Chapter 4

Formalization of the Theory with Combinatory Categorial Grammar

In the previous chapter, we have mentioned that two potential problems for binomial partition of information structure, i.e., non-traditional constituency and discontiguous information structure, can be solved by adopting Combinatory Categorial Grammar (CCG) and by integrating structured meaning, respectively. This chapter demonstrates that how these two points can be achieved within a variant of CCG formalism. We also show that the characterization of contextual links can be specified within the same framework. In the present work, we use the term 'formalization' in the sense of 'specification' within a grammar formalism as a basis for implementation. Thus, it is distinct from the level of formalization commonly pursued by formal semanticists.

Section 4.1 introduces and discusses CCG. Topics include a review of several motivating cases, derivations of a simple sentence, a summary of the standard framework, and some extensions of the framework. We also discuss computational properties of CCGs in Subsection 4.1.5. Then, Section 4.2 discusses specification of contextual links. Finally, the idea of structured meaning is integrated with the framework (Section 4.3).

4.1 Combinatory Categorial Grammar

This section introduces the CCG framework. We start from simple examples of derivations in CCG. Then, a summary of standard CCG and two types of extensions are presented. Finally, generative power and theoretical parsing efficiency are discussed.

4.1.1 Motivation

In Section 3.4, we observe that tight syntax-semantics relation in the Montagovian tradition [Montague, 1974] can simplify the analysis of information structure. We have also argued that this direction can be extended to include 'non-traditional' constituency such as subject-verb sequence, e.g., "*Felix praised*" as in (18). These points can be captured by a group of extended Categorial Grammars including Combinatory Categorial Grammar (CCG) [Ades and Steedman, 1982; Steedman, 1985; Dowty, 1988].[1]

Let us first explore several motivating cases involving 'non-traditional' constituency. Probably the most discussed aspect of non-traditional constituency is in association with coordination. For example, the following pattern [Steedman, 1996, (86), p. 37] poses a problem to most traditional grammar formalisms because subject-verb sequence cannot be readily recognized.

(104) { <u>Keats steals</u> and <u>Chapman eats</u> } apples.

Similar situations are observed in other languages as well. The following is a coordination of NP sequences in Japanese [Komagata, 1997a, (1)].

(105)		<u>John-ga</u>	<u>Mary-o</u>	,	<u>Ken-ga</u>	<u>Naomi-o</u>		tazuneta.
	{	John-NOM	Mary-ACC	(and)	Ken-NOM	Naomi-ACC	}	visited

"John visited Mary and Ken, Naomi."

Again, this is a problem for most grammar formalisms.

Many constructions involving 'extraction' are often handled with the help of empty categories, i.e., 'trace'. Let us now turn to the following example [Steedman, 1996, modified from (34), p.59]:

(106) the apples which <u>I think Keats likes</u>

A textbook-style analysis [Haegeman, 1991, p. 370] of such a case may look like the following:

(107) [whom$_i$ [I think Keats likes t_i]]

But this type of analysis is not re-usable for Right Node Raising (RNR) [Steedman, 1996, (35), p. 59].

(108) { <u>I think Keats likes,</u> but <u>you say he detests</u> }, the man in the grey flannel suit.

[1]Wood [1993] is a good overview of Categorial Grammars in general.

The traditional work often assumes that the above two phenomena require separate analyses. But they are parallel with respect to both surface structure and interpretation. Thus, it is desirable to have a grammar that can demonstrate this point [Steedman, 1996, p. 59].

Non-traditional constituency is also observed in relation to prosodic structure in English [Steedman, 1991a, (49), p. 282].

(109) *Q*: I know what Fred **cooked**. But then, what did he **eat**?

A: [Fred $\underset{\text{L+H* LH\%}}{\textbf{a-ate}}$] [the $\underset{\text{H* LL\%}}{\textbf{beans}}$].

The symbols below (*A*) indicate intonation. L+H* and H* tones are argued to be theme and rheme markers, respectively [Steedman, 1991a]. The traditional approach is forced to take a position that prosodic structure is independent of syntactic structure (this is in fact the line taken by many recent researchers, see the discussion and references in Steedman [1999, Chapter 5]). But Steedman [1999] argues that it is more intuitive if the prosodic structure is close to syntactic structure.

In addition, there is an interesting observation about prosodic structure in Japanese. Kubozono [1993, p. 3] analyzes Japanese prosody in detail and discovers that right-branching cases, but not left-branching ones, are marked. This suggests that the prosodic structure in Japanese has a left-branching structure as in the case of English. This is striking from the view point that Japanese syntax is strictly head-final, i.e., right-branching [Kubozono, 1993, p. 158]. This situation for a simple Subj-Obj-Verb pattern is shown below.

(110) *a*. Prosodic phrasing: [[Subj Obj] Verb]

b. Syntactic structure: [Subj [Obj Verb]] (as assumed by Kubozono)

Kubozono [1993, p. 222] proposes a solution to adjust prosodic structure to match right-branching syntactic structure. Although discussion on this point is beyond the scope of the current work, we can also address the problem from the syntactic side. Namely, the assumption that syntactic structure in Japanese is categorically right-branching may not be correct. In fact, we have already observed in (105) that Subj-Obj sequence (NP sequence) can be a constituent for the coordination purpose. Thus, it may well be the case that the observed prosodic phrasing directly corresponds to the syntactic structure recognized by CCG.

There is yet another point from psycholinguistic view point, i.e., incremental processing. If a human processes utterances in the left-to-right order, the string consists of the subject and the verb in an utterance in English must have been (at least partially) processed before the object is encountered [Ades and Steedman, 1982].

Among the family of Categorial Grammars that naturally capture non-traditional constituency, we adopt CCG for the following reasons. Various linguistic analyses have been undertaken within the framework, e.g., coordination/extraction [Steedman, 1985; Dowty, 1988; Steedman, 1996], interface to prosody (in English) and information structure [Steedman, 1991a; Prevost, 1995; Hoffman, 1995]. The standard version of CCG [Steedman, 1996] has a desirable generative capacity, i.e., mildly context-sensitive and weakly equivalent to Lexicalized Tree-Adjoining Grammar (LTAG), [Vijay-Shanker and Weir, 1994] and is polynomially parsable [Vijay-Shanker and Weir, 1993]. Several forms of extensions have been proposed and their generative power and parsing efficiency have been analyzed [Hoffman, 1995; Komagata, 1997a]. Yet another area is relation to quantifier scope [e.g., Park, 1996]. In addition, a practical parser has been constructed [Wittenburg, 1986; Komagata, 1997a]. Not all of these aspects have been explored in other related extended Categorial Grammars such as Lambek Calculus [Lambek, 1988, originally published in 1958] and Unification Categorial Grammar (UCG) [Zeevat, 1988].

4.1.2 Derivation Examples

Traditional Case

In this subsection, we first introduce the basics of CCG through a 'traditional' derivation of "Felix praised Donald". Then, the second half presents a 'non-traditional' derivation of the same sentence.

First, the lexical entry for each word is specified in the following manner:

(111) Lexicon: $\langle$phonological form$\rangle \underset{\text{(assignment)}}{:=} \langle$category$\rangle$
where $\langle$category$\rangle$:= $\langle$syntactic type$\rangle$: $\langle$semantic representation$\rangle$
For example,

a. Felix := $NP : felix'$

b. praised := $(S\backslash NP)/NP : \lambda X.\lambda Y.praise'(X)(Y)$

c. Donald := $NP : donald'$

For simplicity, we may also refer to syntactic type as category where no confusion arises. The complex category $(S\backslash NP)/NP$ can be read that it first takes an NP category to the right and then another NP category to the left ('result-leftmost' representation).[2] We assume left-associativity for the slash symbols '$/$' and '$\backslash$'. Thus, we may abbreviate $(S\backslash NP)/NP$ as $S\backslash NP/NP$ without parentheses. Each category may be associated with a finite, non-recursive set of features such as $NP_{[agr=(3pers,sing,nom)]}$ although not shown in this chapter to avoid complexity. Steedman [1996, Section 2.1] discusses the use of features for agreement and binding. Features are used extensively in the implementation (Chapter 6).

