A Computational Semantics for Natural Language

Lewis G. Creary and Carl J. Pollard
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304, USA

Abstract

In the new Head-driven Phrase Structure Grammar (HPSG) language processing system that is currently under development at Hewlett-Packard Laboratories, the Montagovian semantics of the earlier GPSG system (see [Gawron et al. 1982]) is replaced by a radically different approach with a number of distinct advantages. In place of the lambda calculus and standard first-order logic, our medium of conceptual representation is a new logical formalism called NFLT (Neo-Fregean Language of Thought); compositional semantics is effected, not by schematic lambda expressions, but by LISP procedures that operate on NFLT expressions to produce new expressions. NFLT has a number of features that make it well-suited for natural language translations, including predicates of variable arity in which explicitly marked situational roles supercede order-coded argument positions, sortally restricted quantification, a compositional (but nonextensional) semantics that handles causal contexts, and a principled conceptual raising mechanism that we expect to lead to a computationally tractable account of propositional attitudes. The use of semantically compositional LISP procedures in place of lambda-schemas allows us to produce fully reduced translations on the fly, with no need for post-processing. This approach should simplify the task of using semantic information (such as sortal incompatibilities) to eliminate bad parse paths.

1. Introduction

Someone who knows a natural language is able to use utterances of certain types to give and receive information about the world. How can we explain this? We take as our point of departure the assumption that members of a language community share a certain mental system — a *grammar* — that mediates the correspondence between utterance types and other things in the world, such as individuals, relations, and states of affairs; to a large degree, this system *is* the language. According to the relation theory of meaning (Barwise & Perry [1983]), linguistic meaning is a relation between types of utterance events and other aspects of objective reality. We accept this view of linguistic meaning, but unlike Barwise and Perry we focus on how the meaning relation is mediated by the intersubjective psychological system of grammar.

In our view, a computational semantics for a natural language has three essential components:

a. a system of conceptual representation for internal use as a computational medium in processes of information retrieval, inference, planning, etc.

b. a system of linkages between expressions of the natural language and those of the conceptual representation, and

c. a system of linkages between expressions in the conceptual representation and objects, relations, and states of affairs in the external world.

In this paper, we shall concentrate almost exclusively on the first two components. We shall sketch our ontological commitments, describe our internal representation language, explain how our grammar (and our computer implementation) makes the connection between English and the internal representations, and finally indicate the present status and future directions of our research.

Our internal representation language, NFLT, is due to Creary [1983]. The grammatical theory in which the present research is couched is the theory of *head grammar* (HG) set forth in [Pollard 1984] and [Pollard forthcoming] and implemented as the front end of the HPSG (Head-driven Phrase Structure Grammar) system, an English language database query system under development at Hewlett-Packard Laboratories. The non-semantic aspects of the implementation are described in [Flickinger, Pollard, & Wasow 1985] and [Proudian & Pollard 1985].

2. Ontological Assumptions

To get started, we make the following assumptions about what categories of things are in the world.

a. There are *individuals*. These include objects of the usual kind (such as Ron and Nancy) as well as *situations*. Situations comprise states (such as Ron's being tall) and events (such as Ron giving his inaugural address on January 21, 1985).

b. There are *relations* (subsuming *properties*). Examples are COOKIE (= the property of being a cookie) and BUY (= the relation which Nancy has to the cookies she buys). Associated with each relation is a characteristic set of *roles* appropriate to that relation (such as AGENT, PATIENT, LOCATION, etc.) which can be filled by individuals. Simple situations consist of individuals playing roles in relations.

Unlike properties and relations in situation semantics [Barwise & Perry 1983], our relations do not have fixed arity (number of arguments). This is made possible by taking explicit account of roles, and has important linguistic consequences. Also there is no distinguished ontological category of locations; instead, the location of an event is just the individual that fills the LOCATION role.

c. Some relations are sortal relations, or *sorts*. Associated with each sort (but not with any non-sortal relation) is a criterion of identity for individuals of that sort [Cocchiarella 1977, Gupta 1980]. Predicates denoting sorts occur in the restrictor-clauses of quantifiers (see section 4.2 below), and the associated criteria of identity are essential to determining the truth values of quantified assertions.

