1 (little2 boy3)>.
- <4, [(PP to) (to) (NP)], NP'>
  - (a) to Mary
  - (b) with NP' = Mary6, PP' -> Mary6;
  - (c) PP verb complements have the same meaning as their NP, as per Gazdar (1981a).
- <5, [(VP) (V)], V'>, V(5) = {run, smile, disappear, ...}
  - (a) smiles
  - (b) with V' = smiles4, VP' -> smiles4.

In operation, the translator generates the minimum number of brackets consistent with the notational equivalences stated earlier. For example, in assembling [NP' VP'], with NP' = John1 and VP' = [loves2 Mary3], the result is

[John1 loves2 Mary3],  
rather than  
[John1 (loves2 Mary3)].

Also, in binding a variable with lambda, the translator replaces all occurrences of the variable with a previously unused variable, thus minimizing the need for later renaming. Finally, it performs lambda conversions on the fly. For example, the result of assembling [NP' VP'] with NP' = John1 and

VP' =  $\lambda x[x \text{ shaves2 } x]$ ,  
is  
[John1 shaves2 John1].

The rules that follow have been adapted from Gazdar (1981a). Note that each rule that involves a lexical category such as PN, N or V is accompanied by a specification of the subset of lexical items of that category admissible in the rule. This feature is particularly important for verb subcategorization. In addition, each rule is followed by (a) a sample phrase accepted by the rule, (b) an indication of how the logical translation of the phrase is obtained, and possibly (c) some words of further explanation.

<sup>12</sup> Siegel (1979) argues rather persuasively that measure adjectives, unlike genuine predicate modifiers such as "consummate", actually combine with terms. For such adjectives we might employ the semantic rule  $\lambda x[[x \text{ ADJP}'] \& [x \text{ N}']]$ ; in the case of "little",

we would use ADJP' = (little-for P), where P is an indeterminate predicate to be replaced pragmatically by a comparison-class predicate. Thus the translation of "little boy" (neglecting indices) would be  $\lambda x[[x \text{ little-for P}] \& [x \text{ boy}]]$ .

<6, [(VP) (V) (NP) (PP to)], ⟨V' PP' NP'⟩>

V(6) = {give, hand, tell, ...}

(a) gives Fido to Mary

(b) with V' = gives4, NP' = Fido5, PP' = Mary6, VP' -> ⟨gives4 Mary6 Fido5⟩.

<7, [(VP INF) (to) (VP BASE)], VP'>

(a) to give Fido to Mary

(b) with VP' = ⟨gives4 Mary6 Fido5⟩, the resultant infinitive has the same meaning.

<8, [(VP) (V) (VP INF)],  $\lambda x[x V' [x VP']]$ >,

V(8) = {want, expect, try, ...}

(a) wants to give Fido to Mary

(b) with V' = wants2, VP' = [gives4 Mary6 Fido5],  
VP' ->  $\lambda x3[x3 wants2 [x3 gives4 Mary6 Fido5]]$ ;

(c) The formal lambda variable x given in the semantic rule has been replaced by the new variable x3. Two pairs of square brackets have been deleted, in accordance with the simplification rules stated earlier.

<9, [(VP) (V) (NP) (VP INF)], ⟨V' [NP' VP']⟩>,

V(9) = {want, expect, imagine, ...}

(a) wants Bill to give Fido to Mary

(b) with V' = wants2, NP' = Bill3, VP' = ⟨gives4 Mary6 Fido5⟩,  
VP' -> ⟨wants2 [Bill3 gives4 Mary6 Fido5]⟩.

<10, [(S DECL) (NP) (VP)], [NP' VP']>

(a) the little boy smiles

(b) with NP' = <the1 (little2 boy3)> and VP' = smiles4, the result is  
S' -> [<the1 (little2 boy3)> smiles4]. After pragmatic postprocessing  
to extract quantifiers, the result might be  
S' = (the1 x5:[x5 (little2 boy3)])[x5 smiles4].

Further postprocessing to determine referents and disambiguate operators  
and predicates might then yield

S' = [INDIV17 SMILES1],

where INDIV17 is a (possibly new) logical constant unambiguously denoting  
the referent of (the1 x5:[x5 (little2 boy3)]) and SMILES1 is an unambiguous  
logical predicate.<sup>13</sup> If constant INDIV17 is new, i.e., if the context provided  
no referent for the definite description, a supplementary assertion like

[INDIV17 (LITTLE2 BOY1)]

would be added to the context representation.

(a)' John wants to give Fido to Mary

(b)' with NP' = John1,

VP' =  $\lambda x3[x3 wants2 [x3 gives4 Mary6 Fido5]]$ ,

S' -> [John1 wants2 [John1 gives4 Mary6 Fido5]];

(c)' Note that John1 becomes the subject of both the main clause and the  
embedded (subordinate) clause.

The reader will observe that we have more or less fully traced the derivation and translation of the sentences "The little boy smiles" and "John wants to give Fido to Mary" in the course of the above examples.

The resultant phrase structure trees, with rule numbers and translations indicated at each node, are shown in Figs. 1 and 2.

<sup>13</sup> Definite singular terms often serve as descriptions to be used for referent determination, and in such cases it is the name of the

referent, rather than the description itself, which is ultimately wanted in the formula.