We first see the derivation of the VP "*praised Donald*". Syntactically, this process can be seen as a result of **functional application** to the two categories (informally, a cancelation of the outermost argument) as follows:

(112) *a*. Rule: Functional Application (in categorial form)

$$X/Y \qquad Y \quad \Longrightarrow \quad X$$

b. Instance of rule application

praised	Donald		praised Donald
$S\backslash NP/\underline{NP}$	$\underline{NP}$	$\Longrightarrow$	$S\backslash NP$

Note: Underline may be used to indicate the cancelation of the involved categories.

Semantically, the process is an instance of functional application (β-reduction in the lambda-calculus term) as follows:

(113) *a*. Rule: functional application (β-reduction)

$$\lambda X.f(X) \qquad a \quad \Longrightarrow \quad f(a)$$

b. Instance of rule application

praised	Donald		praised Donald
$\lambda X.\lambda Y.praise'(X)(Y)$	$donald'$	$\Longrightarrow$	$\lambda Y.praise'(donald')(Y)$

The next step of deriving a sentence from the subject and the VP is analogous except for the directionality of the functor.

(114)

	Felix	praised Donald		Felix praised Donald
Syntactically:	$\underline{NP}$	$S\backslash\underline{NP}$	$\Longrightarrow$	S
Semantically:	$felix'$	$\lambda Y.praise'(donald')(Y)$	$\Longrightarrow$	$praise'(donald')(felix')$

[2]The result-leftmost representation is seen in contrast to the European tradition [e.g., Morrill, 1994], where argument categories are placed either to the left or right depending on the slash direction as in $NP\backslash S/NP$. It is more difficult to read off the type in this notation.

This way, surface structure and semantic representation can be associated in a straightforward manner following the Montagovian tradition.

Non-traditional Case

For the non-traditional derivation, we need two more rules: *type raising* and *functional composition*. Intuitively, **type raising** is an operation of transforming, say, an NP into a functor category that takes a VP as its argument. This shift was originally motivated to capture the property of quantified NPs whose quantifier scopes over a VP [Montague, 1974]. Type-raising the individual type such as *Felix* also allows us to coordinate it with a quantified NP, as in "*Felix and some dogs*". The following example illustrates the application of type raising to an NP, *Felix*:

(115) *a*. Syntactically: $X \Longrightarrow S/(S\backslash X)$

e.g., $NP \Longrightarrow S/(S\backslash NP)$

Note: We may abbreviate $S/(S\backslash NP)$ as $NP^{\uparrow}$.

b. Semantically: $a \Longrightarrow \lambda F.F(a)$

e.g., $felix' \Longrightarrow \lambda F.F(felix')$

Functional composition (for CCG) is basically the same as its mathematical counterpart. In mathematics, functional composition provides a means of analyzing function application in an associative way: e.g., $f(g(X)) = [f \circ g](X)$. In CCG, functional composition enables the grammar to recognize subject-verb sequence as a constituent, still looking for an object. Assuming that type raising is applied to *Felix*, we describe the next step involving functional composition as follows:

(116) *a*. Rule: Functional Composition (in categorial form)

$$X/Y \quad Y/Z \Longrightarrow X/Z$$

b. Instance of rule application

Felix	praised		Felix praised
$S/\underline{(S\backslash NP)}$	$\underline{(S\backslash NP)}/NP$	$\Longrightarrow$	S/NP

Semantically, the process is as follows:

(117) *a*. Rule: functional composition (in mathematical term)

$\lambda Y.f(Y)$	$\lambda X.g(X)$	$\Longrightarrow$	$\lambda X.f(g(X))$	
f	g		$f \circ g$	(in another notation)

b. Instance of rule application[3]

Felix	praised		Felix praised
$\lambda f.f\,(felix')$	$\lambda X.\lambda Y.praise'\,(X)\,(Y)$	$\Longrightarrow$	$\lambda X.praise'\,(X)\,(felix')$

The resulting category, "$S/NP : \lambda X.praise'\,(X)\,(felix')$", is the CCG representation of the non-traditional constituent we are concerned with. Another step of functional application leads to derive exactly the same category including the semantic representation.

(118)

	Felix praised	Donald		Felix praised Donald
Syntactically:	$S/\underline{NP}$	$\underline{NP}$	$\Longrightarrow$	S
Semantically:	$\lambda X.praise'\,(X)\,(felix')$	$donald'$	$\Longrightarrow$	$praise'\,(donald')\,(felix')$

This demonstrates that CCG is capable of recognizing non-traditional constituents needed for our analysis of information structure. Staying with the Montagovian tradition, the grammar still tightly interfaces surface syntactic structure and semantic representation. This allows us to provide an interface not only between linguistic expression and semantic representation, but also between semantic representation and our notion of referent, a unit of information structure, as we will see shortly.

4.1.3 Standard CCG: A Summary

In CCG, like other lexicalized formalisms such as Lexicalized Tree-Adjoining Grammar (LTAG) [Schabes, 1990], much of the syntactic information is stored in the lexicon. But a great deal of syntactic generality comes from the use of a small number of combinatory rules introduced in the previous section. Mostly following the framework outlined in Steedman [1996], we summarize our combinatory rules as follows:

(119) Functional application:

					Rule symbol
a.	$X/Y : f$	$Y : a$	$\Longrightarrow$	$X : f\,(a)$	$(>)$
b.	$Y : a$	$X\backslash Y : f$	$\Longrightarrow$	$X : f\,(a)$	$(<)$

(120) Functional composition:

[3] $[\lambda f.f\,(felix')]\,(\lambda x.\lambda y.praise'\,(x)\,(y)) = [\lambda x.\lambda y.praise'\,(x)\,(y)]\,(felix') = \lambda x.praise'\,(x)\,(felix')$

$$
\begin{array}{llllll}
a. & X/Y:f & Y/Z:g & \Longrightarrow & X/Z:\lambda X.f(g(X)) & (>B) \\
b. & Y\backslash Z:g & X\backslash Y:f & \Longrightarrow & X\backslash Z:\lambda X.f(g(X)) & (<B)
\end{array}
$$

To be precise, the combinatory rules are rule schemata. The meta-variables X, Y, etc. need to be instantiated for a particular choice of categories. We will also call the application of these rule schemata **combination**.

There are directional variations for functional composition as follows:

(121) Functional composition (crossing variation):

$$
\begin{array}{llllll}
a. & X/Y:f & Y\backslash Z:g & \Longrightarrow & X\backslash Z:\lambda X.f(g(X)) & (>B\times) \\
b. & Y/Z:g & X\backslash Y:f & \Longrightarrow & X/Z:\lambda X.f(g(X)) & (<B\times)
\end{array}
$$

The backward variety (b) is used for combining certain verb arguments in a non-traditional way, as in the following example:

(122) John put on the table "John put on the table"

$$
\begin{array}{lllll}
 & PP & & & \\
 & \Downarrow & & & \\
\underline{S/PP}/NP & S\backslash(\underline{S/PP}) & \Longrightarrow & S/NP & (<B\times)
\end{array}
$$

The resulting phrase may be a part of a relative clause "*the book John put on the table*" or a heavy NP shift "*John put on the table that incredibly-heavy dictionary*". The forward variety is not used in English [Steedman, 1996, p. 53], but is a basis for long-distance fronting in Japanese.

Generalized versions of functional composition with multiple arguments, e.g., "$X/Y \quad Y/Z_1/Z_2 \Longrightarrow X/Z_1/Z_2$" are also used in Steedman [1996, p. 35]. These can be labeled as $>B^k$ and $<B^k$ where k corresponds to the number of 'passed' arguments $Z_1,...,Z_k$. In this regard, the basic functional composition above may be written as $>B^1$ and $<B^1$, and functional application as $>B^0$ and $<B^0$.

