EXTENDED CATEGORIAL GRAMMAR*

Suson Yoo and Kiyong Lee

This work aims at constructing a simple and yet descriptively adequate grammar that best suits a semantic program based on situation theory. For this we propose an extended version of categorial grammar called ECG consisting of feature-based categories and operations of category cancellation. We demonstrate that this version can successfully account for such phenomena as agreement, case, and unbounded dependency in English. We also show that the use of indices in ECG makes it possible to adopt an equational approach to the representation and unification of information contents and thus to obtain the content of a complex experession by solving equations that represent the contents of its constitutent expressions. We, however, understand our proposed grammar ECG as an amalgamation, a successful amalgamation of current linguitic models such as GPSG, HPSG, and some version of categorial grammar.

1. Introduction

The purpose of this work is to construct a simple, elegant, and computationally tractable grammar that suits a semantic program based on Jon Barwise's (1987) situation theory.[1] For such a candidate we propose an extended version of categorial grammar called ECG and demonstrate that,

*This work is an interim report on part of the long-term research project on Natural language processing. It has been partially supported by a grant (No. 860207) from Korea Science and Engineering Foundation. It is also part of the doctoral thesis being developed by the first author. Both of the authors are grateful to Roland Hausser, Lauri Karttunen, Stanley Peters, Gordon Plotkin, Craige Roberts, and Peter Sells for reading and commenting on earlier versions of the present work and for their help and suggestions for improving them. They also wish to thank Center for the Study of Language and Information, Stanford University, for providing them with an ideal research environment. The second author also would like to take this opportunity to express his gratitude to Korea University, the Fulbright Commission, and the Council for International Exchange of Scholars for their support, which has made it possible for him to spend a rewarding year at CSLI, Stanford.

[1] The earliest version, ACG, of the present system has been reframed in Prolog to show its computational tractability. See Lee, Kiyong, Suson Yoo, Key-Sun Choi, and Sangki Han(1986).

with its feature-based categories and with its operations of category cancellation, it can successfully account for such phenomena as agreement, case, and unbounded dependency in English.[2] We also show that the use of indices in ECG makes it possible to obtain the content of a complex expression through a simple process of unifying the contents of its constituent expressions.[3] We understand our proposed grammar ECG as an amalgamation of current linguistic models such as GPSG, HPSG, and some versions of categorial grammar.[4]

2. Syntactic Features and Categories

As in GPSG and HPSG, we define categories as sets of syntactic features, or attribute-value pairs. An NP, for instance, will be defined as consisting of the features *nominal* and *maximal*. Likewise, an expression, e.g. **she,** will be treated as belonging to the NP category whose *agreement* (number and person) attributes are specified as *singular* and *third,* and whose *case* is assigned the value *nominative.* But since, for most categories of natural language, not every feature is specified, categories in general will be defined as partial functions from syntactic attributes to appropriate values.[5]

2.1 Core Attribute

We first introduce the basic attribute *core* whose values are either *n* (for

[2] Unlike some of the recent versions of categorial grammar, ECG does not contain any wrapping operations nor any composition of functional applications.

[3] For an extensive discussion, see Yoo, Suson and Kiyong Lee (1987).

[4] Among them are Hausser's (1986) NEWCAT, Uszkoreit's (1986) CUG, and Zeevat, Klein, and Calder's (1986) UCG. For GPSG, or Generalized Phrase Structure Grammar, refer to Gazdar, Klein, Pullum, and Sag (1985), and for HPSG, or Head-driven Phrase Structure Grammar, to Sag and Pollard (1987). In developing ECG, we have, however, been most indebted to various notions and insights proposed in Sag and Pollard's (1987) HPSG and, as a consequence, ECG may be considered as one of its variations. These two, however, differ from each other at least in one important respect: while the latter is based on the notion of *head,* the former essentially relies on the notion of *quotient,* or *functor,* category. Our efforts, we believe, are also in line with the current research project on unification categorial grammar being carried out at Centre for Cognitive Science, University of Edinburgh.

[5] As will be seen, our treatment of features and categories differs in many significant details from that of GPSG or HPSG. We shall not, however, be making constant references to any of the differences in the present paper.

nominal expressions), *s* (for sentences or statements) or *p* (for prepositional or postpositional phrases) as a basis of defining such categories as N, NP, S, and PP.[6] These categories constitute a core in forming other major syntactic categories.

(1) Basic Categories

N, NP:	{⟨core, n⟩}
S:	{⟨core, s⟩}
PP:	{⟨core, p⟩}

2.2 Boolean Attributes

We introduce a pair of Boolean attributes which take as their value either + or −. The attribute *max* (for maximal expressions) distinguishes between phrasal and non-phrasal categories.[7] Secondly, the attribute *que* (for question words) distinguishes **wh**-words from non-**wh**-words.

(2) Boolean Features

attributes	possible values
max	+, −
que	+, −

These attributes are used to subcategorize basic categories, as illustrated in:

(3) Subcategories

Abbreviations	Categories	Expressions
N	{⟨core,n⟩, ⟨max,−⟩}	man
		bald man

[6] The first two values may be considered as corresponding to the types *e* and *t* in Montague semantics.

[7] The attribute *max* contrasts with the attribute *lexical* in HPSG. First, while HPSG distinguishes a lexical expression such as **man** from a non-lexical expression such as **bald man**, ECG treats them both as belonging to the same non-maximal category {⟨*core,n*⟩, ⟨*max*, −⟩}. Our treatment is based on a semantic intuition that they are of the same semantic type, each describing a property of individuals. Note that, in Montague semantics, these two are also treated as denoting objects of the same type ⟨*e,t*⟩, whereas an NP expression such as **every bald man** is treated as denoting an object of another type ⟨*e*, ⟨*e,t*⟩⟩. Secondly, in ECG, a compound verb, e.g. **look for**, is treated as of a non-maximal category just like a simple verb such as **seek**, for they may both be used to denote the same relation.

NP[+que]	$\{\langle core,n\rangle, \langle max,+\rangle, \langle que,+\rangle\}$	**what** **who**
NP[−que]	$\{\langle core,n\rangle, \langle max,+\rangle, \langle que,-\rangle\}$	**Tom** **every man**

Here we have used two abbreviatory conventions: one is to abbreviate a Boolean feature of the form either $\langle a, +\rangle$ or $\langle a, -\rangle$ as $+a$ or $-a$, respectively: a second is to separate out a particular set of features constituting a category, in particular one containing core-maximality features, from the rest of features by enclosing these remaining features with a pair of square brackets. NP[+que] is thus to be understood as an abbreviation for $\{\langle core,n\rangle, \langle max,+\rangle\} \cup \{\langle que,+\rangle\}$. In general, given an attribute *a*, value *v*, and a category (or a set of attribute-value pairs) *C*, we have:

(4) Abbreviatory Conventions I
[i] $\langle a, v\rangle := va$, where v is either $+$ or $-$.
[ii] $C[\langle a, v\rangle,\ldots] := C \cup \{\langle a, v\rangle,\ldots\}$

2.3 Contextual Attributes

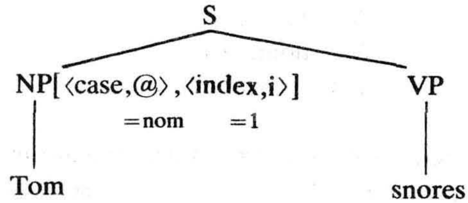
Besides core and Boolean attributes, ECG introduces the contextual attributes *num* (for number), *pers* (for person), *case*, and *index* whose values are contextually determined. The values of these features associated with an NP expression such as **Tom**, for instance, are usually determined when it combines with another expression such as **snores.** The proper noun Tom belongs to the NP whose attributes *num* and *pers* are assigned the values *singular* and *third*, respectively, or simply *sg3*, but whose *case* and *index* values are indeterminate.[8] When combined with the verb **snores** to form the sentence **Tom snores**, the *case* of the NP will be assigned the value *nominative*, and its *index*, the value 1 associated with its argument-role of snoring.[9] On the other hand, if it combines with a transitive verb such as **kiss**, then its *case* will be assigned the value *accusative* and its *index*, the value 2 associated with its argument-role of being kissed.

[8] We shall often treat the pair of *num* and *pers* as a single set of attributes, naming it *agr* (for agreement).

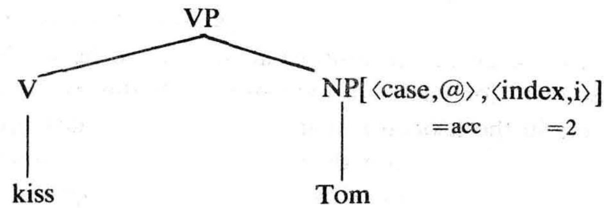
[9] One of the uses of an index is to associate a particular category with its argument-role in a relation, thereby linking syntactic information with semantic information. By convention, we assign 1 to the agent role, 2 to the recipient or patient role, and 3 to the object role.

(5)

a.



b.



Note in the above trees that the NP **Tom** first has the indeterminate values @ and *i* for its attributes *case* and *index*, respectively, and then these indeterminate values are *anchored* to some determinate values, as represented immediately underneath.[10] We shall treat indeterminate values as quasi-real objects called *parametric values*.

Many of the lexical entries, or *lexemes*, will have parametric values for their contextually dependent attributes. Besides the proper noun **Tom** just illustrated, we have nouns such as **fish**, **sheep**, and **deer** each of which has the parametric value ♯ for the attribute *num*, the parametric value @ for its *case*. Even a pronoun has the parametric value *i* for its *index*. Here are some examples:

(6) Lexemes

Tom: NP[⟨num, sg⟩, ⟨pers, 3⟩, ⟨case, @⟩, ⟨index, i⟩]
fish: NP[⟨num, ♯⟩, ⟨pers, 3⟩, ⟨case, @⟩, ⟨index, i⟩]
me: NP[⟨num, sg⟩, ⟨pers, 1⟩, ⟨case, acc⟩, ⟨index, i⟩]

The following table shows what values each contextual attribute may take:

(7) Contextual Features

Attributes	Possible Values

[10] The categories VP and V will be defined presently as *quotient*, or functor categories.

num	# : sg, pl
pers	% : 1, 2, 3
case	@ : nom, acc
index	f, g,...: 1, 2,...f, g,...