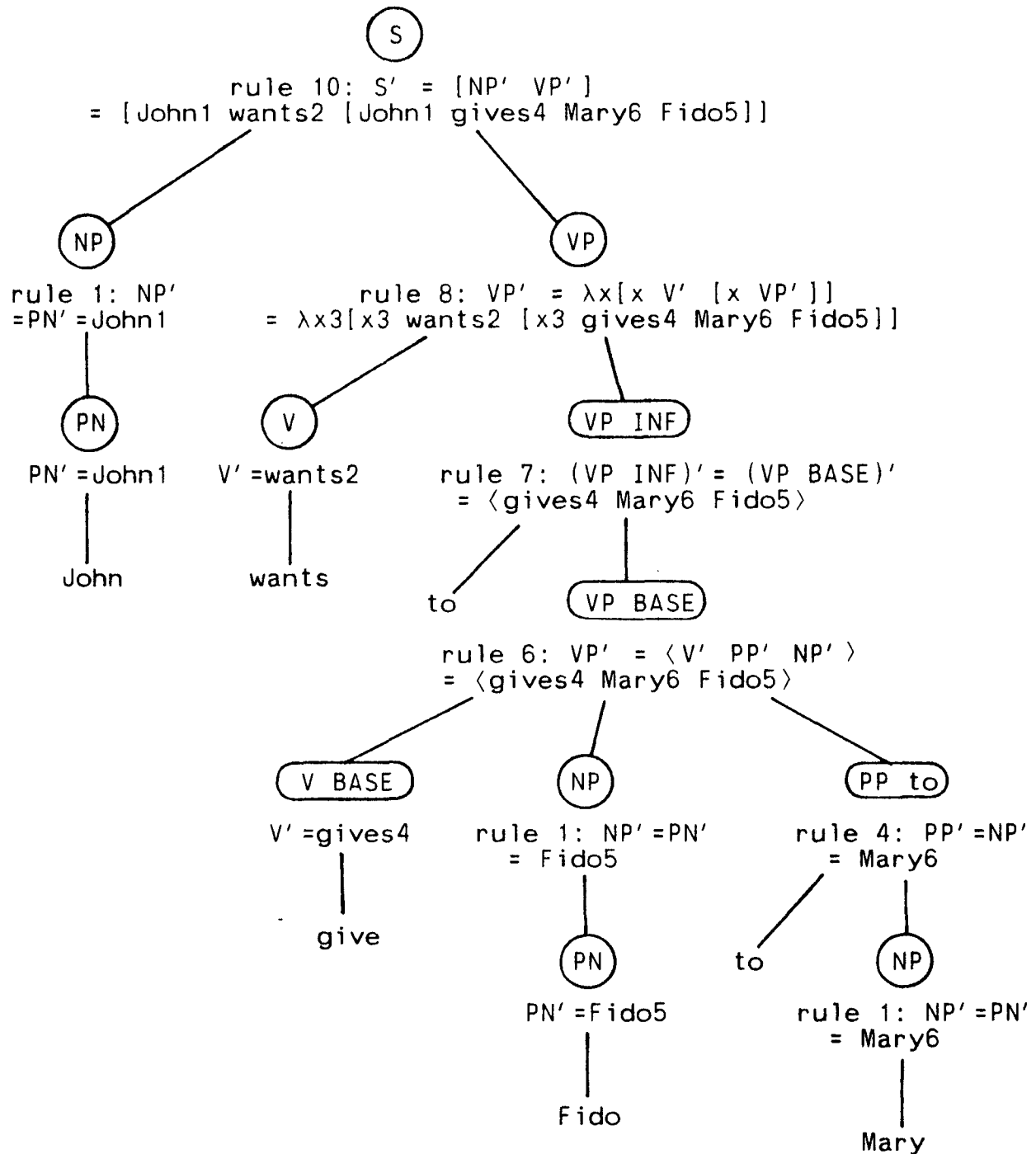


Figure 1. Phrase structure and translation of the sentence "John wants to give Fido to Mary".