We repeat the basic form of type raising as follows:

(123) Type raising:

$$
\begin{array}{lllll}
a. & X:a & \Longrightarrow & S/(S\backslash X):\lambda f.f(a) & (>T) \\
b. & X:a & \Longrightarrow & S\backslash(S/X):\lambda f.f(a) & (<T)
\end{array}
$$

Here, we assume syntactic type raising (dynamically applied during derivation). But it is also possible to type raise categories in the lexicon [Steedman, 1991b, p. 75].

Finally, we give the coordination rule schemata as follows:

(124) Coordination:

$$X : f \qquad\qquad Coord : c \qquad\qquad X : g$$

$$\Longrightarrow \quad X : \begin{cases} c\,(f)\,(g) & (<\Phi^0>) \\ \lambda X.c\,(f\,(X))\,(g\,(X)) & (<\Phi^1>) \\ \lambda X.\lambda Y.c\,(f\,(X)\,(Y))\,(g\,(X)\,(Y)) & (<\Phi^2>) \end{cases}$$

Separate semantic cases are needed for different arities.

In addition, there is another type of combinatory rule called ‘substitution’, “$(X/Y)/Z \quad Y/Z \Longrightarrow X/Z$” (one direction), used for the analysis of parasitic gap, also a part of the CCG framework [Steedman, 1996, p. 39], but not used in the current work.

4.1.4 Extensions of CCG

While the standard CCG is capable of dealing with a wide range of linguistic constructions, there are cases where some extensions are called for. We present two such cases in this section, namely Multiset-CCG [Hoffman, 1995] and CCG-GTRC [Komagata, 1997c; Komagata, 1997a].

Multiset-CCG

Languages like German and Turkish are known for their extremely flexible word order. Becker et al. [1991] observe that German long-distance fronting (scrambling) involved in this phenomenon has the following properties: (i) there is no bound on the distance of movement and (ii) there is no bound on the number of constituents that are moved. The same situation is also observed in Turkish. Hoffman [1995, (11), p. 46] follows Becker et al. [1991] and represent the phenomenon in the following form:

(125) $(NP_1 \ldots NP_m)_{scrambled}\ V_m \ldots V_1$

Hoffman [1995] then argues that the competence grammar must be able to capture the set of all of these scrambled strings. She points out that standard CCG does not have this property and that a more powerful grammar is called for. Hoffman [1995] develops a formalism called ‘Multiset-CCG’. The idea behind Multiset-CCG is that the ‘bags’ of arguments of different verbs can mix freely. The term ‘bag’ is used here to indicate that they are multisets, allowing duplicate entries without order, and neither sets (non-redundant) nor lists (ordered).

In support for her choice of Multiset-CCG, Hoffman [1995, Section 2.4] discusses several types of extensions of standard CCG. One of such extension is to use type raising and backward crossing composition (see the previous subsection) to partially cover the case of (125). The following example shows the case where the NP arguments of the inner verb V_1 are fronted to the sentence-initial position.

(126)

$$
\begin{array}{ccccc}
NP_{1b} & NP_1 & NP_{2a} & V_1 & V_2 \\
S/\underline{(S\backslash NP_{ACC})} & S/\underline{(S\backslash NP_{NOM1})} & S/\underline{(S\backslash NP_{NOM2})} & \underline{S}\backslash NP_{NOM1}\backslash NP_{ACC} & S\backslash NP_{NOM2}\backslash \underline{S} \\
\hline
 & & & \multicolumn{2}{c}{<B^2} \\
 & & & \multicolumn{2}{c}{\underline{S\backslash NP_{NOM2}}\backslash NP_{NOM1}\backslash NP_{ACC}} \\
 & & \multicolumn{3}{c}{>B^2_{\times}} \\
 & & \multicolumn{3}{c}{\underline{S\backslash NP_{NOM1}}\backslash NP_{ACC}} \\
 & \multicolumn{4}{c}{>B_{\times}} \\
 & \multicolumn{4}{c}{\underline{S\backslash NP_{ACC}}} \\
\multicolumn{5}{c}{>} \\
\multicolumn{5}{c}{S}
\end{array}
$$

But Hoffman [1995, p. 34] points out that this approach cannot deal with scrambled coordination such as the following:[4]

(127)

$$
\begin{array}{cccccc}
NP_{ACC} & NP_{NOM} & \& & NP_{ACC} & NP_{NOM} & V \\
S/(S\backslash NP_{ACC}) & S/(S\backslash NP_{NOM}) & & S/(S\backslash NP_{ACC}) & S/(S\backslash NP_{NOM}) & S\backslash NP_{NOM1}\backslash NP_{ACC} \\
\multicolumn{2}{c}{>B} & & \multicolumn{2}{c}{>B} & >B \\
\multicolumn{2}{c}{S/(S\backslash NP_{ACC}\backslash NP_{NOM})} & & \multicolumn{2}{c}{S/(S\backslash NP_{ACC}\backslash NP_{NOM})} & \\
\multicolumn{5}{c}{\rule{0pt}{0pt}} & <\&> \\
\multicolumn{6}{c}{*}
\end{array}
$$

But there is a simple solution. We can admit that local scrambling is a reflection of the ambiguous verb categories between $S\backslash NP_{NOM}\backslash NP_{ACC}$ and $S\backslash NP_{ACC}\backslash NP_{NOM}$ [Baldridge, 1998, Section 3.2]. Then, the above situation can be handled within the framework of standard CCG.[5]

[4] This situation is the same in Japanese.

[5] But there is an even worse possibility. The following example is acceptable in Japanese (or Korean).

(1)		Donald-o	Felix-ga		,		Mickey-ga	Roger-o		hometa.
	{	Donald-ACC	Felix-NOM	}	CONJ	{	Micky-NOM	Roger-ACC	}	praised

"Felix praised Donald, and Mickey [praised] Roger."

We cannot discuss this situation any further in the current work.

There also is a warning against the power of Multiset-CCG. Joshi et al. [1994] question the property (125) and point out that two levels of argument mixture can be covered within the framework of TAG. Their argument is that if a competence grammar can characterize the practical bound on a phenomenon, it is more appropriate to assume such a grammar for description. In either case, since we do not readily encounter this situation in our English and Japanese data, we are not committed to Multiset-CCG.

CCG-GTRC

There is another extension of CCG, which involves the use of variables in type raising [Komagata, 1997c; Komagata, 1997a]. This extension is motivated by the constituency of NP sequences in Japanese.[6] A sequence of NPs can form a non-traditional constituent with respect to coordination, as seen in (105) repeated below.

(129) John-ga Mary-o , Ken-ga Naomi-o tazuneta.
{ John-NOM Mary-ACC (and) Ken-NOM Naomi-ACC } visited
"John visited Mary and Ken, Naomi."

This is a very common construction frequently found in real text (there is an example (1) on p. 211). Unfortunately, this case has been neglected from legitimate analyses. By type-raising the NPs, CCG can provide a straightforward analysis of NP sequences corresponding to (129), as shown below.

(130) John- ga$_{\text{NOM}}$ Mary- o$_{\text{ACC}}$

NP NP

$\Downarrow$ $\Downarrow$ type raising

$S/(S\backslash NP)$ $(S\backslash NP)/((S\backslash NP)\backslash NP)$

functional composition

$S/((S\backslash NP)\backslash NP)$

The NPs are assigned type-raised categories associated with the basic category NP, and these functions can compose to derive another function category, which represents the NP-NP sequence. The two instances of such a category can then be coordinated and/or take the transitive verb category, $(S\backslash NP)\backslash NP$, as the argument to derive the sentence category S.

[6] Earlier analyses of Japanese using Categorial Grammar include [Kurahone, 1983 (focus on verb semantics)] and [Whitelock [1988] (focus on morphology)].

If the length of NP sequence is not bounded, we are forced to extend the formalism with a mechanism that can handle NP sequences of potentially infinite length. A natural move is to use variables in type raising as follows:

(131) Variable type raising:

$$a.\quad X : a \quad\Longrightarrow\quad \mathsf{T}/\left(\mathsf{T}\backslash X\right) : \lambda F.F\,(a)$$

$$b.\quad X : a \quad\Longrightarrow\quad \mathsf{T}\backslash\left(\mathsf{T}/X\right) : \lambda F.F\,(a)$$

Note: T is a variable over categories.