Although these parametric values remain indeterminate in the lexicon, they are anchored to determinate values when their associated expressions are used as parts of a larger expression in an actual discourse. For this, with the exception of not assigning an index to expletive NPs, we first require every assignment for each of the contextual features to be totally defined.[11] This guarantees that, for each NP category (with the exception of expletives) which occurs in the lexicon as part of the syntactic description of a lexeme, some value, whether or not determinate, must be assigned to each of its associated contextual features. Secondly, for an appropriate use of any of these lexemes in an actual discourse, we anchor each parametric value of its contextual attributes to some determinate value. For any category *C* [*par*] with a parametric value *par*, we shall call *C* [*par* = *v*] an instantiated category if *v* is a value to which *par* is anchored.

2.4 Forms

Some categories have various distinct forms. Most NP's have a normal form, but, as in GPSG, expletives such as **it** and **there** will be treated as having a special value either *it* or *there* for their *form* and thus as belonging respectively to the categories NP[⟨form, it⟩] or NP[⟨form, there⟩].

The forms of prepositional phrases PP are differentiated by their prepositions. A prepositional phrase **in California,** for instance, is treated as belonging to the category NP[⟨form, *in*⟩].

Associated with the category S are two types of forms: *comp* (for complementizers) and *vform* (for verbal forms).[12] The attribute *comp* takes *that, for, if,* and *whether* as its possible values, while the attribute *vform* takes *fin* (for finite), *bas* (for base), inf (for infinitive), *part* (for participle), and *pass*

[11] Expletives are not assigned an *index* feature because they play no argument roles whatsoever. Here, we list two expletives in English:

it: NP |⟨*num, sg*⟩, ⟨*pers*,3⟩, ⟨*case*,@⟩|

there: NP |⟨*num*, #⟩, ⟨*pers*,3⟩, ⟨case,@⟩| .

[12] Although, in English, a complementizer forms a separate word, it is analyzed only as a morphological part of a (verbal) expression in a language such as Korean or Japanese. We thus assume complementizers to be of the type of (morphological) categorial forms.

(for passive) as its possible values. A sentence **for her to win**, for example, belongs to the category S[⟨*comp, for*⟩, ⟨*vform, inf*⟩].

We shall introduce another set of abbreviatory conventions:

(8) Abbreviatory Conventions II
[i] C [⟨*index, i*⟩] := C_i for any category C.
[ii] ⟨*a, v*⟩: = *v* for any uniquely identifiable attribute *a*.

For example, NP[⟨index, i⟩] can be written simply as NP_i, and NP[⟨case, nom⟩] and NP[⟨num, 3⟩] can be abbreviated as NP[nom] and NP[3], respectively.

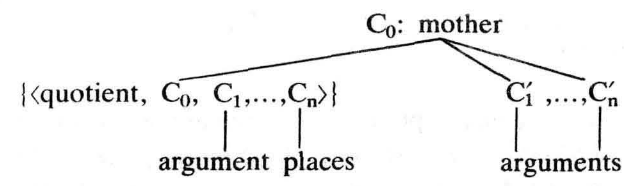
2.5 Category-valued Attributes

All of the above attributes are atomic-valued; no basic, Boolean, contextual, or form attributes take a category as their value. But in ECG, as in GPSG, there are category-valued attributes: *quotient* and *stored.*

2.5.1 Quotient Attribute:

Being a categorial grammar, ECG is characterized by the category-valued attribute *quotient.* This attribute takes as its value a sequence of categories C_0, C_1,...,C_n, where $n>1$, forming a quotient feature ⟨*quotient*, C_0,C_1,...,C_n⟩, where C_0 is interpreted as the resultant (or *mother*) category and the remaining sequence of categories C_1,...,C_n as *argument place* categories. A category containing a *quotient* feature is called a *quotient category.* As will be discussed in the next section on cateogry cancellations, the quotient category may combine an n-sequence of categories called *argument* categories to yield the mother category C_0.[13]

(9)



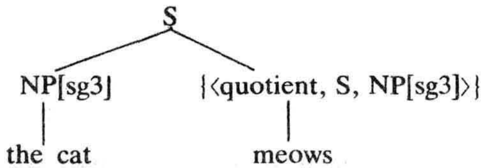
[13] One may think of a quotient category the same as a functor category in classical categorial grammars, but we have avoided using this terminology because, unlike a functor category in Montague semantics, our quotient category is no longer interpreted as denoting an object called a functor.

Here each argument place C_i is required to subsume its corresponding argument C'_j so that, while they are compatible, C'_j may contain more features than C_i.[14]

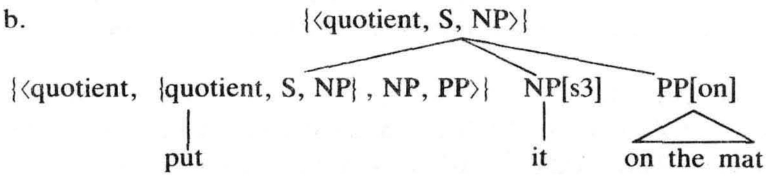
Let's consider two simple examples: the verb **meow** will be treated as of the quotient cateogry $\{\langle \text{quotient}, S, NP\rangle\}$, and the transitive verb **put** of another quotient category $\{\langle \text{quotient}, \{\langle \text{quotient}, S, NP\rangle\}, NP, PP\rangle\}$. The former category may combine with an NP to yield an S, and the latter with a sequence of categories NP, PP to yield the category $\{\langle \text{quotient}, S, NP\rangle\}$.

(10)

a.



b.



Note that, unlike ordinary categorial grammars, ECG allows not only a single category, but also a sequence of categories to combine with a quotient category. As a result, we obtain flatter structures in which a verb, for instance, is represented as taking all of its object arguments as sisters.

By introducing a third set of abbreviatory conventions, we can simplify the representation of quotient categories.

(11) Abbreviatory Conventions III
[i] $\{\langle \text{quotient}, C_0, C_1, \ldots, C_n\rangle\} := (C_0, C_1, \ldots, C_n)$
[ii] VP: $=(\text{S}, \text{NP})$

Corresponding to the argument places of a quotient category are two kinds of arguments, *obligatory* and *optional*. In principle, these argument places are exhaustively listed in the lexicon with some extra feature that distinguishes optionality; here we may place each optional argument place within a pair of curly brackets. This distinction, however, disappears when

[14] The notion of *subsumption* will be defined in section 3.

an optional element is chosen to be used as part of an actual discourse. On the tree or in the process of category cancellations, therefore, there is no distinction of optionality between categories. Here we list some lexemes belonging to quotient categories with the distinction of optionality:

(12) Lexicon
snore: ((S, NP), {...})
love: ((S, NP), NP, {...})
put: (S, NP), NP, PP[loc], {...}), where loc: on, in, ...
get: ((S, NP), NP, {PP[from],...})
a, the: (NP, N, {S/NP} ,...)

Verbal expressions belong to a quotient category. The intransitive verb **snore** is of the quotient category (S, NP) which takes an NP as argument to form an S; the transitive verb **love** takes two NPs as arguments to form an S; and the verbs **put** and **get** take, respectively, two NP's and one PP as arguments to form an S. The determiners **a** and **the** may have two arguments, a common noun N and a relative clause S/NP.[15] Note here the distinction between obligatory and optional arguments, illustrated by our examples **put** and **get**: in the former case, PP is an obligatory argument, or a *complement,* but, in the latter case, it is only an optional argument, or an *adjunct.* A relative clause is also an optional argument of a determiner. The lack of optional arguments in some of the examples, however, does not rule out the possibility of their having optional arguments. For example, verbs in general may have time or place expressions as their adjuncts.

Besides verbal expressions, there are other types of quotient categories, as listed in:

(13) **on**: (PP[on], NP)
tiny: (N, N)
will: /(VP, VP)

The prepositions **on**, **about** and **with** are of the quotient category (PP, NP); the adjectives **tiny**, **former** and **female**, of the quotient category (N, N); and the auxiliary verbs **will**, **be** and **must**, of the quotient cateogry (VP, VP).

2.5.2 Stored Attribute:

The category-valued attribute *stored* is introduced to deal with gaps and

[15] This type of category is called a storage(NP)-carrying category. See the section 5.2 for its definition.

unbounded dependencies associated with such constructions as **wh**-questions, topicalization, and relative clauses in English. It takes an *index*-specified *maximal* category as its possible value. A verb phrase with a feature ⟨*stored*, NP_i [+*que*]⟩, for instance, is interpreted as having a gap to be filled in by a **wh**-phrase of the category NP_i [+*que*]. We shall call a category containing a *stored* feature a *storage* and a category containing a storage a storage-carrying category. It should be noted here that storages never occur by themselves, but only with other categories.[16]

Consider the following construction:

(14) What did Paul get [ ] from Kim?
(15) A kitten, Paul got [ ] from Kim.
(16) The kitten which Paul got [ ] from Kim.

The verb **get** takes two NP's (subject and object) as its obligatory arguments, and a PP as its optional argument, thus belonging to the category {⟨*quotient*, (S,NP_1), NP_2, PP_3[from]⟩}. But note in each of the above examples that the argument place NP_2 has no corresponding category in the argument sequence of **get**: in other words, an object NP corresponding to NP_2 is missing from **Paul get** (or **got**) **from Kim**. To account for such a gap, we treat the quotient category expression **get** as also belonging to a storage-carrying category as follows:

(17) **get**: {⟨*quotient*, (S, NP_1), PP_3[from]⟩, ⟨*stored*, NP_2⟩}

Here the category NP_2 is stored away for a later use. This verb may also belong to another storage-carrying category:

(18) **get**: {⟨*quotient*, (S, NP_1), NP_2⟩, ⟨*stored*, PP_3[*from*]⟩}

This category is needed to account for such cases, where the category PP is missing from the argument sequence of **get**, as in:

(19) From whom did Paul get the kitten [ ] ?

The information that, under certain restrictions, an expression belonging to a quotient category may also belong to a storage-carrying category can

[16] Although the attribute *stored* in ECG resembles the attribute SLASH in GPSG, their differences in use should become apparent as our present discussion proceeds.

be conveyed by the following lexical rule:

(20) Storage Rule
If δ is an expression of a quotient category of the form $\{\langle \text{quotient}, C_0, C_1, \ldots, C_i, \ldots, C_n \rangle\}$, where $n \geq 1$, then it can also be of a storage-carrying category either of the forms:
[i] for $n > 1$,
$\{\langle \text{quotient}, C_0, C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_n \rangle\} \cup \{\langle \text{stored}, C_i \rangle\}$
[ii] for $n = 1$,
$C_0 \cup \{\langle \text{stored}, C_i \rangle\}$
provided C_i is an indexed maximal category.