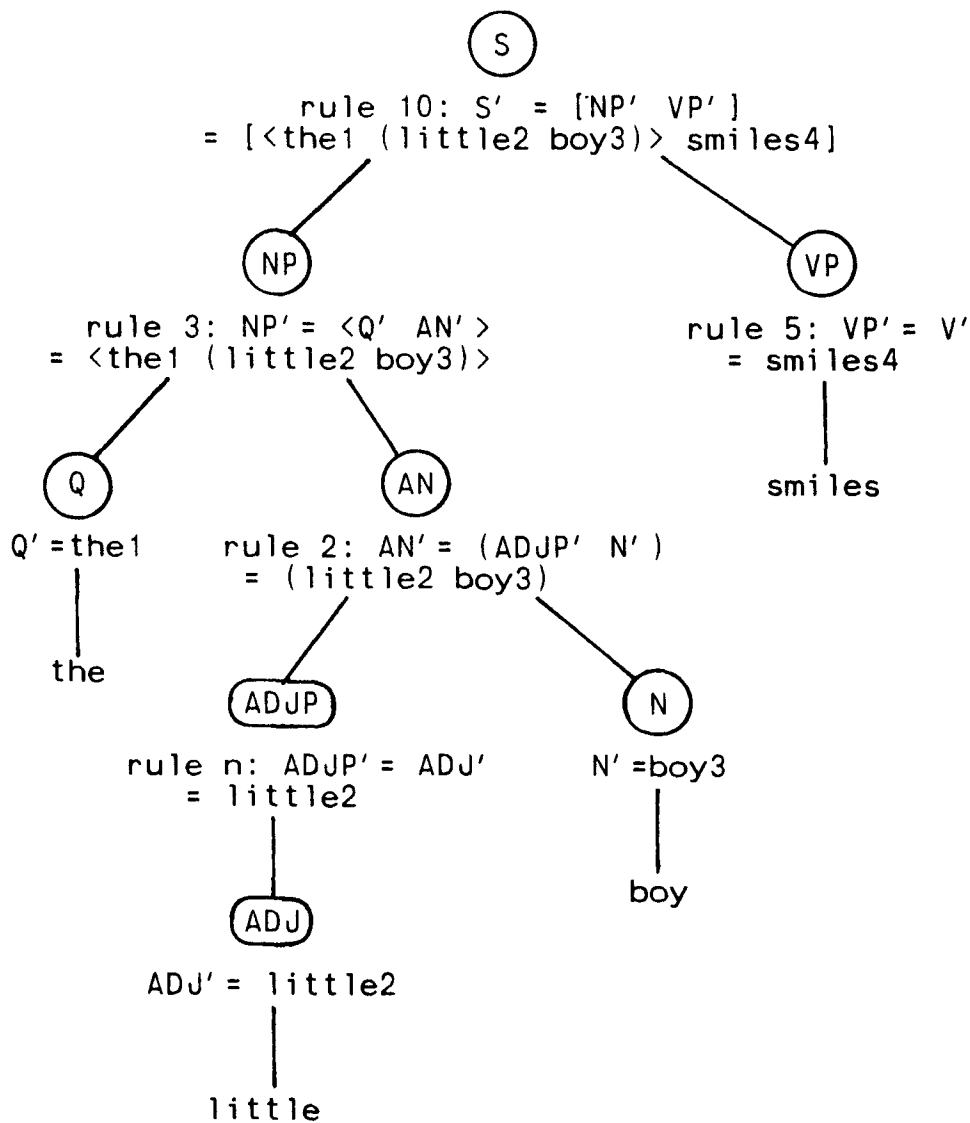


Figure 2. Phrase structure and translation of the sentence "The little boy smiles."

All of the above rules, as well as our versions of the remaining rules in Gazdar (1981a), are as simple as the intensional logic versions or simpler. For example, our semantic rule 8, i.e.,  $\lambda x[x V' [x VP']]$ , may be contrasted with the corresponding rule suggested by Gazdar:

$$\lambda P\{P \lambda x[(V' \wedge (VP' \lambda P(P x))) \lambda P(P x)]\}.$$

Here the lambda variable  $x$ , as in our formula, is used to feed a common logical subject to  $V'$  (the translation of the main verb) and to  $VP'$  (the translation of the embedded infinitive); the variables  $P$  and  $P$ , on the other hand, serve to ensure that the arguments of the  $V'$  and  $VP'$  functions will be of the correct type. Our 'conventional' rule is simpler because it makes no such use of lambda abstraction for type-raising and dispenses with the intension operator.

Gazdar's approach to unbounded dependencies carries over virtually unchanged and can be illustrated with the sentence

To Mary John wants to give Fido.

Here the PP "to Mary" has been topicalized by extraction from "John wants to give Fido to Mary", leaving a PP 'gap' at the extraction site. This 'gap' is syntactically embedded within the infinitive VP "to give Fido", within the main VP "wants to give Fido", and at the highest level, within the sentence "John wants to give Fido". In general, the analysis of un-

bounded dependencies requires *derived rules* for propagating 'gaps' from level to level and *linking rules* for creating and filling them. The linking rules are obtained from the correspondingly numbered basic rules by means of the metarule

$$[A X C Y] ==> [A/B X C/B Y],$$

where  $A$ ,  $B$  and  $C$  may be any basic (i.e., non-slash) syntactic categories such that  $C$  can dominate  $B$ , and  $X$ ,  $Y$  may be any sequences (possibly empty) of basic categories. The linking rules for topicalization are obtained from the rule schemata

$$\langle 11, [B/B t], h \rangle, \text{ and}$$

$$\langle 12, [(S) B (S)/B], (\lambda h S' B') \rangle,$$

where  $B$  ranges over all basic phrasal categories, and  $t$  is a dummy element (trace). The first of these schemata introduces the free variable  $h$  as the translation of the gap, while the second lambda-abstracts on  $h$  and then supplies  $B'$  as the value of the lambda variable, thus 'filling the gap' at the sentence level. At syntactic nodes intermediate between those admitted by schemata 11 and 12, the B-gap is transmitted by derived rules and  $h$  is still free.

Of the following rules, 6, 8, and 10 are the particular derived rules required to propagate the PP-gap in our example and 11 and 12 the particular linking rules that create and fill it:

$$\langle 11, [(PP \text{ to})/(PP \text{ to}) t], h \rangle$$

- (a)  $t$
- (b)  $PP' \rightarrow h$

$$\langle 6, [(VP)/(PP \text{ to}) (V) (NP) (PP \text{ to})/(PP \text{ to})], \langle V' PP' NP' \rangle$$

- (a) give Fido
- (b) with  $V' = \text{gives}_5$ ,  $NP' = \text{Fido}_6$ ,  $PP' = h$ ,  
 $VP' \rightarrow \langle \text{gives}_5 h \text{ Fido}_6 \rangle$
- (c) Note that the semantic rule is unchanged.

$$\langle 8, [(VP)/(PP \text{ to}) (V) (VP \text{ INF})/(PP \text{ to})], \lambda x[x V' [x VP']] \rangle$$