We may also abbreviate the above two with corresponding directionality as follows:

(132) $$X : a \quad\Longrightarrow\quad \mathsf{T}\langle^{/}_{\backslash}\left(\mathsf{T}\rangle^{\backslash}_{/} NP_1\right) : \lambda F.F\,(a)$$

Then, unbounded length of NP sequence can be analyzed as a constituent, e.g., $\mathsf{T}\langle^{/}_{\backslash}\left(\mathsf{T}\rangle^{\backslash}_{/} X_1 \ldots \rangle^{\backslash}_{/} X_k\right)$. The resulting category may be called ‘generalized type-raised categories’ (GTRC). We abbreviate the extension of CCG with GTRC as CCG-GTRC.

4.1.5 Generative Power and Theoretical Parsing Efficiency

The most notable milestone regarding the generative power of CCG in relation to other formalisms including Tree-Adjoining Grammar (TAG) [Joshi et al., 1975; Joshi, 1985] and Linear Index Grammar (LIG) is Vijay-Shanker et al. [1986] and Weir and Joshi [1988], also published as Joshi et al. [1991], and more recently reworked as Vijay-Shanker and Weir [1994]. These formalisms are among the class called ‘mildly context-sensitive grammars’. The finding of these papers is that these formalisms are all weakly equivalent (i.e., with respect to string generation capacity but not structural isomorphism). This finding is also important in relation to the processor. There is a class of automata called Embedded Push-down Automata that processes exactly the class of these grammars [Vijay-Shanker, 1988, Chapter 3].

All three variants of Multiset-CCG developed by Hoffman retain desirable formal properties: they are mildly context-sensitive.[7]

As for CCG-GTRC, one might be concerned about the use of variables that may introduce unexpected effects. It is not apparent whether the resulting formalism retains the same computational

[7] One variant of Multiset-CCG (Curried Multiset-CCG) is more powerful than the standard CCG, but the other two (Pure and Prioritized Multiset CCG) are incomparable to the standard CCG in this respect Hoffman [1995, Section 4.1.2].

properties as before. A general use of variables in a variant of categorial grammar makes it difficult even to demonstrate decidability [Emms, 1993]. But, since the use of variables in CCC-GTRC is fairly limited, an intuition is that it does not much increase the power of the formalism. My earlier paper [Komagata, 1997c] investigated all the possible occurrences of GTRCs in combinatory rules and argued that, with certain conditions, CCG-GTRC is weakly equivalent to the standard CCG. The main idea of the weak equivalence between CCG-GTRC and the standard CCG is that every derivation in CCG-GTRC can be simulated in the way the languages generated by the two grammar instances are exactly the same. The simulation uses the idea related to 'wrapping' [Bach, 1979; Dowty, 1979]. The propositions are proved by an extensive use of mathematical induction on the structure of derivation.

Another issue is theoretical parsing efficiency . Naturally, it is highly desirable that our grammar exhibits some polynomial parsing algorithm, as in the case of Context-Free Grammar (CFG), where CKY-style parsing algorithm has the $O\left(n^3\right)$ worst-case performance [Aho and Ullman, 1972, p. 317, for analysis]. But, since the number of categories in a CKY table cell is not bounded for CCG, a naive CKY-style algorithm for CCG does not have a polynomial bound.

There is a potential computational problem with accepting a wider variety of constituents. As seen above, multiple derivations may derive multiple instances of a single category (e.g., through a traditional and a non-traditional derivations). This situation is often called **spurious ambiguity** [Wittenburg, 1986]. If this kind of ambiguity is left untreated in the process, the number of categories being processed can easily explode in an exponential manner. Theoretical and practical solutions to this situation are discussed in Subsection 6.2.2.

Through a careful study of the properties possessed by CCG categories, Vijay-Shanker and Weir [1990] present a worst-case polynomial parsing algorithm for CCG. Later, they presented a more general algorithm covering several mildly context-sensitive grammar formalisms [Vijay-Shanker and Weir, 1993]. Their polynomial parsing algorithm employs a structure sharing technique [Billot and Lang, 1989; Dymetman, 1997] for efficient storage of potentially unboundedly-long categories. Crucially, the proposed structure sharing does not suffer from the existence of spurious ambiguities. Although this result alone does not demonstrate the practicality of the formalism, one without this property is unlikely to be practical.

The CCG formalism ('standard' CCG) used in the above comparison consists of combinatory rules: functional application and functional composition of fixed k. The coordination is not included as a rule but basically the same effect can be achieved by categories such as $S\backslash S/S$.[8]

All three variants of Multiset-CCG developed by Hoffman [1995] are polynomially parsable. Similarly, my earlier paper [Komagata, 1997a] shows that CCG-GTRC is polynomially parsable. The worst-case polynomial algorithm for CCG-GTRC is an extension of the polynomial algorithm for the standard CCG. The algorithm for CCG-GTRC also utilizes the idea of structure sharing for efficient storage and retrieval of GTRCs. A more detailed discussion of CCG-GTRC including formal and computational properties is found in Appendix A.

4.2 Specification of Contextual-Link Status

In this section, we confirm that the specification for contextual-link status can be formalized with the CCG framework. The discussion includes: discourse status, domain-specific knowledge, and linguistic marking for contextual link and information structure.

Discourse Status

For each CCG-constituent recognized by the grammar, there are corresponding semantic representations. These semantic representations can be used as discourse referents in a general sense (Subsection 3.2.2).

In order to formalize the notion of discourse-oldness, we need the following: (i) a mechanism to store all these objects and (ii) a mechanism to search through the storage for redundancy. Identification of discourse-old status checks the applicability of the identity relation on semantic representations between the semantic representation under consideration and one in the storage.

As we have mentioned earlier (Section 2.2), lack of exact reference resolution is not very crucial to information-structure analysis, especially for the case where the expression linguistically marks discourse-oldness. If the contextual-link status can be determined only through discourse-oldness and this depends on exact reference resolution, the formalization based on the use of

[8]While the coordination *schema* can deal with any category, coordination *category* can deal with only a closed set of categories. But this is not a limitation in practice. Another point is that the coordination category can compose with a functor. For example, $NP\backslash NP/NP$ may compose with a determiner NP/N if there is no further restriction.

semantic representations (rather than semantic values) would fail to recognize the correct discourse status. But, as we will see in Chapter 7, such cases rarely occur in our domain. The difficulty associated with INFERRABLE is far more common.

Domain-specific Knowledge

For the domain-specific knowledge, we only assume that physician(s) and patient(s) are available in the initial context regardless of the discourse. We can formalize this by simply asserting properties *physician* and *patient* in the initial context. Then, when these nouns are used in the text, they appear as if they were discourse-old, and can be identified as a contextual link.

Linguistic Marking

Linguistic marking is the case where the grammatical information is required. The mechanism of contextual-link assignment and projection is straightforward. For example, definite determiners of a category NP/N can assign a contextual-link status as specified on the result category, NP in this case, as shown below.

(133)		Definite determiner	Noun
	Example:	*the*	*door*
	Syntactic type:	$\underset{CL\quad -}{NP/N}$	$\underset{CL \text{ or } NL}{N}$
	Syntactic type:	$\underset{\boxed{CL}}{NP}$	

The specification of the contextual-link status, which may be realized as a feature, is shown below the result category, NP. The indefinite article is analogous, but it assigns a non-contextual-link status instead.

In Section 3.3, we have also discussed special cases with definite and indefinite articles. This introduces more complication to the above story. First, definite expressions with a special pre-nominal modifier such as *first* and *last* may be non-contextual links (p. 65). Thus, to be precise, contextual-link assignment of a definite determiner must check the lexical instantiation of the argument (e.g., through semantics). It should not categorically assign a contextual-link status if the semantics involves one of the special pre-nominal modifiers. But we do not formalize this particular aspect because this is not critical in our experiment data (Chapter 7).