By this rule, any indexed maximal category occurring in a quotient category may become a storage. For example, the preposition **from** of the quotient category $\{\langle \text{quotient}, PP_i[\text{from}], NP_j[\text{acc}] \rangle\}$ may also belong to the storage-carrying category $\{\langle \text{quotient}, PP_i, \{\langle \text{stored}, NP_j[\text{acc}] \rangle\} \rangle\}$, since this storage NP_j is an indexed maximal category. This categorization then allows a sentence such as (21) to be derived:

(21) Who(m) did Paul get the kitten from [ ] ?

The above sentence is derived by carrying up the storage NP_j to the top of the tree and having it replaced by the question word **whom** at the last stage of derivaiton.

2.6 Summary

The attributes and their respective possible values that are needed to form appropriate categories in ECG can be summarized in the tabular form (22), where, for any sets A and B, the notation $A \rightharpoonup B$ is to be understood as the set of all possible partial functions from A to B.

(22) Categories
Cat : *Attributes* $\rightharpoonup$ [*Atomic Values* $\cup$ *Sequences of Cats*]

core	n,s,p
max	+, −
que	+, −
num	#, sg, pl

pers	1,2,3	
case	@, nom, acc	
index	f,g,h,..,1,2,3...	
comp	for, that, if, whether	
vform	fin, bas, inf, part, ger, pass	
nform	norm, it, there,	
pform	in, of, from,...	
stored		$C_i[+max]$
quotient		$C_0, C_1, \ldots, C_n$ where $n \geq 1$

On the basis of this table, we can formalize the definition of categories. Let A be the set of attributes, and V_p and P be functions from A to sets. Then we define the set Cat_p of lexical categories, or lexemes, as follows:[17]

(23) Definition of Categories

$$Cat_p = \prod_{a \in A} V_p(a) \cup P(a)$$

where, for each attribute $a \in A$,
V_p(a) and P(a) are defined as below:
$V_p(\text{core}) = \{n, s, p\}$
$V_p(\text{max}) = V_p(\text{que}) = \{+, -\}$
$V_p(\text{case}) = \{\text{nom, acc}\}$
...
$V_p(\text{stored}) = Cat_p$
$V_p(\text{quotient}) = \cup_{n \geq 1} Cat_p^n$
P(core) undefined
P(max) undefined
$P(\text{case}) = \{@\}$ ...
P(stored) undefined
P(quotient) undefined

Note, first, that the set Cat_p contains categories some of whose attribute values are parameters. In actual use, however, these parametric values are anchored to specific values by a (partial) function called *Anchor* as defined in:

[17] As already mentioned, we treat a category as a *partial* function from attributes to their possible values, and their set Cat_p as the partial product of such functions. In general the partial product $\prod_{i \in I}$ is to be understood as $\{f: I - \cup_{i \in I} F(i) \mid \forall_{i \in I} f(i) \in F(i)\}$, where I is a set of indices and F is a (partial) function: I — *Sets*.

(24)

$$Anchor = \prod_{a \in A} P(a) \rightarrow V(a)$$

where A and P are the same as those defined in (23) and V is exactly the same as V_p except that $V(\text{stored}) = Cat$ and $V(\text{quotient}) = \cup_{n \geq 1} Cat^n$.

Here the set of non-parametric categories *Cat* is obtained by the instantiation function as follows:

(25) For some function *Instant*,
$Cat = Instant\ (Cat_p \times Anchor)$.

Secondly, besides those restrictions on V_p and P in (23), we need a series of restrictions called *FCR* (for feature co-occurrence restrictions) as part of the definition of categories, as illustrated in:

(26) FCR: feature co-occurrence restrictions
Let f be either V_p or P as defined in (23). Then,
[i] $\exists_v f(\text{case}) = v - [f(\text{core}) = n \wedge f(\text{max}) = +]$
[ii] $\exists_v f(\text{stored}) = v - \exists_u [f(\text{index}) = u \wedge f(\text{max}) = +]$

FCR [i] states that a *case* value is assigned to an NP and only to an NP, and *FCR* [ii] states that only an indexed maximal category can be *stored*. This list is only illustrative and, when necessary, may be expanded.

3. Category Cancellations

The basic syntactic device of classical categorial grammar is functional application. It combines a functor expression with an appropriate argument expression to yield another expression, as is usually stated in the following form.

(1) Functional Application
If δ is an expression of the category (C, A) and α is an expression of the category A, $f(\delta, \alpha)$ is an expression of the category C.[18]

[18] We take the operation f to be a concatenation operation f_{concat}.

There are two basic operations involved in such functional application cancellation and concatenation. An operation of cancellation applies to a functor category of the form (C, A) and an argument category A to yield a category C, cancelling the argument place category A in (C, A). An operation of concatenation, on the other hand, combines a functor expression with an appropriate argument expression into a well-formed string.

Although ECG, as a version of categorial grammar, also employs these operations, it treats them as two distinct operations each of which applies to a different component of the grammar.[19] As in classical categorial grammar, the cancellation operation f_{cancel} here applies to pairs of syntactic categories, in particular to pairs consisting of quotient or storage-carrying categories and their appropriate argument categories, whereas the concatenation operation f_{concat} applies to linguistic expressions, that is, phonological entities, combining them in accordance with a set of statements on linear precedence. In this section, we discuss three sub-operations of cancellation, f^{Σ}_{cancel}, f^{*}_{cancel}, and $f^{A\backslash *}_{cancel}$ that are needed for ECG. In the following section on ordered trees, we'll briefly discuss the operation of concatenation.

We turn now to the condition on cancellation. In classical categorial grammar, the cancellation operation must satisfy the condition of identity as follows:

(2) Category Cancellation

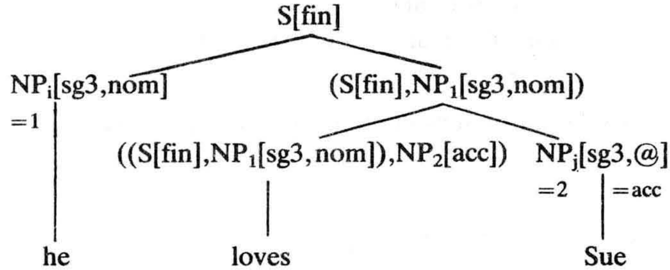
$$\begin{array}{ccc} & \mathrm{C} & \\ \diagup & & \diagdown \\ (\mathrm{C},\ \mathrm{A}) & & \mathrm{A}' \end{array} \qquad \text{where } \mathrm{A} = \mathrm{A}'$$

Here the category A is cancelled from the category (C, A) to yield the mother category C provided A is identical with A'. Otherwise, the cancellation fails.

But in ECG such an identity condition will be considered too strong. Consider:

[19] This approach was originally taken by GPSG. See also Hausser's (1986) left-associative grammar.

(3)



This tree consists of two local trees: one is topped with the quotient category (S[fin],NP_1[sg3,nom]), and the other with the highest node S[fin].[20] The lower part of the tree represents how the quotient category of **loves** combines with its argument category of **Sue**. Here we note that the cancellation of NP_2[acc] in the quotient category would fail if we were to insist on its strict identity with the given argument NP_j[sg3,@] or even with its instantiated NP_2[sg3,acc]. But if we weaken this identity condition to a more general condition—for example, to a requirement that, for any given quotient category, its argument place category simply *subsumes* some appropriate category given as an argument,— then the cancellation of NP_2[acc] as shown in (3), for instance, will be successful. Likewise the cancellation of NP_1[sg3,nom] in the upper tree also becomes successful, for it subsumes the instantiated category $NP_{i=1}$ [sg3,nom] of **he**.

From the point of view of information flow, the notion of *subsumption* is a very intuitive one. It simply means that if an infomration carrier is at least as informative as another information carrier, then the former is subsumed by the latter. Compare our examples $NP_{j=2}$[sg3,acc] and NP_2[acc]. Since the former contains more information than the latter, it is subsumed by the latter. These two are subsumed by a simple NP and, for the most extreme case, the empty category containing no information subsumes every category including itself.[21] Since in ECG categroies are treated as set-theoretic objects, we can formally equate the notion of subsumption with the notion of *subset* :

[20] Although each category on the tree may and sometimes must carry in *index* feature, we mark, by a general convention, only those features which are considered relevant to a given analysis.

[21] For further discussion, see Sag and Pollard (1987: 30).

(4) Subsumption

For any given categories C and C',
C is said to subsume C' if and only if $C \subseteq C'$.

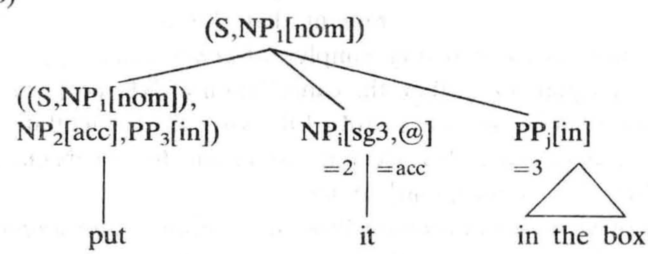
We thus adopt a relation of subsumption or subset as a condition on the operation f_{cancel}:

(5) f_{cancel} $((C,A\ ),\ A') = C$ if $A \subseteq A'$.

The operation f_{cancel} now applies to cases in which the argument place category A in a quotient category $(C,\ A\ )$ subsumes an argument category A', as well as to those particular cases in which they are identical.