Two important sorts of situations are *states* and *events*. One can characterize a wide range of subsorts of these (which we shall call *situation types*) by specifying a particular configuration of relation, individuals, and roles. For example, one might consider the sort of event in which Ron kisses Nancy in the Oval Office, i.e. in which the relation is KISS, Ron plays the AGENT role, Nancy plays the PATIENT role, and the Oval Office plays the LOCATION role. One might also consider the sort of state in which Ron is a person, i.e. in which the relation is PERSON, and Ron plays the INSTANCE role. We assume that the INSTANCE role is appropriate only for sortal relations.

d. There are *concepts*, both subjective and objective. Some individuals are information-processing *organisms* that use complex symbolic objects (subjective concepts) as computational media for information storage and retrieval, inference, planning, etc. An example is Ron's internal representation of the property COOKIE. This representation in turn is a token of a certain abstract type ↑COOKIE, an *objective* concept which is shared by the vast majority of speakers of English.[1] Note that the objective concept ↑COOKIE, the property COOKIE, and the extension of that property (i.e. the set of all cookies) are three distinct things that play three different roles in the semantics of the English noun *cookie*.

e. There are computational processes in organisms for manipulating concepts e.g. methods for constructing complex concepts from simpler ones, inferencing mechanisms, etc. Concepts of situations are called *propositions*; organisms use inferencing mechanisms to derive new propositions from old. To the extent that concepts are accurate representations of existing things and the relations in which they stand, organisms can contain information. We call the system of objective concepts and concept-manipulating mechanisms instantiated in an organism its *conceptual system*. Communities of organisms can share the same conceptual system.

f. Communities of organisms whose common conceptual system contains a subsystem of a certain kind called a *grammar* can communicate with each other. Roughly, grammars are conceptual subsystems that mediate between events of a specific type (called utterances) and other aspects of reality. Grammars enable organisms to use utterances to give and receive information about the world. This is the subject of sections 4–6.

3. The Internal Representation Language: NFLT

The translation of input sentences into a logical formalism of some kind is a fairly standard feature of computer systems for natural-language understanding, and one which is shared by the HPSG system. A *distinctive* feature of this system, however, is the particular logical formalism involved, which is called NFLT (Neo-Fregean Language of Thought).[2] This is a new logical language that is being developed to serve as the internal representation medium in computer agents with natural language capabilities. The language is the result of augmenting and partially reinterpreting the standard predicate calculus formalism in several ways, some of which will be described very briefly in this section. Historically, the predicate calculus was developed by mathematical logicians as an explication of the logic of mathematical proofs, in order to throw light on the nature of purely mathematical concepts and knowledge. Since many basic concepts that are commonplace in natural language (including concepts of belief, desire, intention, temporal change, causality, subjunctive conditionality, etc.) play no role in pure mathematics, we should not be especially surprised to find that the predicate calculus requires supplementation in order to represent adequately and naturally information involving these concepts. The belief that such supplementation is needed has led to the design of NFLT.

While NFLT is much closer semantically to natural language than is the standard predicate calculus, and is to some extent inspired by psychologistic considerations, it is nevertheless a formal logic admitting of a mathematically precise semantics. The intended semantics incorporates a Fregean distinction between sense and denotation, associated principles of compositionality, and a somewhat non-Fregean theory of situations or situation-types as the denotations of sentential formulas.

3.1. Predicates of Variable Arity

Atomic formulas in NFLT have an explicit role-marker for each argument; in this respect NFLT resembles semantic network formalisms and differs from standard predicate

[1] We regard this notion of *objective concept* as the appropriate basis on which to reconstruct, in terms of information processing, Saussure's notions of *signifiant* (signifier) and *signifié* (signified) [1916], as well as Frege's notion of *Sinn* (sense, connotation) [1892].