Another exceptional case is indefinite INFERRABLES (p. 68). The point was that the main class of indefinite INFERRABLES are lexically marked, i.e., as two-place common nouns. Thus, these nouns can be marked as a contextual link. As in the case of "*a page (of a book)*" (65*b*), a countable two-place noun may be attached with an indefinite article. According to the description of indefinite article above, it is a non-contextual-link assigner. But there may be another type of indefinite article that contrasts with other quantifiers, but does not assign non-contextual-link status to these two-place nouns.[9]

Utterance-initial modifiers are also similar to the definite determiner except that only the utterance-initial variety, i.e., S/S, assigns a contextual-link status. The post-modifier type $S\backslash S$ does not have any special function.

Projection of contextual-link status can be done by using variable unification, as shown for a non-definite determiner below.

(134)

	Determiner	Noun
Example:	*many*	*researchers*
Syntactic type:	NP_X/N_X	N_{Status}
Syntactic type:	$NP_{\boxed{Status}}$	

This class includes auxiliary verbs and coordinators (for multiple arguments).

The other case of projection from the functor is similar. But the contextual-link information is carried over from its own contextual-link status as follows:

(135)

	Pre-modifier	Noun
Example:	*exercise*	*program*
Syntactic type:	N_{Status}/N_{-}	$N_{CL \text{ or } NL}$
Syntactic type:	$N_{\boxed{Status}}$	

As we have discussed in Subsection 2.3.3, direct information-structure marking is a matrix-level phenomenon. Thus, our grammar must be able to distinguish between the matrix and embedded environment. One way to do this is to assume utterance boundary categories, say $\$/S$ and/or

[9]Combined with the analysis of indefinite generics (p. 68), there is another possibility that an indefinite article actually projects the contextual-link status of the argument. If this is the case, the distinction between indefinite articles and other non-definite determiners disappears.

$\$\backslash S$, and assign the matrix feature only to the immediately composed S. We may associate the latter category with the period for the case of written text. A possible semantics for such a category is 'assertion' of the proposition corresponding to the category S. This point naturally connects to dynamic semantics (Subsection 2.3.2).

The special constructions in English discussed in Section 3.3.2 involves linguistic marking of both contextual link and information structure. A simpler case is pseudocleft. We analyze it simply as a contextual-link assigner, as in the case of a definite determiner. The subordinator *since* as a theme marker can be specified for the status with a feature, which may be checked at the time the information structure is identified.

VP preposing and inversion mark the "$Theme - Rheme$" partition. These cases require special syntactic types that license these constructions. For example, inversion of a *PP* may need a special *PP* category such as "$S/\underset{\text{subj}}{NP}/(S\backslash \underset{\text{verb}}{NP}/PP)/\underset{\text{arg of }PP}{NP}$". There are different ways of characterizing such a construction, but this category assumes that the exceptional behavior comes from the fact that *PP* is preposed and not from the verb or the subject. Since this is a matrix-level phenomena, we may require that this category is available only immediately to the right of utterance-boundary category $\$/S$. This can be done by, e.g., making the preposed *PP* "$S/\underset{\text{subj}}{NP}/(S\backslash \underset{\text{verb}}{NP}/PP)/\underset{\text{arg of }PP}{NP}\backslash(\$\backslash S)$". Once inversion is available only at the matrix level, we only need to mark the *PP* as a contextual link. As the *PP* is involved in the last semantic composition, *PP* is identified as a theme.

The conditions involved in topicalization, focus movement, and left dislocation are rather complex. Three different cases of information-structure marking must be considered. The category for the preposed NP are (i) $S/(S/NP)$ for the case of topicalization and focus movement and (ii) S/S for the case of left dislocation.[10] Since these are matrix-level phenomena, we can use contextual-link assignment for specifying information structure. We have seen that these constructions weakly partition theme and rheme. Thus, the preposed NP for all three cases may either set or reset the contextual-link status on itself. The contextual-link status of the remaining part of the utterance is determined in relation to that of the preposed NP.

For the case of topicalization, the topicalized NP is a contextual link and must be a part of the theme. The remaining part must contain a rheme due to the assumption (49) in Section 2.2. But it may also contain a part of the theme. This is consistent with the analysis (74) repeated below.

---

[10] Other cases which prepose non-NPs are analogous.

(136) [This dream] [I've had t [maybe three, four times]]

$Theme$ $Rheme$

For the case of focus movement, the preposed NP is assigned a non-contextual-link status. Thus, it must be a part of the rheme. If the remaining part is a contextual link, a "$Rheme-Theme$" pattern emerges as shown in (76) repeated below.

(137) [**A bite**] [he wouldn't eat t]

$Rheme$ $Theme$

Summary

While the inference behind a contextual link can be complex, the three conditions for identifying a contextual link can be formalized within a grammar in a fairly straightforward manner.

4.3 Integration of Structured Meaning

One way in which we have departed from standard Montagovian, and from the standard CCG as well, is the use of structured meaning as semantic representation. As has been discussed in Section 3.5, this approach has the advantage of keeping the simple binomial information structure and being systematic.

The description in Section 3.5 demonstrates that structured meaning *can* be used for representing discontiguous information structure but does not show how it can be done. Further investigation of the way structured meanings are derived shows that it is not a simple issue.

In the following subsection, we demonstrate that the structured-meaning analysis approach presented in Section 3.5 can be integrated within CCG in a precise manner. In the second subsection, we demonstrate that the same mechanism can be applied to the analysis of 'gapping', i.e., a construction of the form "*Harry will buy bread, and Barry, potatoes*".

4.3.1 Composition of Structured Meanings

In our case, the structured meaning is adopted to capture the contrast between a contextual link and a non-contextual link. Its representation is as follows: $\left\langle Contextual\text{-}link, Non\text{-}contextual\text{-}link \right\rangle$ or $\langle C, N \rangle$ for short. We now assume that each constituent is associated with a structured meaning in addition to a non-structured representation. The semantic composition in the general case is to combine $\langle C_1, N_1 \rangle$ and $\langle C_2, N_2 \rangle$ to obtain $\langle C', N' \rangle$ where C' and N' must be determined from the input components depending on the condition. In the following, we check distinct cases depending on the type of input. We denote semantic composition of structured meaning (and also its component) as "$\langle C_1, N_1 \rangle + \langle C_2, N_2 \rangle$" (here, '$+$' is used to separate the two categories). Two special cases are $\langle C, - \rangle$ and $\langle -, N \rangle$ where the entire semantic representation is either a contextual link or a non-contextual link. '$-$' here indicates a null component. Although this approach is naturally more complicated than the case without structured meaning, all possibilities can be completely specified.

Composition Type: $\langle -, N_1 \rangle + \langle -, N_2 \rangle$

This case is exactly like the usual semantic composition. We can simply operate on the non-link field as the following example shows.

(138)

$$\frac{\overset{\text{Felix}}{\left\langle -, \underset{\text{non-link}}{\lambda P.P\,(felix')} \right\rangle} \quad \overset{\text{praised}}{\left\langle -, \underset{\text{non-link}}{\lambda X.\lambda Y.praise'\,(X)\,(Y)} \right\rangle}}{\left\langle -, \underset{\text{non-link}}{\lambda X.praise'\,(X)\,(felix')} \right\rangle}$$

In this case, the component $\lambda X.praise'\,(X)\,(felix')$ is obtained by functional composition. But, in general, either functional application or composition may apply.

Composition Type: $\langle C_1, - \rangle + \langle -, N_2 \rangle$

This is a representative case of forming a structured meaning.

(139)

$$\frac{\overset{\text{Felix praised}}{\left\langle \lambda X.praise'\,(X)\,(felix'), - \right\rangle} \quad \overset{\textbf{Donald}}{\left\langle -, donald' \right\rangle}}{\left\langle \underset{\text{contextual link}}{\lambda X.praise'\,(X)\,(felix')}, \underset{\text{non-link}}{donald'} \right\rangle}$$

Further composition involving this type of structured meaning gets more complicated. The analysis for the mirror image, $\langle -, N_2 \rangle + \langle C_1, - \rangle$, is analogous.