However, some quotient categories in ECG take not just a single category, but a sequence of categories as its arugment categories, we must generalize the operation f_{cancel} to accommodate those cases shown in:

(6)



Here the category ((S,NP$_1$[nom], NP$_2$[acc], PP$_3$[in]) of **put** may take categories the NP$_{i=2}$[sg3,@=acc] and PP$_{j=3}$[in] as its arguments, since these categories are subsumed, respectively, by their corresponding argument place categories NP$_2$[acc] and PP$_3$[in]. We can thus make a reasonable conjecture that a sequence of argument place categories is cancelled from its quotient category if each category in the sequence subsumes a uniquely corresponding argument category. This process we capture first by defining the notion of *generalized subsumption* and then by setting up on its basis the operation of *generalized cancellation*:

(7) Generalized Subsumption

A sequence Γ of categories subsumes a set Σ of categories iff every category A_i in Γ subsumes a unique category A'_j in Σ.[22]

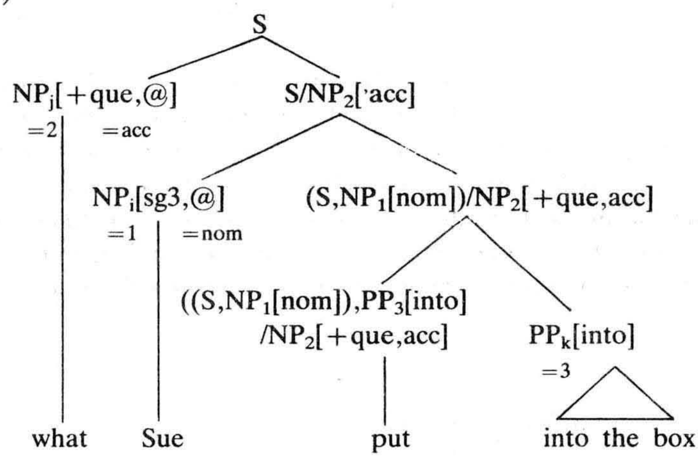
[22] Here we assume that no two identical categories ever occur in Σ, since each category,

(8) Generalized Cancellation [tentative]
For any category C, any nonnull sequence Γ of categories, and any set Σ of categories, $f^{\Sigma}_{cancel}((C, \Gamma), \Sigma) = C$ provided Γ subsumes Σ.

Unlike some of the recent versions of categorial grammar, ECG does not employ operations, such as wrapping or functional composition in order to deal particularly with unbounded dependencies or free word order. Instead, it utilizes storages, that is, categories containing a stored feature ⟨*stored, C*⟩ and a related set of cancellation operations. Among this set are the generalized cancellation operation f^{Σ}_{cancel}, the storage cancellation operation f^{*}_{cancel}, and the storage percolation operation $f^{A\backslash *}_{cancel}$.

First, we revise the operation f^{Σ}_{cancel} to accommodate cases in which a quotient category contains a storage. Consider:

(9)



In ECG, this tree is to be treated as admissible. In order thus to be able to apply f^{Σ}_{cancel} to the category of **put**, we should ignore the presence of the storage NP_2[+que,acc] after the slash in it.[23] This storage, however, must not be cancelled and passed up to the mother category. We accommodate these observations in the following revised formulation f^{Σ}_{cancel}:

with the exception of expletives, contains at least a distinct *index* feature. However, if they did, we would treat Σ as a multi-set. Or else, Σ may be treated not as a set of categories, but as the set of possible sequences over a set of categories.

[23] By our convention, $C/D = C \cup \{\langle stored, D\rangle\}$, where D is called a storage.

(10) Generalized Cancellation
For any category C, any nonnull sequence Γ of categories, and any set Σ, f^{Σ}_{cancel} $((C, \Gamma)_i \cup E, \Sigma) = C \cup E$ provided Γ subsumes Σ, where E is either an empty set or a category $\{\langle stored, D\rangle\}$.

Our earlier formulation of generalized cancellation is now seen as a particular case of (10) in which E is an empty set. However, if E is a nonempty set just consisting of a storage, this storage percolates up to the mother node, as shown at the two lower nodes of (9). At the topmost node this storage is used up, or cancelled. But our generalized cancellation f^{Σ}_{cancel} fails to effect such cancellation.

We thus introduce another sub-operation of cancellation:

(11) Storage Cancellation
For any non-quotient category C and any categories D and D', f^{*}_{cancel} $(C \cup \{\langle stored, D\rangle\}, D') = C$ provided D subsumes D'.

For illustration, consider (9) again. This operation f^{*}_{cancel} applies to the storage-carrying category S/NP$_2$[+que,acc] and the category NP$_{j=2}$[+que, @=*acc*], yielding S only. Here the category NP$_2$[+que,ace] may be considered to have *popped* from the storage when the storage-carrying category, being no longer a quotient category, is forced to use up something stored.

Finally, we introduce the operation $f^{A\backslash *}_{cancel}$ in order to deal with cases in which an argument category contains a storage. Consider the following example:

(12)

S[fin]/NP$_2$[+que, acc]

(S[fin],S[bas]) S[bas]/NP$_2$[+que,acc]

does Paul love

This tree again seems to be intuitively admissible. Here the auxiliary verb **does** is treated as of the category (S[fin],S[bas]) because it combines with an expression of the category S[bas] such as **Paul love her** to form a direct question such as **dose Paul love her**. But since, in the above tree, the storage-carrying category S[bas]/NP$_2$[+que,acc] is given as the argument category corresponding to the argument place category S[bas] in the quotient category of **does**, none of our previous formulations of cancellation applies here. To accommodate cases like (12), we introduce a third form of cancellation:

(13) Storage percolation
For any categories C, A, A', and D,
$f^{A \backslash *}_{cancel}((C,A),\ A'/D) = C/D$ provided A subsumes A'.

Now this new operation allows the category S[bas] to be cancelled from its quotient category (S[fin], S[bas]), yielding the mother category S[fin]/NP_2[+que,acc]. Note here that the storage NP_2[+que,acc] percolated up to the mother S[fin] may be cancelled, at a later stage in the derivation, by f^{*}_{cancel} when the mother node of (12) combines the category $NP_{j=2}$[+que,acc] of **whom**, yielding the **wh**-question **whom does Paul love**.

Although we have presented three different forms of cancellation f^{Σ}_{cancel}, f^{*}_{cancel}, and $f^{A \backslash *}_{cancel}$, these must not be considered as isolated operations. They are three particular cases of the single partial function f_{cancel} mapping pairs of a category and a category set into categories:

(14) Generalized Form
$f_{cancel} : Cat \times Pow(Cat) \rightharpoonup Cat$
where *Cat* is a set of categories and
Pow(Cat) is the power set of categories.

Each of the above three types of f_{cancel} is a restricted case of the general form (14) with different conditions on its domain and range. First, the operation f^{Γ}_{cancel} has its domain restricted to a set of pairs consisting of a quotient or storage-carrying category and a set of categories. Secondly, the operation f^{*}_{cancel} applies to non-quotient storage-carrying categories and categories that fill in storages. Thirdly, the domain of the operation $f^{A \backslash *}_{cancel}$ is restricted to pairs of a quotient category and a storage-carrying category; and its range, storage-carrying categories. With these restrictions, we could have stated all of the three operations of cancellation under the general form (14).

We now view ECG as providing a format for unification. On the basis of cancellation operations f_{cancel} and other required operations, it unifies the attribute-value matrices, or *avm's,* of linguistic objects into one matrix carrying an increased amount of information. Let m_i be an *avm* of the form:

(15) $\begin{bmatrix} Cat_i \\ Sem_i \\ Phon_i \end{bmatrix}_{Cat_i}$

Then the result of unifying attribute-value matrices in ECG will be defined as follows:

(16) Unification in ECG

Given a well-formed *avm* m_f and a set Σ of *avm's* $m_1,\ldots m_j$,

$unify(m_f, \Sigma) = m_n$

where $Cat_n = f_{cancel}\ (Cat_f,\ \{Cat_1,\ldots,Cat_j\}\ )$,

$Sem_n = f_{cancel}\ (Sem_f,\ \{Sem_1,\ldots Sem_j\}\ )$,

$Phon_n = f_{cancel}\ (Phon_f,\ \{Phon_1,\ldots,Phon_j\}\ )$,

where f_{cancel} is the operation of concatenation satisfying the conditions on linear precedence.

First, the category Cat_n of the resultant matrix is obtained by applying f_{cancel} to a quotient or storage-carrying cateogry Cat_f and a set of argument categories $Cat_1,\ldots,Cat_j$. Secondly, the semantic content Sem_n of the resultant matrix is obtained by merging pieces of information contained in each of the input Sem_k's associated with the input Cat_k's. As will be shown in the section 5.2, the operation of merging applies to situation-theoretic objects, that is, sets of equations, satisfying the conditions of unification constrained by co-indexing and type-matching. Since semantic contents consist of equations, their merging simply means solving these equations. Thirdly, the resultant expression $Phon_n$ is obtained by concatenating the expressions of input *avm*'s, constrained by linear precedence (LP) statements based on syntactic information. In addition, it must be emphasized here that these operation of f_{cancel}, f_{merg}, and f_{concat} are not autonomous: in the process of unifying *avm*'s into one *avm*, they interact simultaneously with one another, each requiring relevant pieces of information from the other operations.

4. Ordered Trees

The process of unification may be represented in a tree form. Each input *avm* constitutes a daughter node and the resultant *avm*, the mother node of a local tree. By convention, these nodes are to be labelled by their associated syntactic categories. Among the daughter nodes of each local tree is a unique element called a quotient or storage-carrying category. The other daughters are called argument categories some of which are obligatory and others of which are optional.

In forming an admissible local tree, we need at least two pieces of information: one about the label, or category, of each node, and the other about the linear order of daughter nodes. Given a quotient or storage-carrying category, the operation f_{cancel} conveys information about what categories are necessary and admissible as its argument categories. However, it fails to give the information about how they should be ordered to each other. This information is conveyed by the same set of LP statements that constrains the operation of f_{cancel}. While some relevant LP statements put the daughter categories admitted by f_{cancel} into a linear sequence, f_{cancel} joins their expressions, or *Phon*$_k$'s, in the same linear sequence. Here is a partial list of LP statements relevant to English:[24]

(1) LP Statements
  [i] Quotient categories precede their arguments.
  [ii] A quotient category of the form (S, *C*) is preceded by its argument category *C*.
  [iii] A storage-filler precedes its corresponding storage-carrying category.
  [iv] The linear order of complements follows the sequential order of their corresponding argument places in a quotient category.

These statements describe the phenomena of word order in English. Statement [i] says that a verb precedes its complements or adjuncts, and a determiner its nominal expression. Statement [ii] allows any subject category *C* to precede its VP. Statement [iii] places **wh**-words and topic words at the beginning of clauses. Finally, statement [iv] fixes the order of verbal complements, resulting in fixed order for the objects in a double object construction as in:

(2) Sue gave him a doll.

The list is not exhaustive. We should, for instance, have a statement

[24] These formulations should be understood as restatements of various proposals about word order made by Gazdar and Pullum (1981), Gazdar, Klein, Pullum, and Sag (1985), and, in particular, Sag (1986).

based on types of categories, giving a certain linear hierarchy among them. But, unless it is constrained by some LP statements, word order in English is not fixed. For instance, there is no fixed order among adjuncts and thus we can have either of:

(3) a. John spoke about Sue with Bob.
 b. John spoke with Bob about Sue.