[2] The formalism is called "neo-Fregean" because it incorporates many of the semantic ideas of Gottlob Frege, though it also departs from Frege's ideas in several significant ways. It is called a "language of thought" because unlike English, which is first and foremost a medium of *communication*, NFLT is designed to serve as a medium of *reasoning* in computer problem-solving systems, which we regard for theoretical purposes as thinking organisms. (Frege referred to his own logical formalism, *Begriffsschrift*, as a "formula language for pure thought" [Frege 1879, title and p. 6 (translation)]).

calculus, in which the roles are order-coded. This explicit representation of roles permits each predicate-symbol in NFLT to take a variable number of arguments, which in turn makes it possible to represent occurrences of the same verb with the same predicate-symbol, despite differences in valence (i.e. number and identity of attached complements and adjuncts). This clears up a host of problems that arise in theoretical frameworks (such as Montague semantics and situation semantics) that depend on fixed-arity relations (see [Carlson forthcoming] and [Dowty 1982] for discussion). In particular, new roles (corresponding to adjuncts or optional complements in natural language) can be added as required, and there is no need for explicit existential quantification over "missing arguments".

Atomic formulas in NFLT are compounded of a base-predicate and a set of rolemark-argument pairs, as in the following example:

(1a) *English:*

Ron kissed Nancy in the Oval Office on April 1, 1985.

(1b) *NFLT Internal Syntax:*

```
(kiss (agent . ron)
      (patient . nancy)
      (location . oval-office)
      (time . 4-1-85) )
```

(1c) *NFLT Display Syntax:*

```
(KISS  agt:RON
      ptnt:NANCY
       loc:OVAL-OFFICE
       att:4-1-85)
```

The base-predicate '**KISS**' takes a variable number of arguments, depending on the needs of a particular context. In the display syntax, the arguments are explicitly introduced by abbreviated lowercase role markers.

3.2. Sortal Quantification

Quantificational expressions in NFLT differ from those in predicate calculus by always containing a restrictor-clause consisting of a sortal predication, in addition to the usual scope-clause, as in the following example:

(2a) *English:*

Ron ate a cookie in the Oval Office.

(2b) *NFLT Display Syntax:*

```
(SOME X5
  (COOKIE inst:X5)
  (EAT agt:RON ptnt:X5
               loc:OVAL-OFFICE))
```

Note that we always quantify over instances of a sort, i.e. the quantified variable fills the *instance* role in the restrictor-clause.

This style of quantifier is superior in several ways to that of the predicate calculus for the purposes of representing commonsense knowledge. It is intuitively more natural, since it follows the quantificational pattern of English. More importantly, it is more general, being sufficient to handle a number of natural language determiners such as *many*, *most*, *few*, etc., that cannot be represented using only the unrestricted quantification of standard predicate calculus (see [Wallace 1965], [Barwise & Cooper 1981]). Finally, information carried by the sortal predicates in quantifiers (namely, criteria of identity for things of the various sorts in question) provides a sound semantic basis for counting the members of extensions of such predicates (see section 2, assumption c above).

Any internal structure which a variable may have is irrelevant to its function as a uniquely identifiable placeholder in a formula. In particular, a quantified formula can itself serve as its own "bound variable". This is how quantifiers are actually implemented in the HPSG system; in the internal (i.e. implementation) syntax for quantified NFLT-formulas, bound variables of the usual sort are dispensed with in favor of pointers to the relevant quantified formulas. Thus, of the three occurrences of **X5** in the display-formula (2b), the first has no counterpart in the internal syntax, while the last two correspond internally to LISP pointers back to the data structure that implements (2b). This method of implementing quantification has some important advantages. First, it eliminates the technical problems of variable clash that arise in conventional treatments. There are no "alphabetic variants", just structurally equivalent concept tokens. Secondly, each occurrence of a quantified "bound variable" provides direct computational access to the determiner, restrictor-clause, and scope-clause with which it is associated.

A special class of quantificational expressions, called *quantifier* expressions, have no scope-clause. An example is:

(3) *NFLT Display Syntax:*

```
(SOME X1 (COOKIE inst: x1))
```

Such expressions translate quantified noun phrases in English, e.g. *a cookie*.