- (a) wants to give Fido
- (b) with  $V' = \text{wants}_3$ ,  $VP' = \langle \text{gives}_5 h \text{ Fido}_6 \rangle$ ,  
 $VP' \rightarrow \lambda x_4[x_4 \text{ wants}_3 [x_4 \text{ gives}_5 h \text{ Fido}_6]]$

$$\langle 10, [(S)/(PP \text{ to}) (NP) (VP)/(PP \text{ to})], [NP' VP'] \rangle$$

- (a) John wants to give Fido
- (b) with  $NP' = \text{John}_2$ ,  
 $VP' = \lambda x_4[x_4 \text{ wants}_3 [x_4 \text{ gives}_5 h \text{ Fido}_6]]$ ,  
 $S' \rightarrow [\text{John}_2 \text{ wants}_3 [\text{John}_2 \text{ gives}_5 h \text{ Fido}_6]]$

$$\langle 12, [(S) (PP \text{ to}) (S)/(PP \text{ to})], (\lambda h S' PP') \rangle$$

- (a) To Mary John wants to give Fido
- (b) With  $S'$  as in 10 (b) above and  $PP' = \text{Mary}_1$ ,  
 $S' \rightarrow [\text{John}_2 \text{ wants}_3 [\text{John}_2 \text{ gives}_5 \text{Mary}_1 \text{ Fido}_6]]$ .
- (c) This translation is logically indistinguishable from the translation of the untopicalized sentence. However, the fronting of "to Mary" has left a pragmatic trace: the

corresponding argument Mary<sub>1</sub> has the lowest index, lower than that of the subject translation John<sub>2</sub> (assuming that symbols are indexed in the order of occurrence of the lexical items they translate). In subsequent pragmatic processing, this feature could be used to detect the special salience of Mary<sub>1</sub>, without re-examination of the superficial sentence form.

Another example of a sentence that can be analyzed by such methods, using relative clause rules similar to those for topicalization, is

Every dog Mary wants to buy is small.

The rules analyze "Mary wants to buy" as an S/NP with translation

[Mary wants [Mary buys h]],

neglecting indices. A further rule reduces the S/NP to an R (relative clause), and its semantic part abstracts on h to yield the predicate

$R' = \lambda h[\text{Mary wants [Mary buys h]}]$

as the translation of the relative clause. The rules for NPs can be formulated in such a way that "every dog" will be translated as

$\langle \text{every } \lambda x[[x \text{ dog}] \ \& \ [x \ R]] \rangle$

where **R** is a free predicate variable that is replaced by the translation of the relative clause when the NP-R rule

$\langle 13, [(NP) (NP) (R)], \langle \lambda RNP' R' \rangle \rangle$

is applied (cf., Gazdar 1981b; we have ignored multiple relative clauses). The resulting NP translation is

$\langle \text{every } \lambda x[[x \text{ dog}] \ \& \ [\text{Mary wants} \\ \text{[Mary buys x]}]] \rangle.$

The translation of the complete sentence, after extraction of the quantifier and conversion of the constraint on the universally quantified variable to an implicative antecedent, would be

$\forall y[[[y \text{ dog}] \ \& \ [\text{Mary wants [Mary buys y]}]] \\ \Rightarrow [y \text{ (small P)}]],$

where P is an undetermined predicate (= dog, in the absence of contrary contextual information).

As a further illustration of Gazdar's approach and how easily it is adapted to our purposes, we consider his metarule for passives:

$\langle [(\text{VP})(\text{V TRAN}) (\text{NP}) X], (\mathcal{F} \text{ NP}'') \rangle \Rightarrow \\ \langle [(\text{VP PASS}) (\text{V}) X \{(\text{PP by})\}], \\ \lambda P((\mathcal{F} \text{ P}) \text{ PP}'') \rangle;$

i.e., "for every active VP rule that expands VP as a transitive verb followed by NP, there is to be a passive VP rule that expands VP as V followed by what, if anything, followed the NP in the active VP rule, followed optionally by a by-PP" (Gazdar 1981a). In the original and resultant semantic rules, ( $\mathcal{F}$  ...) represents the original rule matrix in which NP'' is embedded; thus ( $\mathcal{F} \text{ P}$ ) is the result of substituting the lambda

variable **P** (which varies over NP intensions) for NP'' in the original rule. Intuitively, the lambda variable 'reserves' the NP'' argument position for later binding by the subject of the passive sentence. It can be seen that the metarule will generate a passive VP rule corresponding to our rule 6 which will account for sentences such as "Fido was given to Mary by John". Moreover, if we introduce a ditransitive rule

$\langle 14, [(\text{VP}) (\text{V TRAN}) (\text{NP}) (\text{NP})], \\ \langle \text{V}' \text{ NP}' \text{ NP}' \rangle \rangle^{14}$

to allow for sentences such as "John gave Mary Fido", the metarule will generate a passive VP rule that accounts for "Mary was given Fido by John", in which the indirect rather than direct object has been turned into the sentence subject.

The only change needed for our purposes is the replacement of the property variable **P** introduced by the metarule by an individual variable x:

$\dots(\mathcal{F} \text{ NP}') \dots \Rightarrow \dots \lambda x((\mathcal{F} \text{ x}) \text{ PP}') \dots$

Once the subject NP of the sentence is supplied via rule 10, x is replaced by the translation of that NP upon lambda conversion.