5. Semantic Merging

ECG adopts situation theory for an adequate description of the use of linguistic expressions to convey information about various objects in the world. A semantic framework based on this theory, we believe, can elegantly accommodate various aspects of meaning and information content. It can treat not only the described content, but also both the embedding circumstances and the resulting impact relevant to the use of a language in a discourse in order to present a fuller account of its correctness and accuracy. In this secton, we introduce some basic elements of situation theory and discuss what types of contents are described by the use of expressions and how these contents are merged by equation solving.

5.1 Situation-theoretic Objects

One of the basic elements in situation theory is a state of affairs, or simply soa. A soa σ consists of a relation R, an assignment a which assigns appropriate objects, $a,b,c...$, to some of the argument roles in the relation R, and a polarity. Since the polarity is either 1 or 0, there are two types of soa's: a positive soa $\langle\!\langle R,\mathbf{a};\ 1\rangle\!\rangle$ and a negative soa $\langle\!\langle R,\mathbf{a};\ 0\rangle\!\rangle$. Consider the relation LOVE and an assignment $\mathbf{a}'$ which assigns Paul to the role of a lover(arg_1) and Sue to the role of being loved(arg_2). Then, we may have a positive soa ⟨⟨LOVE, arg_1: Paul, arg_2: Sue;1⟩⟩, or simply ⟨⟨LOVE, Paul, Sue; 1⟩⟩. This soa is a fact if it holds in some actual situation.

Another element is a proposition. A typical example is the claim that a soa σ holds in a situation s, which is represented as $(s \models \sigma)$. While being a fact is a property of soas, truth is a property of propositions only.

Soas and propositions may contain indeterminates, or parameters. Reconsider the loving relation. Intead of assigning Sue to arg_2, we may also assign the indexed parameter x_2, thus obtaining the soa $\langle\langle$LOVE, Paul, x_2; 1$\rangle\rangle$. This soa is called a parameteric soa, or simply psoa. Likewise we can form a parametric proposition by allowing indeterminates in it: e.g. $(s \models \langle\langle \text{LOVE}, \textit{Paul}, x_2; 1\rangle\rangle)$.

These parametric objects are used to impose conditions on other parametric objects to form complex indeterminates, properties, and types. First, we can obtain a restricted individual parameter $x^{p(x)}$. A proper noun such as 'Paul' may be interpreted as referring to an individual to which the restricted parametric object $x^{(s \models \langle\langle \text{NAMED}, x, \text{'Paul'}; 1\rangle\rangle)}$ is anchored. Next, we can form a property $[x \mid \sigma(x)]$. The property of loving Sue, for instance, can be represented as $[x \mid \langle\langle \text{LOVE}, x, \text{Sue}; 1\rangle\rangle]$. Here the psoa restricts the range of values for the parameter x. Thirdly, we may also obtain a type $\hat{x}p(x)$: for example, $\hat{x}(s \models \langle\langle \text{LOVE}, x, \text{Sue}; 1\rangle\rangle)$. These types are a subcollection of properties. They differ from each other in the occurrence of a situation s in their characterization. In situation semantics, we thus have various kinds of real objects: individuals, properties, and types, as well as states of affairs and propositions. Corresponding to these we also admit parametric objects.

5.2 Semantic Content and Merging

In ECG, we assign a set of equations to each use of an expression as its semantic content. What kind of equations are assigned is closely associated with its syntactic category. If an expression belongs to a quotient category, then its content will be associated with a psoa or a parametric proposition. A verbal expression **snore** of the base form for instance will be assigned an equation of the following form:

(1) **snore**: (S[bas], NP_1)
$$\sigma = \langle\langle \text{SNORE}, x_1, l; 1\rangle\rangle$$

On the other hand, the same verb **snored** of a finite form will be assigned a set of equations:

(2) **snored**; (S[fin], NP_1)
$$\begin{bmatrix} p = (s_d \models \sigma) \\ \sigma = \langle\langle \text{SNORE}, x_1, t_d; 1\rangle\rangle \\ t_d = t^{(s_u \models \langle\langle \text{PRECEDE}, t, t_u; 1\rangle\rangle)} \end{bmatrix}$$

Here s_d, s_u, t_d, and t_u are a described situation, an utterance situation, a described time, and an utterance time, respectively.

The semantic content of a singular NP such as **Tom** will consist of an equation of the following form:

(3) **Tom:** $NP_i[sg3]$

$$x_i = x^{((s_d \models \langle\text{REFERRED-TO}, x, \text{'}Tom\text{'}; 1\rangle) \wedge (s_b \models \langle\text{NAMED}, x, \text{'}Tom\text{'}; 1\rangle) \wedge (s_i \models \langle\text{SALIENT}, x, MALE; 1\rangle))}$$

The proper noun **Tom** is used to refer to an individual in a described situation s_d. Furthermore, for its appropriate use, one must presume as background that that person is named Tom. As a result of such a use, this individual then becomes salient with respect to malehood so that he may be referred to by a use of the pronoun **he**.

Now, we will show how the whole content of **Tom snored** can be obtained. We can obtain it by simply unifying the contents, i.e. sets of equations, of its constituent expressions.

(4)

$$\begin{bmatrix} S[fin] \\ p = (s_d \models \langle\!\langle \text{SNORE}, x^{tom}, t_d; 1 \rangle\!\rangle) \\ t_d = t^{(s_u \models \langle\text{PRECEDE}, t, t_u; 1\rangle)} \end{bmatrix}$$

$$\begin{array}{cc} \begin{bmatrix} NP_{i=1}[sg3] \\ i = 1 \\ x_i = x^{tom} \\ \textbf{Tom} \end{bmatrix} & \begin{bmatrix} (S[fin], NP_1[sg3]) \\ p = (s_d \models \sigma) \\ \sigma = \langle\!\langle \text{SNORE}, x_1, t_d; 1 \rangle\!\rangle \\ t_d = t^{(s_u \models \langle\text{PRECEDE}, t, t_u; 1\rangle)} \\ \textbf{snored} \end{bmatrix} \end{array}$$

As shown in the above tree, the content of a complex expression is obtained by putting together the contents of its constituent expressions and simply solving them by substituting appropriate objects into indices or parameters. Note first that , unlike the content of **Tom** in the lexicon, its content on the tree contains the information $i = 1$ about coindexing; this piece of information is crucial for substitutions in the process of equation solving. It has resulted from anchoring the index i of the category NP_i to 1 in order to meet the subsumption condition on cancellation. Furthermore, we have used the abbreviated form x^{tom} for the content of **Tom**.

In ECG, as has been briefly discussed, the semantic merging operation f_{cancel} is nothing but a simple process of unifying the contents of constituents. In a forthcoming paper, Yoo and Lee (1987), we will treat a larger fragment which includes many of the constructions discussed in the present paper and show how the semantic merging works in our framework.

6. Applications

In this section we will concentrate on illustrating how the syntax of ECG works in particular for English.

6.1 Grammatical Relations and Semantic Roles

In ECG, every NP is to be assigned a grammatical relation *gr*. The relation *gr* is purely a syntactic notion such that even so-called dummy NP's, or expletives, are assigned *grs*. We have two *grs*, subject and object, in the grammar. Closely following Dowty (1982), we define these notions as follows:

(1) Grammatical Relations

Subject: The category C in a quotient category (S, C) is the subject of S.

Object: Let C be a quotient category of the form $(C_0, C_1, \ldots, C_i \ldots)$, where $C_0 = S$ and $C_i = NP$. Then each C_i $(i > 1)$ is its object.

For illustration, consider the category ((S,NP_1),NP_2). Here NP_1 is the subject of S, and NP_2 an object of the entire verbal category. In the case of a prepositional category such as (PP,NP), the NP is its object.

Some categories may have multiple objects. In such a case, these objects are distinguished by their position in the sequence. By convention, we may name an object NP occurring in the left-most position of a quotient category *direct object*, and any object following it *indirect object*. Our definition of grammatical relations can be illustrated as follows:

(2)

$$\begin{array}{l} ((\underset{|}{\underset{\text{Subj}}{S,NP}}),\underset{\text{DObj}}{NP},\ldots,\underset{\text{IObj}}{NP},\ldots) \end{array}$$

Theoretically speaking, there may be some non-NP categories intervening between direct and indirect objects. So we allow dots between two object NP's.

Although grammatical relations are, in a proper sense, relations holding between quotient categories and their argument categories, we have defined them as relations between constituent categories of a quotient category or between a quotient category and its argument place categories. Grammatical relations have thus been defined in terms of a quotient category and its constituent categories. This is certainly a departure from the traditional way of defining grammatical relations. It will not, however, result in any confusion as long as we understand *gr* as an attribute of a category, for all information encoded in an argument place category is transferred to its corresponding argument category by the subsumption condition on cancellation.

We turn now to the notion of semantic roles. While the grammatical relation of an argument category is determined by its location in a quotient category, its semantic role (of being an agent, a recipient, or a location, etc.) is not determined structurally. It should be assigned to an object persisting in a described state of affairs, but only derivatively to an expression or a category. Consider a statement made by the following sentence:

(3) Paul loves Pommy.

This statement may be interpreted as describing a situation in which Paul loves Pommy. Here it is not the NP's, but the boy named Paul and the female kitten named Pommy referred to by those NP's that are understood as playing the roles in the relation LOVE. We can say in general that, given a basic state of affairs of the form $\langle R, x_1, \ldots, x_n; 1\rangle$, the object assigned to each x_i plays a unique *argument role* with respect to the relation R. However, we may also be able to speak of argument roles in a derivative sense as if they were associated with certain categories of expressions used in a statement. In this paper, we shall coin the slightly different term *semantic role* and use it in a derivative sense. We also use it in a narrower sense by treating it as a subtype of argument role assigned only to an object which is described by a maximal category.

We have introduced indices $f,g,h,i,j,\ldots; 1,2,3,\ldots$ to link categories with their described semantic objects. We, in particular, use numerical indices to mark semantic roles so that maximal categories can be linked with their associated argument roles. A transitive verb **love,** for instance, belongs to a quotient category of the form $((S,NP_1),NP_2)_i$, where the index of each NP is assigned a numerical value. A use of this verb in an actual discourse will then be understood as conveying the information that an individual x_1 corresponding to NP_1 loves an individual x_2 corresponding to NP_2. In our framework, this information will be represented in an equational form: $\sigma_i = \langle\langle LOVE, x_1, x_2; 1\rangle\rangle$. This is interpreted as saying that the type of an object described by the verb **love** is a state of affairs σ_i and that this state of affairs is identified with a LOVE relation between two indeterminate individuals x_1 and x_2.