3.3. Causal Relations and Non-Extensionality

According to the standard semantics for the predicate calculus, predicate symbols denote the extensions of relations (i.e. sets of ordered n-tuples) and sentential formulas denote truth values. By contrast, we propose a *non-extensional* semantics for NFLT: we take predicate symbols to denote relations themselves (rather than their extensions), and sentential formulas to denote situations or situation types (rather than the corresponding truth values).[3] The motivation for this is to provide for the expression of propositions involving causal relations among situations, as in the following example:

[3] The distinction between situations and situation types corresponds roughly to the finite/infinitive distinction in natural language. For discussion of this within the framework of situation semantics, see [Cooper 1984].

(4a) *English:*

John has brown eyes because he is of genotype XYZW.

(4b) *NFLT Display Syntax:*

```
(CAUSE
     conditn:(GENOTYPE-XYZW inst:JOHN)
      result:(BROWN-EYED bearer:JOHN) )
```

Now, the predicate calculus is an extensional language in the sense that the replacement of categorical subparts within an expression by new subparts having the same extension must preserve the extension of the original expression. Such replacements within a *sentential* expression must preserve the *truth-value* of the expression, since the extension of a sentence is a truth-value. NFLT is not extensional in this sense. In particular, some of its predicate-symbols may denote causal relations among situations, and extension-preserving substitutions within causal contexts do not generally preserve the causal relations. Suppose, for example, that the formula (4b) is true. While the extension of the NFLT-predicate '**GENOTYPE-XYZW**' is the set of animals of genotype XYZW, its *denotation* is not this set, but rather what Putnam [1969] would call a "physical property", the property of having the genotype XYZW. As noted above (section 2, assumption d) a property is to be distinguished both from the set of objects of which it holds and from any *concept* of it. Now even if this property were to happen by coincidence to have the same extension as the property of being a citizen of Palo Alto born precisely at noon on 1 April 1956, the substitution of a predicate-symbol denoting this latter property for '**GENOTYPE-XYZW**' in the formula (4b) would produce a falsehood.

However, NFLT's lack of extensionality does not involve any departure from compositional semantics. The *denotation* of an NFLT-predicate-symbol is a property; thus, although the substitution discussed earlier preserves the *extension* of '**GENOTYPE-XYZW**', it does *not* preserve the *denotation* of that predicate-symbol. Similarly, the denotation of an NFLT-sentence is a *situation* or *situation-type*, as distinguished both from a mere truth-value and from a proposition.[4] Then, although NFLT is not an extensional language in the standard sense, a Fregean analogue of the principle of extensionality does hold for it: The replacement of subparts within an expression by new subparts having the same denotation must preserve the denotation of the original expression (see [Frege 1892]). Moreover, such replacements within an NFLT-sentence must preserve the truth-value of that sentence, since the truth-value is determined by the denotation.

3.4. Intentionality and Conceptual Raising

The NFLT notation for representing information about propositional attitudes is an improved version of the neo-Fregean scheme described in [Creary 1979], section 2, which is itself an extension and improvement of that found in [McCarthy 1979]. The basic idea underlying this scheme is that propositional attitudes are relations between people (or other intelligent organisms) and propositions; both terms of such relations are taken as members of the domain of discourse. Objective propositions and their component objective concepts are regarded as abstract entities, roughly on a par with numbers, sets, etc. They are person-independent components of situations involving belief, knowledge, desire, and the like. More specifically, objective concepts are abstract *types* which may have as *tokens* the subjective concepts of individual organisms, which in turn are configurations of information and associated procedures in various individual memories (cf. section 2, assumption d above).

Unlike Montague semantics [Montague 1973], the semantic theory underlying NFLT does *not* imply that an organism necessarily believes all the logical equivalents of a proposition it believes. This is because distinct propositions have as tokens distinct subjective concepts, even if they necessarily have the same truth-value.

Here is an example of the use of NFLT to represent information concerning propositional attitudes:

(5a) *English:*

Nancy wants to tickle Ron.