Finally in this section, we shall briefly consider coordination. Gazdar has supplied general coordination rule schemata along with a cross-categorical semantics that assigns appropriate formal meanings to coordinate structures of any category (Gazdar 1980b). Like Gazdar's rules, our rules generate logical-form translations of coordinated constituents such as

$\langle \text{and John Bill} \rangle, \langle \text{or many few} \rangle, \\ \langle \text{and (hugs Mary) (kisses Sue)} \rangle,$

echoing the surface forms. However, it should be clear from our discussion in Section 2 that direct interpretation of expressions translating, say, coordinated NPs or VPs is not compatible with our conventional conception of formal semantics. For example, no formal semantic value is assigned directly to the coordinated term in the formula

$[\langle \text{and John Bill} \rangle \text{ loves Mary}].$

Rather, interpretation is deferred until the pragmatic processor has extracted the coordinator from the embedding sentence (much as in the case of quantified

<sup>14</sup> In the computational version of the semantic rules, primed symbols are actually represented as numbers giving the positions of the corresponding constituents, e.g.,  $\langle 1 \ 2 \ 3 \rangle$  in rule 14. Thus no ambiguity can arise.

terms) and distributed the coordinated terms over duplicate copies of that sentence, yielding

[[John loves Mary] and [Bill loves Mary]].

We adopt the following coordination schemata without change. The superscript denotes sequences of

length  $\geq 1$  of the superscripted element. The schemata are accompanied by examples of phrases they admit, along with (unindexed) translations. The bracketing in (a) and (a)' indicates syntactic structure.

<15, [(A  $\mathcal{F}$ ) ( $\mathcal{F}$ ) (A)], A'>, where A is any syntactic category and  $\mathcal{F} \in \{\text{and, or}\}$   
 (a) and admires  
 (b) admires  
 (a)' or Mary  
 (b)' Mary

<16, [(A) (A)+ (A  $\mathcal{F}$ )], < $\mathcal{F}$ ' A'A'...A'>>  
 (a) loves [and admires]  
 (b) <and loves admires>  
 (a)' [Fido Kim] [or Mary]  
 (b)' <or Fido Kim Mary>

<17, [(A) (A) (A  $\mathcal{F}$ )>], < $\mathcal{F}$ ' A'A'...A'>>  
 (a) Fido [[or Kim] [or Mary]]  
 (b) <or Fido Kim Mary>

The order in which coordinators are extracted and distributed is a matter of pragmatic choice. However, a crucial constraint is that multiple occurrences of a particular coordinated expression (with particular indices) must be extracted and distributed in a single

operation, at the level of a sentential formula whose scope encompasses all of those occurrences (much as in the case of quantifier extraction). The following examples illustrate this process.

- (a) John loves and admires Fido or Kim
- (b) [John<sub>1</sub> <and<sub>3</sub> loves<sub>2</sub> admires<sub>4</sub>> <or<sub>6</sub> Fido<sub>5</sub> Kim<sub>7</sub>>] -> [[John<sub>1</sub> loves<sub>2</sub> <or<sub>6</sub> Fido<sub>5</sub> Kim<sub>7</sub>>] and<sub>3</sub> [John<sub>1</sub> admires<sub>4</sub> <or<sub>6</sub> Fido<sub>5</sub> Kim<sub>7</sub>>]] -> [[[John<sub>1</sub> loves<sub>2</sub> Fido<sub>5</sub>] and<sub>3</sub> [John<sub>1</sub> admires<sub>4</sub> Fido<sub>5</sub>]] or<sub>6</sub> [[John<sub>1</sub> loves<sub>2</sub> Kim<sub>7</sub>] and<sub>3</sub> [John<sub>1</sub> admires<sub>4</sub> Kim<sub>7</sub>]]].
- (c) Note that once the and<sub>3</sub>-conjunction has been chosen for initial extraction and distribution, the simultaneous extraction and distribution of both occurrences of the or<sub>6</sub>-disjunction at the highest sentential level is compulsory. The resultant formula expresses the sense of "John loves and admires Fido or loves and admires Kim". Initial extraction of the or<sub>6</sub>-disjunction would have led to the (implausible) reading "John loves Fido or Kim and admires Fido or Kim" (which is true even if John loves only Fido and admires only Kim).

- (a)' All men want to marry Peggy or Sue
- (b)' [<all<sub>1</sub> man<sub>2</sub>> wants<sub>3</sub> [<all<sub>1</sub> man<sub>2</sub>> marries<sub>4</sub> <or<sub>6</sub> Peggy<sub>5</sub> Sue<sub>7</sub>>]]-> (all<sub>1</sub> x:[x man<sub>2</sub>])[x wants<sub>3</sub> [x marries<sub>4</sub> <or<sub>6</sub> Peggy<sub>5</sub> Sue<sub>7</sub>>]] -> (all<sub>1</sub> x:[x man<sub>2</sub>])[x wants<sub>3</sub> [[x marries<sub>4</sub> Peggy<sub>5</sub>] or<sub>6</sub> [x marries<sub>4</sub> Sue<sub>7</sub>]]].
- (c)' In the second step above, the coordinator or<sub>6</sub> might instead have been raised to the second highest sentential level, yielding (all<sub>1</sub> x:[x man<sub>2</sub>])[x wants<sub>3</sub> [x marries<sub>4</sub> Peggy<sub>5</sub>] or<sub>6</sub> [x wants<sub>3</sub> [x marries<sub>4</sub> Sue<sub>7</sub>]]], or to the highest sentential level, yielding



```
[(all1 x:[x man2])[x wants3 [x marries4 Peggy5]] or6
 (all1 x:[x man2])[x wants3 [x marries4 Sue7]]].
```

The three readings are logically distinct and all are quite plausible (in the absence of additional context). The reader can verify that the first and second readings, but not the third, could have been obtained by extracting the coordinator first and the quantifier second.