The use of indices is crucial in capturing the semantic roles related to constructions such as Passive. Consider:

(4) He is loved by Sue.

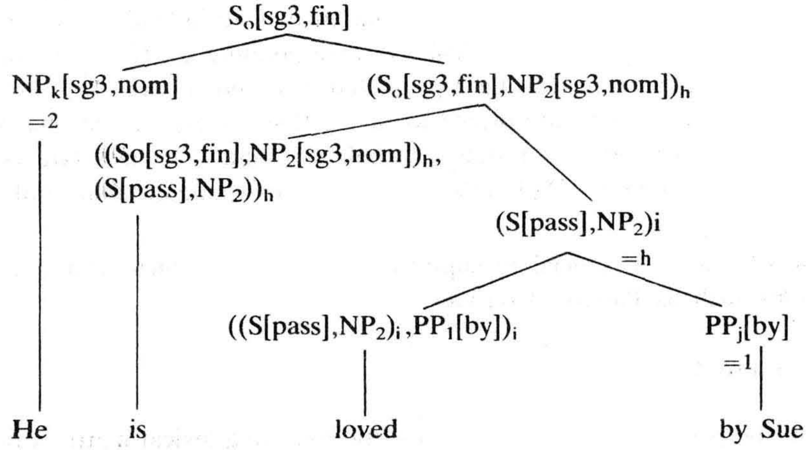
To analyze the passive sentence (4), we list the following lexical matrices in the lexicon:

$$(5)\quad \begin{bmatrix} ((S[pass],\ NP_2)_i,\ \{PP_1[by]\})_i \\ \sigma_i = \langle\langle LOVE, x_1, x_2; 1\rangle\rangle \\ \textbf{loved} \end{bmatrix}$$

$$(6)\quad \begin{bmatrix} ((S_o[sg3,fin],\ NP_2[sg3,nom])_h,\ (S[pass],NP_2))_h \\ p_o = (s_d \models \sigma_h(t_h)) \\ t_h = t^{(s_d \models \langle\langle OVERLAPS, t, t_u; 1\rangle\rangle)} \\ \textbf{is} \end{bmatrix}$$

Here are a few important remarks: First, $PP_1[by]$ in (5) is the agent *by*-phrase with the semantic role 1 of some relation involved in the state of affairs σ_i described by the verb **loved**, while NP_2, the subject of S[pass], takes the semantic role 2 of the same relation. Secondly, the content of the auxiliary verb **is** consists of two equations: one provides a propositional content p_o and another sets up its temporal location. By making use of thse matrices, we can first construct the following tree:

(7)



Secondly, on the basis of this analysis, we can unify sets of equations provided by each constituent expression, thus obtaining:

$$(8)\quad \begin{bmatrix} p_o = (s_d \models \langle\!\langle \mathrm{LOVE}, x^{\mathrm{SUE}}\, y^{(s_b \models \langle\!\langle \mathrm{MALE}, y; 1\rangle\!\rangle)}, t_h; 1\rangle\!\rangle \\ t_h = t^{(s_d \models \langle\!\langle \mathrm{OVERLAP}, t, t_u; 1\rangle\!\rangle)} \end{bmatrix}$$

This resultant matrix successfully captures the information content conveyed by the passive sentence (4), where Sue plays the role of loving, and some male that of being loved. As has been shown, the unification of equations simply means putting equations together and solving them on the basis of pieces of indexical information.

Both *gr* and *index* are syntactic attributes. But their functions are not the same: *gr* is associated with *case,* whereas *index* with semantic, or argument, roles. For example, the NP **he** in (4) is the subject and thus its case is marked nominative, but, due to the numerical *index* 2, its semantic role is associated with the argument role *arg* 2 of being loved. Secondly, while *gr* is assigned to every NP including expletives, a numerical *index* is not assigned to expletives. Note, however, that, while *gr* and *case* are assigned to NP's only, an *index* may be assigned to categories other than NP such as S, PP, or VP. In our previous example, we have assigned the *index* 1 to the agent phrase PP[by] in the quotient category of the passive form **loved**. Hence we claim that the attribute *index,* distinct from *gr,* is needed to establish an interface between a syntactic category and its corresponding semantic content.

6.2 Agreement

In ECG, agreement is accounted for by some constraints on quotient categories. In English, for instance, subject NP's agree with their verbs in number and person; determiners with their nouns in number. By adding a list of some pertinent constraints on agreement to the lexicon and then by encoding relevant pieces of information into quotient categories, we can properly constrain the specification of agreement attributes.

First, consider agreement between a verb and its subject:

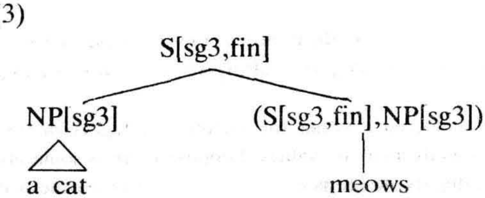
(1) a. A cat meows.
    b. *Cats meows.
    c. *A cat meow.

(1a) is well-formed because the verb form of **meows** agrees with its subject NP **a cat** in number and person, while (1b, c) are ill-formed because there is no such agreement there. To deal with this sort of agreement, we introduce the following constraint between categories within a quotient category:

(2) Subject-Verb Agreement
    If a category is a subcategory of (S[fin $\vee$ bas], NP), then it is of a category: (S[#%, fin $\vee$ bas], NP[#%]).

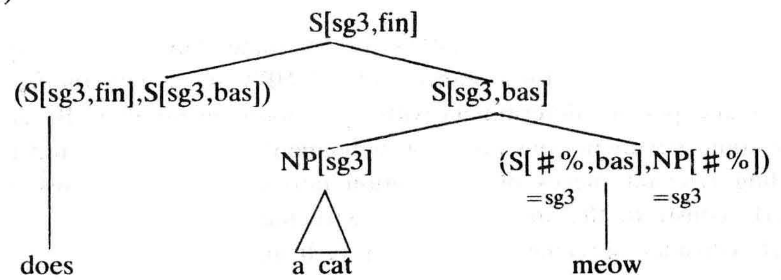
This constraint states that the *number* and *person* attributes of NP and S must be defined and agree in value, if the form of S is either finite or base. This requires a verb such as **meows** to belong to the category (S[sg3,fin],NP[sg3]). f_{cancel} then admits the following tree:

(3)



Subj-V Agreement is not restricted to cases involving finite verbs only. It also applies to base-form verbs, thus allowing the category of the verb **meow** to be (S[#%,bas],NP[#%]).[25] Since the auxiliary verb **does** is of the category (S[sg3,fin], S[sg3,bas]), we obtain the following tree:

(4)



Here the parameters #% in the category of **meow** are anchored to *sg*3 by forcing.[26] Then f_{cancel} applies routinely to yield the tree (4).

Secondly, we discuss a constraint on number agreement between an adnominal expression and a nominal expression.

(5) Adnom-N Agreement
If a category is a subcategory of ([+N, −V], [+N, −V]), then it is of a category ([+N, −V, #], [+N, −V, #]).[27]

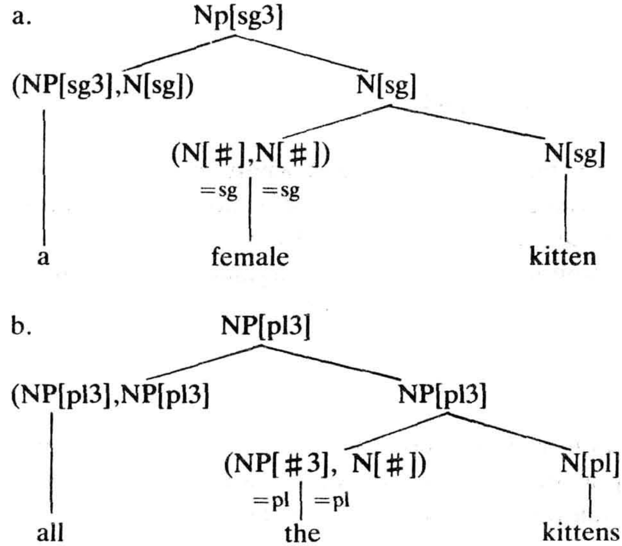
Because of this constraint, every adnominal expression (adjective or determiner) must agree in number with its corresponding nominal argument. The following are some examples:

[25] Here we assume some conditions on feature specification like the following: (i) if a category is S, then its *form* must be specified. (ii) If a category is S[fin], then its *number* and *person* attributes must be assigned values.

[26] In applying f_{cancel}, matching must be forced if either the argument category or the argument place category contains parameters as its attribute values. Suppose there is some attribute *a* such that its value is specified in one category *A*, but is only parametrized in another category *A′* and, furthermore, that these two categories are compatible with each other except for each of such an attribute *a*. Then, each parameter is anchored to the specific value of its corresponding attribute. Thereby, these two categories are forced to match. The result of forcing is marked below each parameter.

[27] Note that both N and NP are subcategories of the category [+N, −V].

(6)



In (a), the determiner **a** agrees in number with its argument **female kitten**, and the adjective **female** with its argument **kitten**. Similarly, in (b), the specifier **all** agrees in number with its argument **the kittens**, and the determiner **the** with its argument **kittens.** Note here that the specifier **all** also requires a constraint on agreement in person between its argument NP and the resultant NP; we thus treat it as belonging to the category (NP[pl3], NP[pl3]). On the other hand, the determiners **a** and **every** simply belong to the category (NP[sg3], N[sg]), and the determiners **the** and **no** to the category (NP[#3], N[#3]). However, they all produce NP[#3]'s when they take either an NP or an N category argument.

6.3 Case

Since oridnary NP's in English carry no case forms, ECG does not assign *case* directly to NP's that function as argument categories, but to those NP's that occur in quotient categories. There are, however, two problems associated with *case*. First, pronouns and **wh**-words in English display *case* forms. Thus, information about their *case* values we encode into the lexicon.[28]

[28] While **wh**-words are subcategorized with respect to *case*, ordinary pronouns are subcategorized with respect to *number, person,* and *case*. The attributes like *gender* and *humanhood* are treated as semantic properties.