(5b) *NFLT Display Syntax:*

```
(WANT appr:NANCY
      prop:↑(TICKLE agt:I ptnt:RON))
```

In a Fregean spirit, we assign to each categorematic expression of NFLT both a *sense* and a *denotation*. For example, the denotation of the predicate-constant '**COOKIE**' is the property COOKIE, while the sense of that constant is a certain objective concept – the "standard public" concept of a cookie. We say that '**COOKIE**' *expresses* its sense and *denotes* its denotation. The result of appending the "conceptual raising" symbol ' ↑' to the constant '**COOKIE**' is a new constant, ' ↑**COOKIE**', that *denotes* the concept that '**COOKIE**' expresses (i.e. ' ↑' applies to a constant and forms a standard name of the *sense* of that constant). By appending multiple occurrences of ' ↑' to constants, we obtain new constants that denote concepts of concepts, concepts of concepts of concepts, etc.[5]

In expression (5b), ' ↑' is not explicitly appended to a constant, but instead is prefixed to a compound expression. When used in this way, ' ↑' functions as a syncategorematic operator that "conceptually raises" each categorematic constant within its scope and forms a term incorporating the raised constants and denoting a proposition.

[4] Thus, something similar to what Barwise and Perry call "situation semantics" [1983] is to be provided for NFLT-expressions, *insofar as those expressions involve no ascription of propositional attitudes* (the Barwise-Perry semantics for ascriptions of propositional attitudes takes a quite different approach from that to be described for NFLT in the next section).

[5] For further details concerning this Fregean conceptual hierarchy, see [Creary 1979], sections 2.2 and 2.3.1. Capitalization, '$'-postfixing, and braces are used there to do the work done here by the symbol ' ↑'.

Thus, the subformula '↑**(TICKLE agt:I ptnt:RON)**' is the name of a proposition whose component concepts are the relation-concept ↑**TICKLE** and the individual concepts ↑**I** and ↑**RON**. This proposition is the sense of the unraised subformula '**(TICKLE agt:I ptnt:RON)**'.

The individual concept ↑**I**, the minimal concept of self, is an especially interesting objective concept. We assume that for each sufficiently self-conscious and active organism X, X's minimal internal representation of itself is a token of ↑I. This concept is the sense of the indexical pronoun *I*, and is itself indexical in the sense that what it is a concept *of* is determined not by its content (which is the same for each token), but rather by the context of its use. The content of this concept is partly descriptive but mostly procedural, consisting mainly of the unique and important role that it plays in the information-processing of the organisms that have it.

4. Lexicon

HPSG's head grammar takes as its point of departure Saussure's [1916] notion of a *sign*. A sign is a conceptual object, shared by a group of organisms, which consists of two associated concepts that we call (by a conventional abuse of language) a *phonological representation* and a *semantic representation*. For example, members of the English-speaking community share a sign which consists of an internal representation of the utterance-type /kUki/ together with an internal representation of the property of being a cookie. In a computer implementation, we model such a conceptual object with a data object of this form:

(6) {*cookie* ; **COOKIE**}

Here the symbol '*cookie*' is a surrogate for a phonological representation (in fact we ignore phonology altogether and deal only with typewritten English input). The symbol '**COOKIE**' (a basic constant of NFLT denoting the property COOKIE) models the corresponding semantic representation. We call a data object such as (6) a *lexical entry*.

Of course there must be more to a language than simple signs like (6). Words and phrases of certain kinds can characteristically combine with certain other kinds of phrases to form longer expressions that can convey information about the world. Correspondingly, we assume that a grammar contains in addition to a lexicon a set of *grammatical rules* (see next section) for combining simple signs to produce new signs which pair longer English expressions with more complex NFLT translations. For rules to work, each sign must contain information about how it figures in the rules. We call this information the *(syntactic) category* of the sign. Following established practice, we encode categories as specifications of *values* for a finite set of *features*. Augmented with such information, lexical signs assume forms such as these:

(7a) {*cookie* ; **COOKIE**; [MAJOR: N; AGR: 3RDSG]}

(7b) {*kisses* ; **KISS**; [MAJOR: V; VFORM: FIN]}

Such features as MAJOR (major category), AGR (agreement), and VFORM (verb form) encode inherent syntactic properties of signs.