Finally, we should remark that the distributive rules are not appropriate for the group reading of coordinate structures in sentences such as

John and Mary carried the sofa (together).

We envisage a mereological interpretation in which John and Mary together comprise a two-component entity. However, we refrain from introducing a logical syntax for such entities here (but see the treatment of plurals in Schubert, 1982).

## 5. Parsing

Phrase structure grammars are relatively easy to parse. The most advanced parser for Gazdar-style grammars that we are aware of is Thompson's chart-parser (Thompson 1981), which provides for slash categories and coordination, but does not (as of this writing) generate logical translations. We have implemented two small parser-translators for preliminary experimentation, one written in SNOBOL and the other in MACLISP. The former uses a recursive descent algorithm and generates intensional logic translations. The latter is a 'left corner' parser that uses our reformulated semantic rules to generate conventional translations. It begins by finding a sequence of left-most phrase-structure-rule branches that lead from the first word upward to the sentence node. (e.g., Mary → PN → NP → S). The remaining branches of the phrase structure rules thus selected form a "frontier" of expectations. Next a similar initial-unit sequence is found to connect the second word of the sentence to the lowest-level (most immediate) expectation, and so on. There is provision for the definition and use of systems of features, although we find that the parser needs to do very little feature checking to stay on the right syntactic track. Neither parser at present handles slash categories and coordination (although they could be handled inefficiently by resort to closure of the grammar under metarules and rule schemata). Extraction of quantifiers from the logical-form translations is at present based on the level of syntactic embedding and left-to-right order alone, and no other form of postprocessing is attempted.<sup>15</sup>

<sup>15</sup> Since submission of this paper for publication, we have become aware of several additional papers on parser-translators similar to ours. One is by Rosenschein & Shieber (1982), another by Gawron et al. (1982); in conception these are based quite directly on the generalized phrase structure grammar of Gazdar and his collaborators, and use recursive descent parsers. A related Prolog-based approach is described by McCord (1981, 1982).

It has been gratifyingly easy to write these parser-translators, confirming us in the conviction that Gazdar-style grammars hold great promise for the design of natural language understanding systems. It is particularly noteworthy that we found the design of the translator component an almost trivial task; no modification of this component will be required even when the parser is expanded to handle slash categories and coordination directly. Encouraged by these results, we have begun to build a full-scale left-corner parser. A morphological analyzer that can work with arbitrary sets of formal affix rules is partially implemented; this work, as well as some ideas on the conventional translation of negative adjective prefixes, plurals, and tense/aspect structure, is reported in Schubert (1982).

## 6. Concluding Remarks

From the point of view of theoretical and computational linguistics, Gazdar's approach to grammar offers profound advantages over traditional approaches: it dispenses with transformations without loss of insight, offers large linguistic coverage, and couples simple, semantically well-motivated rules of translation to the syntactic rules.

We have attempted to show that the advantages of Gazdar's approach to grammar can be secured without commitment to an intensional target logic for the translations of natural language sentences. To motivate this endeavour, we have argued that there are philosophical and practical reasons for preferring a conventional target logic, and that there are as yet no compelling reasons for abandoning such logics in favour of intensional ones. More concretely, we have shown how to reformulate Gazdar's semantic rules to yield conventional translations, and have briefly described some extant PSG parsers, including one that is capable of parsing and translating in accordance with the reformulated Gazdar grammar (minus metalinguistic constructs).

We believe that a parser-interpreter of this type will prove very useful as the first stage of a natural language understanding system. Since the grammar rules are expressed in a concise, individually comprehensible form, such a system will be easy to expand indefinitely. The assignment of a well-defined logical form to input sentences, compatible with favoured knowledge representation formalisms, should help to

bring a measure of precision and clarity to the rather murky area of natural language interpretation by machine.

### Acknowledgements

The authors are indebted to Ivan Sag for a series of very stimulating seminars held by him at the University of Alberta on his linguistic research, and valuable follow-up discussions. The helpful comments of the referees and of Lotfi Zadeh are also appreciated. The research was supported in part by NSERC Operating Grants A8818 and A2252; preliminary work on the left-corner parser was carried out by one of the authors (LKS) under an Alexander von Humboldt fellowship in 1978-79.