(1) **she**: NP[sg3, nom]
her: NP[sg3, acc]

(2) **who**: NP[nom, +que]
who, whom: NP[acc, +que]

Note here that the genitive forms **her and whose** are not treated as NP's, but of the quotient category (NP[#3], NP[#]). So, in ECG, we have only two values for the *case* attribute: *nom* and *acc*.

Secondly, when pronominal NP's serve as arguments of a verbal category, their case forms must be properly selected.

(3) She (*Her) meows.

If a pronoun occurs in a subject position, its case form must be *nominative*. For this, we set up a constraint on *case* value assignment in the lexicon and encode pieces of relevant information on verbal categories.

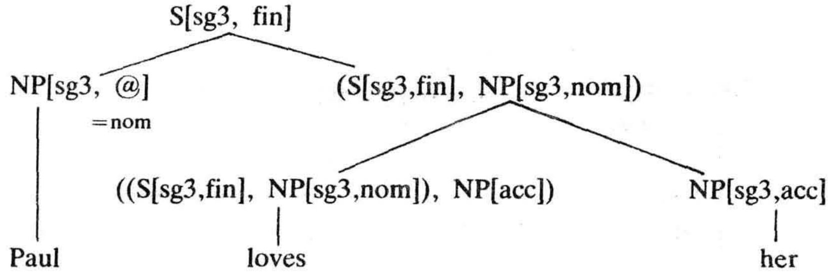
(4) Case Assignment
Given a quotient category of the form ((C, NP_i), Γ), where Γ is a (possibly null) sequence of categories,
a. NP_i is assigned the case value *nom,* if C = S[fin $\vee$ bas]; otherwise, it is assigned *acc;*
b. Any other NP occurring in Γ is also assigned the case value *acc*.

According to this constraint, the following quotient categories are well -formed:

(5)
a. **loves**: ((S[sg3, fin], NP[sg3,nom]), NP[acc])
b. **meow**: (S[#%, bas], NP[#, nom])
c. **in**: (PP[in], NP[acc])
d. **expects**: ((S[sg3, fin], NP[sg3, nom]), S[to])

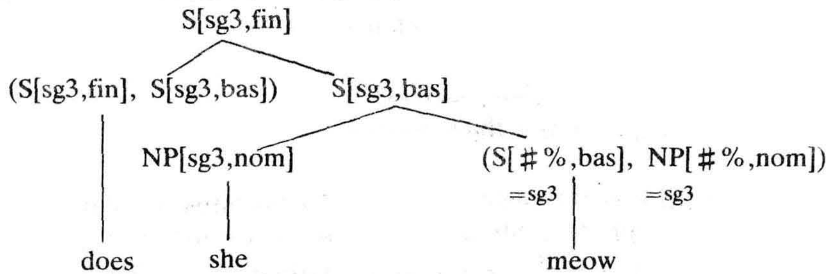
For illustration, consider:

(6)



In this tree, each application of f_{cancel} is acceptable because the category of **her** is subsumed by NP[acc] and the instantiated category of **Paul** is also subsumed by NP[sg3,nom]. Consider another example:

(7)

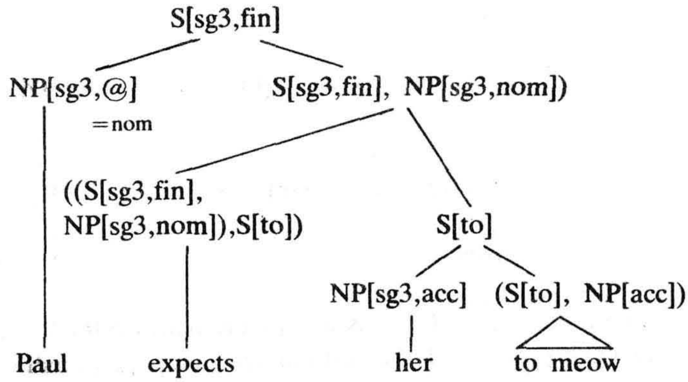


Since the verb **meow** is of the category (S[bas], NP), the *case* value of NP in it is *nom*. By the constraint on agreement, the two categories NP and S[bas] agree in *number* and *person* and then the parametric values of their *number* and *person* are anchored to *sg3* to meet the subsumption condition. The operation f_{cancel} then derives the above tree in a routine manner. Here is a third example:

(8) a. Paul expects her to meow.
 b. *Paul expects she to meow.

In ECG, (8a) is analyzed as follows:

(9)



Since the *vform* of **to meow** is not *fin* nor *bas,* its subject NP must be assigned *acc.* Hence, (8a) is admissible, while (8b) is not. Finally, consider a case involving long-distance dependency:

(10) a. Who do you think meows?
b. *Whom do you think meows?

Here while (a) is well-formed, (b) is not, for the latter violates the principle of case assignment. This difference can also be accounted for just by checking how the lexicon works. The lexicon, restricted by various constraints on agreement and case assignment, allows the following attribute specifications, where the attribute value complex *sg3* occurring in the category of **do** stands for any pair of *agr* values other than *sg3.*

(11) a. **who**: NP[#%, nom, +que]
b. **do**: (S[fin], S[bas,−sg3])
c. **you**: NP[#2,@]
d. **think**: ((S[#%,bas], NP[#%,nom]), S[fin])
e. **meows**: (S[sg3,fin], NP[sg3,nom])

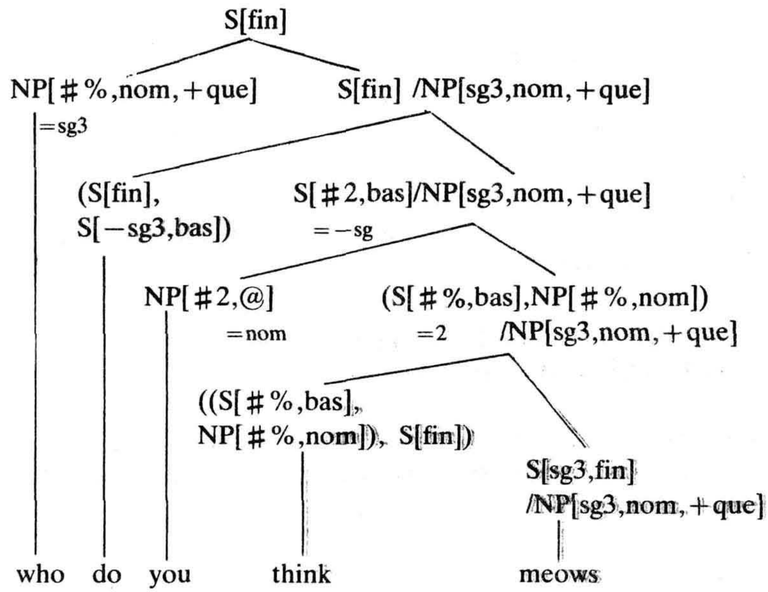
Here we have just omitted marking indices because they are not relevant to the present discussion. On the basis of (11e), we can create a storage-carrying category as shown in (12), with the storage creating lexical rule:[29]

[29] See section 2.5.2 about a storage creating rule.

(12) **meows**: S[sg3,fin]/NP[sg3,nom,+que]

This then admits the following tree:

(13)



At the lexical level, the storage here is assigned attribute values *sg3*, *nom*, *+que*.[30] This storage then is carried up to the higher nodes till it is filled in by the question word **who** the *case* of which is *nom*.

6.4 Relative Clauses

We have already shown how we treat the following **wh**-questions:

(1) Whom does Paul love?
(2) Who do you think meows?

We obtain these sentences by treating **wh**-phrases as fillers for storages. The

[30] We propose the following condition of feature specification: Every NP is either +que or −que. This proposal seems reasonable because we can always tell whether any given NP is a question word or not. But we mark this distinction only when it is relevant.

verb **love** in (1), for instance, contains a storage in the object position. This storage is eventually filled in by its NP filler **whom**.

Storages, or so-called gaps, are also found in relative clause constructions in English. In this section, we will continue to show how storages are put to use for the analysis of the following relative clause constructions:

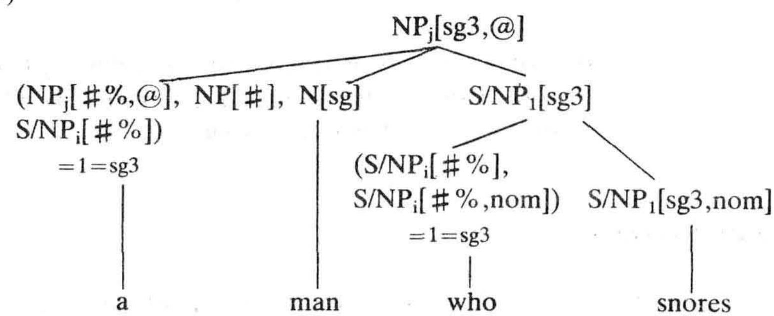
(3) A man who [ ] snores
(4) A girl who we think [ ] meows
(5) A man bothers me who [ ] snores

First, while **wh**-question phrases such as **who** and **whom** are of the category NP, we treat (restrictive) relative pronouns as belonging to a quotient category as shown in (6)[31]:

(6) **who**: (S[fin]/NP_i[#%], S[fin]/NP_i[#%,nom])
whom: (S[fin]/NP_i, S[fin]/NP_i[acc])

Secondly, as in HPSG, we treat a (restrictive) relative clause of the form S/NP_i[#%] as an adjunct of a determiner of the form (NP_j[#%,@], N, S/NP_i [#%]). Now, (3) will be analyzed as follows:

(7)



Since the instantiated argument-place category S/$NP_{i=2}$[#%=sg3,nom] of **who** subsumes the input argument category S/NP_i[sg3,nom] of **snores**, the operation f^{Γ}_{cancel} applies to the categories of **who** and **snores**, yielding the category S/NP_1[sg3] of **who snores**. Note here that the information about agreement conveyed by the verb **snores** is passed up to the relative clause.

[31] Nonrestrictive relative pronouns are also treated as belonging to a quotient category, but their category is different from that of restrictive ones as follows: $(NP_i/NP_h, S/NP_h)_i$.