Still more information is required, however. Certain expressions (*heads*) characteristically combine with other expressions of specified categories (*complements*) to form larger expressions. (For the time being we ignore optional elements, called *adjuncts*.) This is the linguistic notion of *subcategorization*. For example, the English verb *touches* subcategorizes for two NP's, of which one must be third-person-singular. We encode subcategorization information as the value of a feature called SUBCAT. Thus the value of the SUBCAT feature is a sequence of categories. (Such features, called *stack-valued* features, play a central role in the HG account of *binding*. See [Pollard forthcoming].) Augmented with its SUBCAT feature, the lexical sign (2b) takes the form:

(8) {*kisses* ; **KISS**; [MAJOR: V; VFORM: FIN]
SUBCAT: NP, NP-3RDSG}

(Symbols like 'NP' and 'NP-3RDSG' are shorthand for certain sets of feature specifications). For ease of reference, we use traditional grammatical relation names for complements. Modifying the usage of Dowty [1982], we designate them (in *reverse* of the order that they appear in SUBCAT) as *subject*, *direct object*, *indirect object*, and *oblique objects*. (Under this definition, determiners count as subjects of the nouns they combine with.) Complements that themselves subcategorize for a complement fall outside this hierarchy and are called *controlled* complements. The complement next in sequence after a controlled complement is called its *controller*.

For the sign (8) to play a communicative role, one additional kind of information is needed. Typically, heads give information about relations, while complements give information about the roles that individuals play in those relations. Thus lexical signs must assign roles to their complements. Augmented with role-assignment information, the lexical sign (8) takes the form:

(9) {*kisses* ; **KISS**; [MAJOR: V: VFORM: FIN]
SUBCAT: ⟨NP, **patient**⟩,
⟨NP-3RDSG, **agent**⟩}

Thus (9) assigns the roles AGENT and PATIENT to the subject and direct object respectively. (Note: we assume that nouns subcategorize for a determiner complement and assign it the instance role. See section 6 below.)

5. Grammatical Rules

In addition to the lexicon, the grammar must contain mechanisms for constructing more complex signs that mediate between longer English expressions and more complex NFLT translations. Such mechanisms are called *grammatical rules*. From a purely syntactic point of view, rules can be regarded as ordering principles. For example, English grammar has a rule something like this:

(10) If X is a sign whose SUBCAT value contains just one category Y, and Z is a sign whose category is consistent with Y, then X and Z can be combined to form a new sign W whose expression is got by

concatenating the expressions of X and Z.

That is, put the final complement (subject) to the left of the head. We write this rule in the abbreviated form:

(11) -> C H [Condition: length of SUBCAT of H = 1]

The form of (11) is analogous to conventional phrase structure rules such as NP - > DET N or S - > NP VP; in fact (11) subsumes both of these. However, (11) has no left-hand side. This is because the category of the constructed sign (*mother*) can be computed from the constituent signs (*daughters*) by general principles, as we shall presently show.

Two more rules of English are:

(12) -> H C [Condition: length of SUBCAT of H = 2]

(13) -> H C2 C1
[Condition: length of SUBCAT of H = 3]

(12) says: put a direct object or subject-controlled complement after the head. And (13) says: put an indirect object or object-controlled complement after the direct object. As in (11), the complement signs have to be consistent with the subcategorization specifications on the head. In (13), the indices on the complement symbols correspond to the order of the complement categories in the SUBCAT of the head.

The category and translation of a mother need not be specified by the rule used to construct it. Instead, they are computed from information on the daughters by universal principles that govern rule application. Two such principles are the Head Feature Principle (HFP) (14) and the Subcategorization Principle (15):

(14) Head Feature Principle:
Unless otherwise specified, the head features on a mother coincide with the head features on the head daughter.

(For present purposes, assume the head features are all features except SUBCAT.)