### References

- Bach, E. 1976 An extension of classical transformational grammar. Mimeo, Univ. of Massachusetts, Amherst, MA.
- Bartsch, R. 1976 *The Grammar of Adverbials*. North-Holland, Amsterdam.
- Bennett, M. 1974 Some extensions of a Montague fragment of English. Ph.D. dissertation, UCLA; available from the Indiana University Linguistics Club.
- Brame, M.K. 1978 *Base Generated Syntax*. Noit Amrofer, Seattle, WA.
- Bresnan, J.W. 1976 On the form and functioning of transformations. *Linguistic Inquiry* 7 3-40.
- Bresnan, J.W. 1978 A realistic transformational grammar. In Halle, M., Bresnan, J.W., and Miller, G.A., Ed., *Linguistic Theory and Psychological Reality* MIT Press, Cambridge, MA.
- Chierchia, G. 1981 Nominalization and Montague grammar. A semantics without types for natural languages. MS, Dept. of Linguistics, Univ. of Massachusetts, Amherst, MA.
- Church, A. 1941 *The Calculi of Lambda Conversion*. Princeton Univ. Press, Princeton, NJ.
- Cooper, R., and Parsons, T. 1976 Montague grammar, generative semantics and interpretive semantics. In Partee 1976a, 311-362.
- Craig, J.A., Berezner, S.C., Carney, H.C., and Longyear, C.R. 1966 DEACON: Direct English access and control. Fall Joint Comp. Conf., Nov. 7-10, San Francisco, CA, *AFIPS Conf. Proc.* vol. 29. Spartan Books, Washington, D.C.: 365-380.
- Cresswell, M.J. 1973 *Logics and Languages*. Methuen, London.
- Dowty, D.R. 1978 A guide to Montague's PTQ. Indiana Univ. Linguistics Club, Bloomington, IN.
- Dowty, D.R. 1979 *Word Meaning and Montague Grammar: The Semantics of Verbs and Times in Generative Semantics and in Montague's PTQ*. D. Reidel, Dordrecht.
- Dowty, D.R., Wall, R., and Peters, S. 1981. *An Introduction to Montague Semantics*. D. Reidel, Dordrecht.
- Fodor, J.A. 1978 Propositional attitudes. *The Monist* 61, 501-523.
- Gawron, J.M., King, J.J., Lamping, J., Loebner, E.E., Paulson, E. A., Pullum, G.K., Sag, I.A., and Wasow, T.A. 1982 Processing English with a generalized phrase structure grammar. CLS-82-5, Comp. Science Lab. Tech. Note Series, Hewlett Packard, Palo Alto, CA. Presented at the 20th Ann. Meet. of the Assoc. for Computational Linguistics, June 16-18, Univ. of Toronto, Toronto, Ont.
- Gazdar, G. 1980a A phrase structure syntax for comparative clauses. *Glott-Lexical Grammar* 165-179.
- Gazdar, G. 1980b A cross-categorical semantics for coordination. *Linguistics and Philosophy* 3, 407-409.
- Gazdar, G. 1981a Phrase structure grammar. To appear in Jacobson, P. and Pullum, G.K., Ed., *The Nature of Syntactic Representation*. D. Reidel, Dordrecht.
- Gazdar, G. 1981b Unbounded dependencies and coordinate structure. *Linguistic Inquiry* 12.2.
- Gazdar, G., Klein, E., Pullum, G. K., and Sag, I. to appear *English Syntax*.
- Gazdar, G., Pullum, G.K., and Sag, I. 1980 A phrase structure grammar of the English auxiliary system. Unpublished paper. A slightly different version entitled "Auxiliaries and related phenomena in a restricted theory of grammar" is available from Indiana Univ. Linguistics Club; to appear as "Auxiliaries and related phenomena" in *Language*.
- Gazdar, G. and Sag, I. 1980 Passive and reflexives in phrase structure grammar. In Groenendijk, J., Janssen, T., and Stokhof, M., Ed., *Formal Methods in the Study of Language, Proc. 3rd Amsterdam Coll., March 25-28, 1980*. Mathematical Centre Tracts, Amsterdam.
- Goebel, R. 1982 Forthcoming Ph.D. thesis, Dept. of Computer Science, Univ. of British Columbia, Vancouver, B.C.
- Grice, H.P. 1975 Logic and conversation. In Davidson, D. and Harman, G., Ed., *The Logic of Grammar*. Dickenson, Encino, CA: 64-75.
- Guenther, F. 1978 Systems of intensional logic and the semantics of natural language. In Guenther, F. and Rohrer, C., Ed., *Studies in Formal Semantics*. North-Holland, Amsterdam: 41-74.
- Harman, G. 1975 Logical form. In Davidson, D. and Harman, G., Ed., *The Logic of Grammar*. Dickenson, Encino, CA: 289-307.
- Hughes, G.E. and Cresswell, M.J. 1968 *An Introduction to Modal Logic*. Methuen, London.
- Jackendoff, R. 1977  $\bar{X}$  *Syntax: A Study of Phrase Structure*. MIT Press, Cambridge, MA.
- Kaplan, R.M. (panelist) 1981 A view of parsing. *Proc. 19th Ann. Meet. of the Assoc. for Computational Linguistics, June 29 - July 1*. Stanford Univ., Stanford, CA, 103-104.
- Katz, J., and Fodor, J.A. 1963 The structure of a semantic theory. *Language* 39, 170-210.
- Knuth, D.E. 1968 Semantics of context-free languages. *Mathematical Systems Theory* 2, 127-145.
- Lakoff, G. 1971 On generative semantics. In Steinberg, D.D. and Jakobvitz, L.A., Ed., *Semantics: An Interdisciplinary Reader in Philosophy, Linguistics and Psychology*. Cambridge Univ. Press, New York: 232-296.
- Langendoen, T. 1979 On the assignment of constituent structures to the sentences generated by a transformational grammar. *City Univ. of New York Forum*. New York, NY.
- Lapointe, S. 1977 Recursiveness and deletion. *Linguistic Analysis* 3, 227-266.
- McCarthy, J. 1979 First-order theories of individual concepts and propositions. In Michie, D., Ed., *Expert Systems in the Micro Electronic Age*. Edinburgh Univ. Press, Edinburgh: 271-287.
- McCord, M.C. 1981 Focalizers, the scoping problem, and semantic interpretation rules in logic grammars. Tech. Rep. No. 81-81. Univ. of Kentucky, Lexington, KY. To appear in *Proc. of the Int. Workshop on Logic Programming for Expert Systems, Logicon*, Woodland Hills, CA, Aug. 1981.
- McCord, M.C. 1982 Using slots and modifiers in logic grammars for natural language. *Artificial Intelligence* 18, 327-367.
- Montague, R. 1970a English as a formal language. In Thomason 1974a, 188-221.
- Montague, R. 1970b Universal grammar. In Thomason 1974a, 222-246.
- Montague, R. 1970c The proper treatment of quantification in ordinary English. In Thomason 1974a, 247-270.