Because of this information, f^{Γ}_{cancel} can again apply to the categories of **a**, **man**, and **who snores**, yielding the category NP_j[sg3,@] of **a man who** snores. Note also that the information $i=1$ about the semantic role of a missing argument is passed up to the determiner through the relative pronoun. This transferred information about the role will then be reflected in the semantic content of the mother NP_j in such a way that the property expressed by the relative clause restricts the property MAN.[32]

We turn next to the relative clause construction (4):

(4) A girl who we think [ ] meows

This construction is supposedly troublesome because a gap corresponding to the relative pronoun **who** is considered to be deeply embedded in the complement of the main verb **think.** However, this causes no problem in ECG, for it can create storage within the category of the verb **moews** and carry it up to the top of the relative clause without being blocked by the main verb **think.**[33] To show this, we first list the category of each of its constituents:

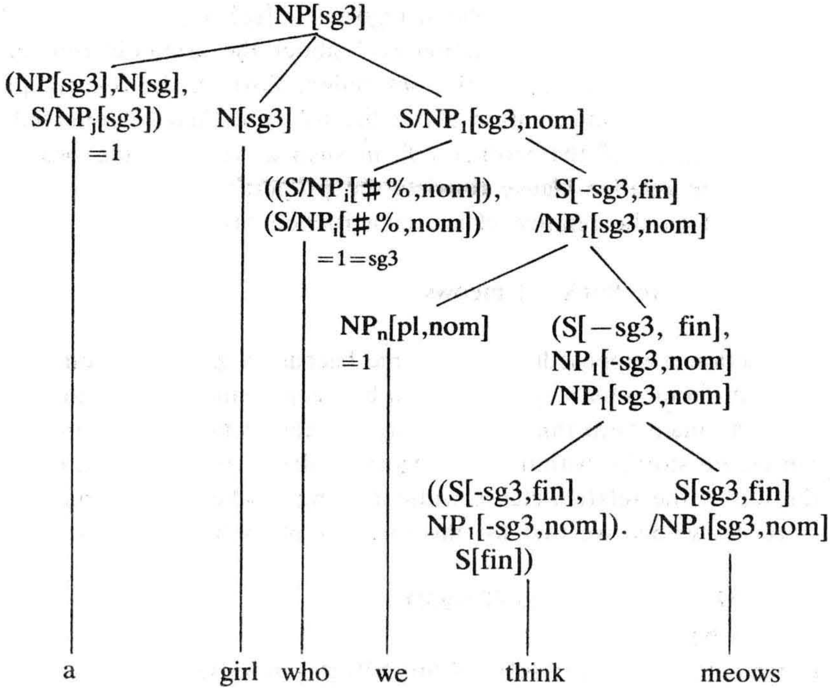
(8) a. **a**: (NP[sg3], N[sg], S/NP_j[sg3])
b. **girl**: N[sg]
c. **who**: ((S/NP_i[#\$,nom]), (S[fin]/NP_i[#%,nom]))
d. **we**: NP_n[pl1,nom]
e. **think**: ((S[-sg3,fin], NP_1[-sg3,nom]), S[fin])
f. **meows**: S[sg3,fin]/NP_1[sg3,nom]

On the basis of (8), f_{cancel} derives:

[32] See Yoo and Lee (1987).

[33] The following set of sentences are also regarded as problematic: [i] a girl whom we think that Tom loves and [ii] *a girl who we think that meows. ECG can easily deal with case [i] in a routine manner by creating a storage in the category of the embedded verb **loves** and then by carrying it up to the top of the relative clause without being blocked by the complementizer **that**. On the other hand, the ill-formed sentence [ii] cannot be blocked in ECG unless we introduce some kind of constraint resembling *that*-filter of extended transformational grammar. In ECG, this constraint can be stated as *S[that]/NP[nom].

(9)



This tree is again obtained by various sub-operations of f_{cancel}. One interesting thing here is to note how the information concerning the *number* and *person* of the subject NP_1 encoded in the category of **meows** is passed up to the top of the tree. This information contained in the storage NP_1[sg3,nom] is eventually transferred to the category of the relative pronoun **who** to meet the subsumption condition, as is marked in the tree, and then is used to select an appropriate antecedent, for the information about agreement between the storage and the mother NP is encoded in the category of the determiner **a**. Thus the following relative clause construction is not admitted.

(10) *a girl who we think meow

Extraposition from NP is also easily accounted for in pur framework. Consider:

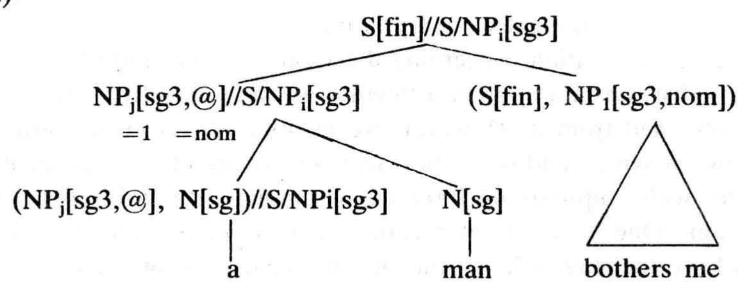
(5) a man bothers me who snores.

Since the determiner **a** is a quotient category, we can change it into a storage-carrying category such as, for example,

(11) **a**: (NP_j[sg3,@], N[sg]) // S /NP_i[sg3]

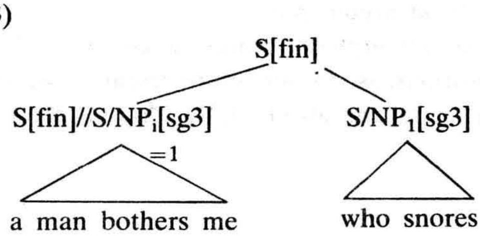
This cateogry carries the storage S/NP_i[sg3].[34] But the storage is ignored when the operation $f^{\Gamma}_{cancel|}$ applies to the categories of **a** and **man**, yielding the category NP_j[sg3,@]//S/NP_i[sg3]. This resultant category, with parameters j and @ anchored to 1 and *nom*, respectively, may now combine with the quotient category (S[fin], NP_1[sg3,nom]) of **bothers me**, as follows:

(12)



Finally, the storage cancellation operation f^{*}_{cancel} applies, combining the top node of (12) with a relative clause such as **who snores**:

(13)



As can be seen easily, ECG will allow the following examples too:

(14) A man bothers me whom everyone hates
(15) A man bothers me who is being constantly chased by a cop

[34] We use a double-slash // as in *A//B/C* when a category *A* carries a storage that also carries another storage *B/C*.

For this, we just need to anchor the index *i* of **a man bothers me** to 2, which will be associated with the argument role of being hated or being chased.

7. Concluding Remarks

The basic aim of this work has been to formulate the general framework of ECG in an attempt to construct a simple, elegant, and computationally tractable grammar model for situation semantics. The best way to achieve this goal in a naive way, we think, is to take some promising ideas from established or current theories and to amalgamate them into one coherent system. This is what we have attempted to do in the present work.

Our proposed ECG heavily relies on quotient, or functor, categories. In the lexicon, each quotient category is treated as providing the fullest possible piece of information concerning the local configuration of a well-formed phrase so that, given a quotient category, we know exactly what kind of tree can be obtained from it. However, we in principle reject any strict seperation between syntax and semantics. Instead, we adopt the relational view of language, with emphasis on interactions among various levels of linguistic description. One type of interactions is provided by the use of indices. Indices have the dual role of, on the one hand, linking categories (tokens) with their corresponding semantic contents and, on the other hand, marking semantic or argument roles.

In this work we have mainly discussed the syntatic part of ECG, leaving its semantics to a forthcoming work (Yoo and Lee, 1987). We have also been rather reluctant to dwell on argumentative aspects of supporting our framework; instead, we have attempted to present concrete illustrative analyses of some English constructions. In our future research, we intend to refine ECG by extending fragments and also by fully adopting the relational view of language.

REFERENCES

Barwise, Jon (1987) 'Notes on a Model of Situation Theory,' Unpublished draft.

Barwise, Jon, and John Perry (1983) *Situations and Attitudes*, The MIT Press, Cambridge, Mass.

Barwise, Jon, and Robin Cooper (1985) 'Generalized Quantifiers in Situation Semantics,' Unpublished draft.

Cooper, Robin, and Elisabeth Engdahl (1987) 'Situation Semantics for Long Distance Dependencies,' Incomplete draft.

Cooper, Robin (1987) *Introduction to Situation Semantics*, Draft.

Dowty, David (1982) 'Grammatical Relations and Montague Grammar,' in P. Jacobson and G. K. Pullum, eds., *The Nature of Syntactic Representation,* D. Reidel, Dordrecht.

Gazdar, Gerald, and Geoffrey Pullum (1981) 'Subcategorization,' in v.d. Hulst, and T. Hoekstra, eds., *The Scope of Lexical Rules,* Foris, Dordrecht.

Gazdar, Gerald, Ewan Klein, Geoffrey Pullum, and Ivan Sag (1985) *Generalized Phrase Structure Grammar,* Basil Blackwell, Oxford.

Hausser, Roland (1986) *NEWCAT: Parsing Natural Language Using Left-Associative Grammar,* Springer-Verlag, Berlin.

Karttunen, Lauri (1986) 'Radical Lexicalism,' Center for the Study of Language and Information, Stanford University, Report No. CSLI-86-68.

Lee, Kiyong, Suson Yoo, Key-Sun Choi, and Sangki Han (1986) 'Augmented Categorial Grammar with Its Computational Implementation,' *Language Research* 22, 415-453.

Pollard, Carl (1985) 'Lectures on HPSG,' Unpublished.

Sag, Ivan (1986) 'Grammatical Hierarchy and Linear Precedence,' Center for the Study of Language and Information, Stanford University, Report No. CSLI-86-60.

Sag, Ivan, and Carl Pollard (1987) *An Information-Based Approach to Syntax and Semantics* Lecture Notes Series, CSLI, Stanford.

Shieber, Stuart M (1986) *An Introduction to Unification-Based Approaches to Grammar,* Center for the Study of Language and Information, Stanford University, Stanford.

Uszkoreit, Hans (1986) Categorial Unification Grammars, Center for the Study of Language and Information, Stanford Universiy, Report No. CSLI-86-66.

Yoo, Suson, and Kiyong Lee (1987) 'Situation Semantics for Extended Categorial Grammar,' Unpublished.

Zeevat, Henk, Ewan Klein, and Jo Calder (1986) 'Unification Categorial Grammar,' Unpublished.

Department of English
Korea University
1 Anam-dong, Seongbuk-ku
Seoul 136-701
Korea