(15) Subcategorization Principle:
The SUBCAT value on the mother is got by deleting from the SUBCAT value on the head daughter those categories corresponding to complement daughters.

(Additional principles not discussed here govern control and binding.) The basic idea is that we start with the head daughter and then process the complement daughters in the order given by the indices on the complement symbols in the rule. So far, we have said nothing about the determination of the mother's *translation*. We turn to this question in the next section.

6. The Semantic Interpretation Principle

Now we can explain how the NFLT-translation of a phrase is computed from the translations of its constituents. The basic idea is that every time we apply a grammar rule, we process the head first and then the complements in the order indicated by the rule (see [Proudian & Pollard 1985]). As each complement is processed, the corresponding category-role pair is popped off the SUBCAT stack of the head; the category information is merged (unified) with the category of the complement, and the role information is used to combine the complement translation with the head translation. We state this formally as:

(16) Semantic Interpretation Principle (SIP):
The translation of the mother is computed by the following program:

a. Initialize the mother's translation to be the head daughter's translation.

b. Cycle through the complement daughters, setting the mother's translation to the result of combining the complement's translation with the mother's translation.

c. Return the mother's translation.

The program given in (16) calls a function whose arguments are a sign (the complement), a rolemark (gotten from the top of the head's SUBCAT stack), and an NFLT expression (the value of the mother translation computed thus far). This function is given in (17). There are two cases to consider, according as the translation of the complement is a determiner or not.

(17) Function for Combining Complements:

a. If the MAJOR feature value of the complement is DET, form the quantifier-expression whose determiner is the complement translation and whose restriction is the mother translation. Then add to the restriction a role link with the indicated rolemark (viz. *instance*) whose argument is a pointer back to that quantifier-expression, and return the resulting quantifier-expression.

b. Otherwise, add to the mother translation a role link with the indicated rolemark whose argument is a pointer to the complement translation (a quantifier-expression or individual constant). If the complement translation is a quantifier-expression, return the quantificational expression formed from that quantifier-expression by letting its scope-clause be the mother translation; if not, return the mother translation.

The first case arises when the head daughter is a noun and the complement is a determiner. Then (17) simply returns a complement like (3). In the second case, there are two subcases according as the complement translation is a quantifier-expression or something else (individual constant, sentential expression, propositional term, etc.) For example, suppose the head is this:

(18) {*jogs* ; **JOG**; [MAJOR: V; VFORM: FIN]
SUBCAT: ⟨NP-3RDSG, **agent**⟩}

If the (subject) complement translation is '**RON**' (*not* a quantifier-expression), the mother translation is just:

(19) **(JOG agt:RON)**;

but if the complement translation is
'**{ALL P3 (PERSON inst:P3)}**'
(a quantifier-expression), the mother translation is:

concatenating the expressions of X and Z.

That is, put the final complement (subject) to the left of the head. We write this rule in the abbreviated form:

(11) -> C H [Condition: length of SUBCAT of H = 1]

The form of (11) is analogous to conventional phrase structure rules such as NP - > DET N or S - > NP VP; in fact (11) subsumes both of these. However, (11) has no left-hand side. This is because the category of the constructed sign (*mother*) can be computed from the constituent signs (*daughters*) by general principles, as we shall presently show.

Two more rules of English are:

(12) -> H C [Condition: length of SUBCAT of H = 2]

(13) -> H C2 C1
[Condition: length of SUBCAT of H = 3]

(12) says: put a direct object or subject-controlled complement after the head. And (13) says: put an indirect object or object-controlled complement after the direct object. As in (11), the complement signs have to be consistent with the subcategorization specifications on the head. In (13), the indices on the complement symbols correspond to the order of the complement categories in the SUBCAT of the head.

The category and translation of a mother need not be specified by the rule used to construct it. Instead, they are computed from information on the daughters by universal principles that govern rule application. Two such principles are the Head Feature Principle (HFP) (14) and the Subcategorization Principle (15):

(14) Head Feature Principle:
Unless otherwise specified, the head features on a mother coincide with the head features on the head daughter.