Composition Type: $\langle C_1, N_1 \rangle + \langle -, N_2 \rangle$

First, we should note that the surface order of the components C_1 and N_1 for $\langle C_1, N_1 \rangle$ can be either $C_1 - N_1$, or $N_1 - C_1$. The following is an example for the $N_1 - C_1$ ordering (e.g., as a response to "Who praised who?"):

(140) **Felix** praised **Donald**

$$a.\ \frac{\left\langle -, \lambda P.P(felix') \right\rangle \quad \left\langle \lambda X.\lambda Y.praise'(X)(Y), - \right\rangle}{\left\langle \lambda X.\lambda Y.praise'(X)(Y), \lambda P.P(felix') \right\rangle} \quad \left\langle -, \lambda P.P(donald') \right\rangle$$

$$b.\ \frac{\left\langle \lambda X.\lambda Y.praise'(X)(Y), \lambda P.P(felix') \right\rangle \quad \left\langle -, \lambda P.P(donald') \right\rangle}{\left\langle \lambda X.\lambda Y.praise'(X)(Y), \lambda P.P(donald')(felix') \right\rangle}$$

Let us focus on the second semantic composition (b). The question here is how we can obtain from $\lambda P.P(felix')$ and $\lambda P.P(donald')$ the correct $\lambda P.P(donald')(felix')$, but not $\lambda P.P(felix')(donald')$. The answer is that the corresponding syntactic composition is complete with the correct semantic representation. Any non-link in the above derivation that cannot derive the correct semantics after composing with $\lambda X.\lambda Y.praise'(X)(Y)$ should be discarded. Naturally, only $\left[\lambda P.P(donald')(felix')\right]\left(\lambda X.\lambda Y.praise'(X)(Y)\right)$ can result in the correct semantics $praise'(donald')(felix')$, and not $\lambda P.P(felix')(donald')$. Thus, $\lambda P.P(felix')(donald')$ should be rejected. Similarly, the other ordering of composition $\left[\lambda X.\lambda Y.praise'(X)(Y)\right]\left(\lambda P.P(donald')(felix')\right)$ should be rejected because it does not result in the correct semantics.

If structured meaning is used only for identifying a pair of contextual-status, the above-mentioned semantic check may be sufficient. In the next subsection on 'gapping', we observe a possibility that this process may also involve syntactic types.

The following is for the other surface ordering, $C_1 - N_1$.

(141) Felix **praised** **Donald**

$$a.\ \frac{\left\langle \lambda P.P(felix'), - \right\rangle \quad \left\langle -, \lambda X.\lambda Y.praise'(X)(Y) \right\rangle}{\left\langle \lambda P.P(felix'), \lambda X.\lambda Y.praise'(X)(Y) \right\rangle} \quad \left\langle -, \lambda P.P(donald') \right\rangle$$

$$b.\ \frac{\left\langle \lambda P.P(felix'), \lambda X.\lambda Y.praise'(X)(Y) \right\rangle \quad \left\langle -, \lambda P.P(donald') \right\rangle}{\text{(see below)}}$$

The focus is again (*b*). In this case, the other derivation, "[Felix] [praised Donald]", is more favorable because both non-links are combined together without complication. Thus, if an alternative derivation is available, we do not need to consider this case. If the alternative derivation is not available for some reason, we are forced to derive $\left\langle -, praise'\,(donald')\,(felix')\right\rangle$ because there is no specification that can upgrade a non-contextual-link material to a contextual link.

Composition Type: $\langle C_1, -\rangle + \langle C_2, -\rangle$

When two contextual links are composed, the resulting phrase must be identified as contextual link by one of the three properties. This case happens when a complex phrase is discourse-old or a special linguistic marking is present. For example, if "*Felix praised*" is already in the context, the following derivation is possible.

(142)

$$\dfrac{\underset{\left\langle \lambda P.P\,(felix')\,,-\right\rangle}{\text{Felix}} \qquad \underset{\left\langle \lambda X.\lambda Y.praise'\,(X)\,(Y)\,,-\right\rangle}{\text{praised}}}{\left\langle \lambda X.praise'\,(X)\,(felix')\,,-\right\rangle}$$

If the resulting unit is not contextually-linked, "$\langle C_1, -\rangle + \langle C_2, -\rangle$" can only result in either $\langle C_1, C_2\rangle$ or $\langle C_2, C_1\rangle$. That is, only one of them can remain as a contextual link and the other is considered a non-contextual link. This pattern is the source of a contextually-linked rheme. But the proposition is not a contextual-link. In this case, the rheme (either C_1 or C_2) must be contrastive as we discussed in Chapter 2, but we do not go into this point in our formalization or implementation.

Composition Type: $\langle C_1, N_1\rangle + \langle C_2, -\rangle$

This case is in a sense a combination of the previous two cases. Let us only consider the subcase where the component ordering of $\langle C_1, N_1\rangle$ is $N_1 - C_1$. For C_1 and C_2 to be combined, the resulting unit must be a contextual link through one of the three possibilities. If the combination of C_1 and C_2 is not a contextual link as a whole, only the full contextual link would survive in the result, as shown below.

(143) **Felix** praised Donald

$$a.\quad \frac{\left\langle -,\lambda P.P\,(felix')\right\rangle \quad \left\langle \lambda X.\lambda Y.praise'\,(X)\,(Y)\,,-\right\rangle}{\left\langle \lambda X.\lambda Y.praise'\,(X)\,(Y)\,,\lambda P.P\,(felix')\right\rangle} \quad \left\langle \lambda P.P\,(donald')\,,-\right\rangle$$

$$b.\quad \frac{\left\langle \lambda X.\lambda Y.praise'\,(X)\,(Y)\,,\lambda P.P\,(felix')\right\rangle \quad \left\langle \lambda P.P\,(donald')\,,-\right\rangle}{\left\langle \lambda P.P\,(donald')\,,\lambda X.praise'\,(X)\,(donald')\right\rangle}$$

Composition Type: $\langle C_1, N_1\rangle + \langle C_2, N_2\rangle$

Here, we consider the ordering $C_1 - N_1 - C_2 - N_2$. This case could end up with $\langle C', N'\rangle$, $\langle C_1, N''\rangle$, $\langle C_2, N'''\rangle$, or $\langle -, N''''\rangle$. The condition for $\langle C', N'\rangle$ is that $C_1 + C_2$ is a contextual link for its own reason and $N_1 + N_2$ can compose to result in a legitimate category. These two intermediate results must compose to the category corresponding to the entire phrase. If $C_1 + C_2$ is not a contextual link, other cases may still apply. For the case where the result is $\langle C_1, N''\rangle$, $\langle -, N_1\rangle + \langle C_2, N_2\rangle$ should not be available. If so, the bracketing $C_1 - [N_1 - C_2 - N_2]$ is available for a simpler derivation. The case for $\langle C_2, N'''\rangle$ is analogous. The last case applies when the previous three fail.

The following is an analysis of (95) on p. 85 assuming that $\lambda X.\lambda Y.want'\,(X)\left(lose'\,(Y)\,(pro)\right)(alice')$ corresponding to "*Alice wants – to loose*" is a contextual link.