- Partee, B.H. 1974 Opacity and scope. In Munitz, M.K., and Unger, P.K., Ed., *Semantics and Philosophy*. New York Univ. Press, New York: 81-101.
- Partee, B.H., Ed. 1976a *Montague Grammar*. Academic Press, New York.
- Partee, B.H. 1976b Some transformational extensions of Montague grammar. In Partee 1976a, 51-76.
- Peters, P.S., and Ritchie, R.W. 1969 Context-sensitive immediate constituent analysis: context-free languages revisited. In *First Ann. Symp. on Theory of Computing*, ACM, New York, 1-8. Also in *Math. Systems Theory* 6 (1973) 324-333.
- Quine, W.v.O. 1960 *Word and Object*. MIT Press, Cambridge, MA.
- Robinson, J.J. (chr.) 1981 Panel: perspectives on parsing issues. *Proc. 19th Ann. Meet. of the Assoc. for Computational Linguistics*, June 29-July 1, Stanford Univ., Stanford, CA, 95-106.
- Rosenschein, S.J., and Shieber, S.M. 1982 Translating English into logical form. Presented at the 20th Ann. Meet. of the Assoc. for Computational Linguistics, June 16-18, Univ. of Toronto, Toronto, Ont.
- Sag, I. 1980 A semantic theory of NP-movement dependencies. To appear in Jacobson, P. and Pullum, G.K., Ed., *The Nature of Syntactic Representation*. D. Reidel, Dordrecht.
- Schank, R.C. 1973 Identification of conceptualizations underlying natural language. In Schank, R.C. and Colby, K.M., Ed., *Computer Models of Thought and Language*. W.H. Freeman, San Francisco: 187-247.
- Schank, R.C. 1975 The primitive ACTs of conceptual dependency. In *Advance Papers of Theoretical Issues in Natural Language Processing Workshop*, June 10-13, MIT, Cambridge, MA, 34-37.
- Schoenfinkel, M. 1924 Ueber die Bausteine der mathematischen Logik. *Math. Annalen* 92, 305-316.
- Schubert, L.K. 1976 Extending the expressive power of semantic networks. *Artificial Intelligence* 7, 163-198.
- Schubert, L.K. 1982 An approach to the syntax and semantics of affixes in 'conventionalized' phrase structure grammar. *Proc. of the 4th Biennial Conf. of the Can. Soc. for Computational Studies of Intelligence (CSCSI/SCEIO)*, 17-19 May 1982, Univ. of Saskatchewan, Saskatoon, Sask.: 189-195.
- Schwind, C. 1978a A formalism for the description of question answering systems. In Bolc, L., Ed., *Natural Language Communication with Computers*. Springer-Verlag, Berlin, Heidelberg & New York: 1-48.
- Schwind, C. 1978b The translation of natural language texts into state logic formulae. Tech. Rep. TUM-INFO-7806, Technische Univ. Muenchen, available from Bibliothek des Fachbereichs Mathematik, Technische Univ. Muenchen, D-8000 Muenchen 2, W. Germany.
- Siegel, M.E.A. 1979 Measure adjectives in Montague grammar. In Davis, S., and Mithun, M., Eds., *Linguistics, Philosophy, and Montague Grammar*. Univ. of Texas Press, Austin, TX: 223-262.
- Thomason, R.H. ed. 1974a *Formal Philosophy: Selected Papers of Richard Montague*. Yale Univ. Press, New Haven, CT.
- Thomason, R.H. 1974b Introduction to Thomason 1974a, 1-69.
- Thomason, R.H. 1980 A model theory for propositional attitudes. *Linguistics and Philosophy* 4, 47-70.
- Thompson, F.B. 1966 English for the computer. Fall Joint Comp. Conf., Nov. 7-10, San Francisco, CA. *AFIPS Conf. Proc. Vol. 29*, Spartan Books, Washington, D.C., 349-356.
- Thompson, H. 1981 Chart parsing and rule schemata in PSG. *Proc. 19th Ann. Meet. of the Assoc. for Computational Linguistics*, June 29-July 1, Stanford Univ., Stanford, CA: 167-172.
- Wilks, Y. 1974 An artificial intelligence approach to machine translation. In Schank, R.C., and Colby, K.M. Ed., *Computer Models of Thought and Language*. W.H. Freeman, San Francisco: 114-151.
- Winograd, T. 1972 *Understanding Natural Language*. Academic Press, New York.
- Woods, W.A. 1977 Lunar rocks in natural English: Explorations in natural language question answerIng. In Zampolli, A., Ed., *Linguistic Structures Processing*. North-Holland, Amsterdam: 521-569.

*Lenhart K. Schubert is an associate professor of computer science at the University of Alberta, Edmonton. He received the Ph.D. degree in computer science from the University of Toronto.*

*Francis Jeffry Pelletier is a professor of philosophy at the University of Alberta. He received the Ph.D. degree from the University of California at Los Angeles.*