(For present purposes, assume the head features are all features except SUBCAT.)

(15) Subcategorization Principle:
The SUBCAT value on the mother is got by deleting from the SUBCAT value on the head daughter those categories corresponding to complement daughters.

(Additional principles not discussed here govern control and binding.) The basic idea is that we start with the head daughter and then process the complement daughters in the order given by the indices on the complement symbols in the rule. So far, we have said nothing about the determination of the mother's *translation*. We turn to this question in the next section.

6. The Semantic Interpretation Principle

Now we can explain how the NFLT-translation of a phrase is computed from the translations of its constituents. The basic idea is that every time we apply a grammar rule, we process the head first and then the complements in the order indicated by the rule (see [Proudian & Pollard 1985]). As each complement is processed, the corresponding category-role pair is popped off the SUBCAT stack of the head; the category information is merged (unified) with the category of the complement, and the role information is used to combine the complement translation with the head translation. We state this formally as:

(16) Semantic Interpretation Principle (SIP):
The translation of the mother is computed by the following program:

a. Initialize the mother's translation to be the head daughter's translation.

b. Cycle through the complement daughters, setting the mother's translation to the result of combining the complement's translation with the mother's translation.

c. Return the mother's translation.

The program given in (16) calls a function whose arguments are a sign (the complement), a rolemark (gotten from the top of the head's SUBCAT stack), and an NFLT expression (the value of the mother translation computed thus far). This function is given in (17). There are two cases to consider, according as the translation of the complement is a determiner or not.

(17) Function for Combining Complements:

a. If the MAJOR feature value of the complement is DET, form the quantifier-expression whose determiner is the complement translation and whose restriction is the mother translation. Then add to the restriction a role link with the indicated rolemark (viz. *instance*) whose argument is a pointer back to that quantifier-expression, and return the resulting quantifier-expression.

b. Otherwise, add to the mother translation a role link with the indicated rolemark whose argument is a pointer to the complement translation (a quantifier-expression or individual constant). If the complement translation is a quantifier-expression, return the quantificational expression formed from that quantifier-expression by letting its scope-clause be the mother translation; if not, return the mother translation.

The first case arises when the head daughter is a noun and the complement is a determiner. Then (17) simply returns a complement like (3). In the second case, there are two subcases according as the complement translation is a quantifier-expression or something else (individual constant, sentential expression, propositional term, etc.) For example, suppose the head is this:

(18) {*jogs* ; **JOG**; [MAJOR: V; VFORM: FIN]
SUBCAT: ⟨NP-3RDSG, **agent**⟩}

If the (subject) complement translation is '**RON**' (*not* a quantifier-expression), the mother translation is just:

(19) **(JOG agt:RON)**;

but if the complement translation is
'**(ALL P3 (PERSON inst:P3))**'
(a quantifier-expression), the mother translation is:

son, Yale University Press, New Haven and London, 1974.

Pollard, Carl [1984]. *Generalized Phrase Structure Grammars, Head Grammars, and Natural Language.* Doctoral dissertation, Stanford University.

Pollard, Carl [forthcoming]. "A Semantic Approach to Binding in a Monostratal Theory." To appear in *Linguistics and Philosophy.*

Proudian, Derek, and Carl Pollard [1985]. "Parsing Head-driven Phrase Structure Grammar." *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics.*

Putnam, Hilary [1969]. "On Properties." In *Essays in Honor of Carl G. Hempel*, N. Rescher, ed., D. Reidel, Dordrecht. Reprinted in *Mind, Language, and Reality: Philosophical Papers* (Vol. I, Ch. 19), Cambridge University Press, Cambridge, 1975.

Saussure, Ferdinand de [1916]. *Cours de Linguistique Generale.* Paris: Payot. Translated into English by Wade Baskin as *Course in General Linguistics*, The Philosophical Library, New York, 1959 (paperback edition, McGraw-Hill, New York, 1966).

Wallace, John [1965]. "Sortal Predicates and Quantification." *The Journal of Philosophy* 62, 8–13.