(144) [Alice wants]$_{Theme}$ [**Australia**]$_{Rheme}$ [to lose]$_{Theme}$ [the **Ashes**]$_{Rheme}$

$$\frac{\dfrac{\left\langle \lambda X.\lambda Y.want'\,(X)\,(Y)\,(alice')\,,-\right\rangle \quad \left\langle -,\lambda P.P\,(australia')\right\rangle}{\left\langle \lambda X.\lambda Y.want'\,(X)\,(Y)\,(alice')\,,\lambda P.P\,(australia')\right\rangle} \quad \dfrac{\left\langle \lambda X.lose'\,(X)\,(pro)\,,-\right\rangle \quad \left\langle -,\lambda P.P\,(ashes')\right\rangle}{\left\langle \lambda X.lose'\,(X)\,(pro)\,,\lambda P.P\,(ashes')\right\rangle}}{\left\langle \lambda X.\lambda Y.want'\,(X)\left(lose'\,(Y)\,(pro)\right)(alice')\,,\lambda P.P\,(australia')\,(ashes')\right\rangle}$$

Composition Type: Coordination

One additional case is coordination. For a coordination of the type "$\langle C_1, -\rangle + \& + \langle C_2, -\rangle$", we adopt the following condition:

(145) *a.* $\langle C', -\rangle$ if the coordination of C_1 and C_2, i.e., C', is a contextual link (where $C' = C_1 + C_2$)

b. $\langle -, N''\rangle$ otherwise (N' is the semantic representation for the entire phrase)

While a more fine-grained analysis is possible, e.g., coordination of $\langle C_1, N_1\rangle$ and $\langle C_2, N_2\rangle$, we only consider the above simple analysis.

Discontiguous Components

Up to here, we have been assuming that the components of a structured meaning are contiguous. But this is not always the case. For example, the composition of $\langle C_1, N_1 \rangle$ and $\langle -, N_2 \rangle$ (with the $N_1 - C_1 - N_2$ surface ordering) may end up with $\langle C_1, N' \rangle$ where the component N' is discontiguous. If there is a further composition of $\langle C_1, N' \rangle$ with another structured meaning, we cannot use the same condition for the contiguous case because the boundaries of $\langle C_1, N' \rangle$ is both N while the boundaries of some $\langle C_2, N_2 \rangle$ with contiguous components C_2 and N_2 are C_2 and N_2 (in either order). In order to close the operation of composition on structured meanings, we can only consider a finite number of subcases. One way to do this is to set up four possible boundary types, $N - N$, $N - C$, $C - N$, and $C - C$, and define the condition for these four subcases. We omit the actual conditions as it is tedious (commonly observed cases have been implemented and described in Chapter 6).

Complexity of Structured Meaning Representation

Naturally, the complexity introduced by the use of structured meaning is a concern. Here, we investigate the complexity of structured meaning and that of composing structured meanings.

First, the structural variation of structured meanings is limited to a pair of semantic representation with two additional cases where either of them is null. But each component can be discontiguous, as has been seen above. For a string of n lexical categories, each lexical category may belong to either C or N of $\langle C, N \rangle$. Thus, for the span of this n lexical categories, in theory, there are at most 2^n distinct structured meaning.[11] But, in practice, structured meanings with an internal division more complex than $C - N - C - N$ or $N - C - N - C$ are extremely rare. This is because in many cases, assignment or projection of a contextual-link status results in either $\langle C, - \rangle$ or $\langle -, N \rangle$ reducing the internal structure. Following the discussion on page 73 (in the last paragraph of the Summary), the two main sources of structured meanings are predicate-argument structure involving a main verb and modification structure involving a clausal modifier.

If the most complicated internal structure for a structured meaning is in practice 4-way, as in $C - N - C - N$, the practical bound on the number of distinct structured meanings for a single category is no more than the number of structure meanings for a 4-category sequence, i.e., $2^4 = 16$. This applies at every step of derivation. As a consequence, the overall increase of complexity due to

[11]This does not include various kinds of ambiguities.

introduction of structured meaning is in practice at most 16 times that of the case without structured meanings.

Next, let us discuss the complexity of composition involving structured meanings. The operation is closed because in addition to the structured meaning itself, we only recognize the contextual-link status of the boundary categories. As stated earlier, there are four boundary status pairs, $C-C$, $C-N$, $N-C$, $N-N$, for a structured meaning $\langle C,N\rangle$. There are two special cases $\langle C,-\rangle$ and $\langle -,N\rangle$ with no partition of the contextual-link status. For a composition of "$\langle C_1,N_1\rangle + \langle C_2,N_2\rangle$" resulting in $\langle C',N'\rangle$, we consider these 6 patterns for each input and result. A simplistic bound on all the possible combinations of the 6 patterns is $6\times 6\times 6 = 216$. This is a large number, but many of these patterns are not necessary in practice. For the current purpose, it is sufficient to show that there is a bound.

In conclusion, processing structured meanings is in practice multiplicative rather than exponential. This property is very important for the practicality of the use of structure meaning.

Summary

This subsection shows that composition of structured meaning can be done precisely and in practice does not increase asymptotic complexity. The conditions for composing structured meaning have been discussed, and summarized as follows:

(146) Conditions for semantic composition of structured meanings:

a. The semantic composition (either through functional application or functional composition) of the two components must be consistent with the semantic representation of the entire phrase.

b. The contextual-link component of a structured meaning (after composition) must satisfy the requirements for a contextual link (through discourse status, domain-specific knowledge, and/or linguistic marking).

Although the present application of structured meaning is for contextual-link status, the technique discussed in this section is applicable to structured meaning for contrast and other purposes.

4.3.2 Identification of Information Structure

Assuming that intermediate steps of compositions of structured meanings go well, identification of information structure is almost trivial. For the final structured meaning $\langle C, N \rangle$, we identify $Theme = C$ and $Rheme = N$. For example, again consider "Felix praised **Donald**." in response to "Who did Felix praise?" The last semantic composition is as follows:

(147)

$$\frac{\overset{\text{Felix praised}}{\left\langle -, \lambda X.praise'(X)(felix') \right\rangle} \quad \overset{\textbf{Donald}}{\left\langle -, donald' \right\rangle}}{\left\langle \underset{\substack{\text{contextual link} \\ \downarrow \\ Theme}}{\lambda X.praise'(X)(felix')}, \underset{\substack{\text{non-link} \\ \downarrow \\ Rheme}}{donald'} \right\rangle}$$

There may be some discontiguity within the theme and/or the rheme. But the information structure can be identified exactly the same way as before. The present approach is an improvement over that in Komagata [1998a]. In that paper, I characterized information structure as the last step of semantic composition. But this approach without structured meaning cannot cover discontiguous information structure in a general way as the present formulation does.

If multiple structured meanings are available at the end, the current theory accepts all the available information structures. For our implementation (Chapter 6), though, we have a few heuristics to choose more likely information structures.

4.3.3 Analysis of Gapping

We have seen that the problem of discontiguous information structure for the binomial-partition analyses can be solved by adopting the structured-meaning approach. We have also noted that the structured meaning approach is applicable to other areas including the analyses of contrast, propositional attitude, and thematic role. In this subsection, we apply structured meaning to yet another phenomenon of 'gapping'. In particular, we recast Steedman's [1990] 'decomposition' analysis in terms of structured meaning.[12]

Gapping in English has a form shown below [Steedman, 1990, (85), p. 242].

(148) Harry will buy bread, and Barry, potatoes.

[12] The analysis presented here is not compatible with Steedman's [1999, Chapter 7] more recent analysis.

It has received much attention for its peculiar construction. While earlier analyses were purely syntactic, Kuno [1976] pointed out pragmatic factors involved in the construction (for an extensive review, see Steedman [1999, Chapter 7]). Since this is a phenomenon involving discontinuity and potentially information structure, it would be a good demonstration if the structured-meaning approach can be applied to it.

Steedman's [1990, (85), p. 242] analysis is shown below. Note that there is a information-structure condition for decomposition that the gap must be 'known' [p. 250].

(149)

$$
\begin{array}{llll}
\text{Harry will buy bread,} & \text{and} & \text{Barry,} & \text{potatoes} \\
 & & S/(S\backslash NP) & (S\backslash NP)\backslash(S\backslash NP/NP) \\
\hline
S & & \multicolumn{2}{c}{S\backslash(S\backslash NP/NP)} \\
\hline
S\backslash NP/NP \quad S\backslash(S\backslash NP/NP) & \text{<decomp} & & \\
\hline
\multicolumn{4}{c}{S\backslash(S\backslash NP/NP) \quad \text{<\&>}}
\end{array}
$$

In the following, we apply our structured-meaning approach to the above case. The derivation of the category S results in a structured meaning such that the verb and the arguments split as $\left\langle \underbrace{Verb}_{Contextual\text{-}link}, \underbrace{Arguments}_{Non\text{-}link} \right\rangle$, and the non-contextual-link component is coordinated with the right conjunct. Let us assume that the split with respect to the contextual-link status is a source of gapping corresponding to Kuno's [1976] intuition and Steedman's [1990] condition on the decomposition. Then, the present approach can provide the following analysis for the left conjunct.

(150)

$$
\begin{array}{ccc}
\text{Harry} & \text{will buy} & \text{bread,} \\
NP & S\backslash NP/NP & NP \\
\left\langle -, harry' \right\rangle & \left\langle \lambda X.\lambda Y.buy'(X)(Y), - \right\rangle & \left\langle -, bread' \right\rangle \\
\hline
\multicolumn{3}{c}{S} \\
\multicolumn{3}{c}{\left\langle \lambda X.\lambda Y.buy'(X)(Y), \lambda P.P(bread')(harry') \right\rangle}
\end{array}
$$

At this point, let us hypothesize that the semantic unit $\lambda P.P(bread')(harry')$ is available as a part of a 'virtual category'. It is not a real category because *Harry* and *bread* are discontiguous. If we only deal with semantic information, this might be enough (as we have been doing up to this point in this section). But, in order to proceed with the coordination with the right conjunct, we need to analyze the syntactic type of the virtual category as well. Assuming that both NP's have a syntactic type $\mathsf{T}\big\langle^{/}_{\backslash}\left(\mathsf{T}\big\rangle^{\backslash}_{/}NP\right)$,[13] this virtual category might correspond to several distinct syntactic types as shown below.

(151)		Harry	bread		*Harry, bread* [virtual category]	
	$a.$	$\mathsf{T}/(\mathsf{T}\backslash NP_1)$	$\mathsf{T}/(\mathsf{T}\backslash NP_2)$	$\Longrightarrow$	$\mathsf{T}/(\mathsf{T}\backslash NP_1\backslash NP_2)$	$(>B)$
	$b_1.$	$\mathsf{T}/(\mathsf{T}\backslash NP_1)$	$\mathsf{T}\backslash(\mathsf{T}/NP_2)$	$\Longrightarrow$	$\mathsf{T}\backslash(\mathsf{T}\backslash NP_1/NP_2)$	$(>B\times)$
	$b_2.$	$\mathsf{T}/(\mathsf{T}\backslash NP_1)$	$\mathsf{T}\backslash(\mathsf{T}/NP_2)$	$\Longrightarrow$	$\mathsf{T}/(\mathsf{T}\backslash NP_2\backslash NP_1)$	$(<B\times)$
	$c.$	$\mathsf{T}\backslash(\mathsf{T}/NP_1)$	$\mathsf{T}/(\mathsf{T}\backslash NP_2)$	$\Longrightarrow$	fail	
	$d.$	$\mathsf{T}\backslash(\mathsf{T}/NP_1)$	$\mathsf{T}\backslash(\mathsf{T}/NP_2)$	$\Longrightarrow$	$\mathsf{T}\backslash(\mathsf{T}/NP_2/NP_1)$	$(<B)$

In the above, we have used forward crossing composition '$>B\times$', which is not generally assumed for English [Steedman, 1996, p. 53]. We will come back to this point shortly. First, the result (b_2) and (d) are excluded because of the semantic condition for composing structured meanings (146*b*). That is, the argument order of subject and object are incorrect, and thus cannot result in the correct semantic representation. Now, we extend this condition to syntactic type as well. This requires that the syntactic types of the components must be composed into the resulting syntactic type. Both the possibilities of having the virtual category to the left and right of the verb category are considered below.

(152)						
	$a.$	$\mathsf{T}/(\mathsf{T}\backslash NP_1\backslash NP_2)$	$S\backslash NP/NP$	$\Longrightarrow$	fail	
		$S\backslash NP/NP$	$\mathsf{T}/(\mathsf{T}\backslash NP_1\backslash NP_2)$	$\Longrightarrow$	fail	
	$b_1.$	$\mathsf{T}\backslash(\mathsf{T}\backslash NP_1/NP_2)$	$S\backslash NP/NP$	$\Longrightarrow$	fail	
		$S\backslash NP/NP$	$\mathsf{T}\backslash(\mathsf{T}\backslash NP_1/NP_2)$	$\Longrightarrow$	S	$(<)$

Thus, the only possibility is that the virtual category is to the right of the verb and the variable T is instantiated as S with the correct argument order. This 'virtual' derivation is shown below.

(153) will buy — *Harry, bread* [virtual category]

$$\dfrac{\begin{array}{c} S\backslash NP/NP \\ \left\langle \lambda X.\lambda Y.buy'(X)(Y), - \right\rangle \end{array} \quad \begin{array}{c} S\backslash(S\backslash NP/NP) \\ \left\langle -, \lambda P.P(bread')(harry') \right\rangle \end{array}}{\begin{array}{c} S \\ \left\langle \lambda X.\lambda Y.buy'(X)(Y), \lambda P.P(bread')(harry') \right\rangle \end{array}}$$

[13] We use the variable notation for conciseness and generality.

Although this scheme appears like decomposition, it is not exactly the type of decomposition proposed in Steedman [1990]. This is because the identification of the virtual category is done constructively at the same time as the category *S* is derived.

The syntactic type, semantic type, and relative position of the virtual category license the following coordination with the right conjunct.

(154) *Harry, bread* [virtual category] and Barry, potatoes

$$\mathrm{T}\substack{/\\ \backslash}\left(\mathrm{T}\substack{\backslash\\ /}NP_1\right) \qquad \mathrm{T}\substack{/\\ \backslash}\left(\mathrm{T}\substack{\backslash\\ /}NP_2\right)$$

$$S\backslash(S\backslash NP/NP) \qquad \begin{cases} \mathrm{T}/(\mathrm{T}\backslash NP_1\backslash NP_2) & \text{fail} \\ \mathrm{T}\backslash(\mathrm{T}\backslash NP_1/NP_2) & \text{success} \\ \mathrm{T}\backslash(\mathrm{T}/NP_2/NP_1) & \text{fail} \\ \mathrm{T}/(\mathrm{T}/NP_2\backslash NP_1) & \text{fail} \end{cases}$$

$$\left\langle -, \lambda P.P\,(br)\,(h)\right\rangle \qquad \left\langle -, \lambda P.P\,(p)\,(ba)\right\rangle \quad <\&>$$

$$S\backslash(S\backslash NP/NP)$$

$$\left\langle -, \lambda P.and\left(P\,(br)\,(h)\right)\left(P\,(p)\,(ba)\right)\right\rangle$$

Then, this can compose with the verb to derive the desired *S* category with the intended semantic representation.

There is one more point we should address. Forward crossing composition ($>B\times$), which is crucial to the derivation of the virtual category and the right conjunct, is not normally allowed in the surface grammar of English [Steedman, 1996, p. 53]. In a sense, this would incorrectly predict 'scrambling' of arguments at the left of a verb. The current position to compromise the demand for forward crossing composition in the above analysis and this constraint is the following. Forward crossing composition is available even in English (for both surface and virtual cases). But the result of this process is available only for compositions and coordination involving a virtual category.

The above analysis of gapping in terms of structured meaning demonstrates usefulness of structured meaning beyond the current project. It also suggests that the structured meaning may involve syntactic types, as well as semantic representation.

4.4 Summary

This chapter demonstrates CCG's advantages in (i) recognizing non-traditional constituents in accordance to units of information structure, (ii) capturing the properties for contextual links, and (iii) integrating structured meaning for analysis of discontiguous information structure. The formalization congruently integrates syntax, semantics, and discourse status, and provides a basis for bridging the theory (Chapter 3) and the implementation (Chapter 6) in a straightforward manner.