# A Computational Approach to English Questions*

Yong-hun Lee
(Chungnam National University & Hannam University)

**Lee, Yong-hun.** 2004. **A Computational Approach to English Questions.** *Korean Journal of English Language and Linguistics* 4-2, 175-194. This paper provides computational algorithms for English Questions, by which we can effectively handle and implement Yes-No Questions and Wh-Questions. Those algorithms will be developed in Categorial Grammar. In this paper, we will modify and revise Steedman's Combinatory Categorial Grammar (CCG) so that we can computationally implement Questions effectively, which will be called a CCG-like system. In this system, semantic interpretations of Questions will be calculated compositionally based on the functor-arguments relations of the constituents. In sum, this paper provides analyses of Questions in Categorial Grammar, by which we can effectively implement Questions in English.

**Key Words**: Yes-No question, Wh-question, computational, CCG-like system, compositional

## 1. Introduction

Question is one of the important sentence forms, by which we ask something.[1] There are several different types of Questions: Yes-No

---

*Many parts of this paper are similar to those in Lee (2004). Lee (2004) handles Questions in Korean whereas this paper concerns those in English. Because these two papers are based on the same framework, i.e., Categorial Grammar, the basic explanations for the framework are identical. But, Question-handling algorithms in these two papers are a little different. For details of Korean Questions, see Lee (2004).

[1]When we say *Questions*, rather than *Interrogatives*, it usually implies that we especially focus on semantic interpretations in addition to syntactic behaviors.

Questions, Wh-Questions, Tag Questions, and so on. We may have different sorts of answers depending on what kinds of Questions we have.

English also has different types of Questions. For example, from a Declarative sentence in (1a), we may derive several different kinds of Questions, as (1b) - (1d) illustrate.

(1) a. John loves Mary.
    b. Does John love Mary?
    c. Who loves Mary?
    d. Who(m) does John love?

(1b) is a Yes-No Question, and (1c) and (1d) are Wh-Questions.

The goal of this paper is to provide *computational* algorithms for Questions in English. Here, *computational* has dual meaning à la Lee (2003). One is *operations on the representations*, and the other is *computational implementations*. The system developed in this paper presupposes its computational implementations, though specific implementational algorithms will not be provided in this paper. But, the system introduced in this paper can easily be changed into computer languages, such as JAVA or C++.

This paper is organized as follows. Section 2 summarizes previous approaches to Questions. Section 3 demonstrates some characteristics of English Questions. Section 4 introduces basic ideas in Categorial Grammar, Steedman's CCG, and a CCG-like system. In Section 5, we analyze English Questions in the CCG-like system. Section 6 discusses the advantages and some remaining problems in analyzing Questions in our system. Section 7 summarizes this paper.

## 2. Previous Approaches to Questions

Before we start to examine Questions in English, it will be helpful to have a short review for previous approaches in

Questions. This section provides a short review of studies in Questions, dividing them into syntax and semantics.

Syntactic approaches to Questions, especially in Chomskyan traditions, have focused on how Questions can be made from their Declarative counterparts. The major tool that they have used in their analyses is *movement*. (2) shows us how the analyses on Yes-No Questions and Wh-Questions have developed historically in Chomskyan traditions.

(2) Syntactic Approach to Questions (Chomskyan Traditions)

| | Yes-No Question | Wh-Question |
|---|---|---|
| TG (Chomsky 1957/1977) | Subject-Aux Inversion | Wh-Movement |
| GB (Chomsky 1981/1986) | I-to-C Movement | Wh-Movement |
| MP (Chomsky 1995) | Move (Head Mvt.) | Move (Wh-Mvt.) |

As you can see in this table, *movement* plays an important role to the analyses of Questions in Chomskyan traditions.

There are also other types of syntactic approaches where *movement* is not used. They include the analyses in GPSG (Generalized Phrase Structure Grammar; Gazdar et al. 1985) and HPSG (Head-driven Phrase Structure Grammar; Pollard and Sag 1994, Sag and Wasow 1999). In those approaches, Questions are analyzed with feature instantiation and feature percolation.

Semantic approaches to Questions can be divided into three groups. The first is Categorial Approaches, where semantic types are used for catching the meaning of Questions. The second is Embedding Approaches, where the meaning of Questions is captured by embedding the Questions into Declarative sentences, along with some verbs such as *know, wonder*, etc. The last one is Propositional Approaches, where the meaning of Questions is represented by propositions.[2)]

There are three major claims in Propositional Approaches to Questions. Hamblim (1973) said that the meaning of a Question

---

[2]For details of these approaches, see Hong (1999).

is the set of possible answers. Karttunen (1977), however, claims that the meaning of a Question is the set of true answers. Groenendijk & Stokhof (1982, 1984) have a little different position, and they said that the meaning of a Question is just propositions, not a set of propositions.

## 3. Questions in English

From a single Declarative sentence, we may derive several different types of Questions. In this section, we overview the types of English Questions, focused on Yes-No Questions and Wh-Questions.

Yes-No Questions in English are made with an auxiliary verb *do* in the sentence-initial position, where the verb *do* carries the tense and agreement information.[3] For example, from a declarative sentence (3a), we can derive a simple Yes-No Question (3b).

(3) a. John loves Mary.
b. Does John love Mary?

As you can see in (3b), *does* in the sentence-initial position has the tense and agreement information.

We may have other types of Yes-No questions. *John* or *Mary* in (3a) can be replaced with an indefinite pronoun, such as *someone* or *anyone*, and we can derive Yes-No Questions from them. For example, we may have (4b) from (4a), and (5b) from (5b).

(4) a. Someone loves Mary.
b. Does anyone love Mary?

---

[3] As one reviewer pointed out, Yes-No Questions in English can be made with the verb *be* or other auxiliary verbs. In those cases, however, the auxiliary verbs can be handled similarly. Therefore, this paper will not include those cases.

(5) a. John loves someone.
b. Does John love anyone?

(4b) and (5b) are also Yes-No Questions that we can have in English.

Wh-Questions in English are made with wh-words. For example, from (6a), we may derive two Wh-Questions (6b) and (6c), where wh-word occupies the subject position of in (6b) but it occupies the object position in (6c).

(6) a. John loves Mary.
b. Who loves Mary?
c. Who(m) does John love?

One of characteristics of English Wh-Questions is that wh-words prefer to be fronted into the sentence-initial position. For example, we may have either (7b) or (7c) from (7a). The wh-word *whom* is in the object position in (7b), whereas it is fronted into the sentence-initial position in (7c).

(7) a. John loves Mary.
b. John loves whom?
c. Who(m) does John love?

In this paper, we will focus on the analysis of (7c) rather than (7b), and we will not handle the sentence (7b).

## 4. Categorial Grammar and A CCG-like System

### 4.1. Basic Ideas in Categorial Grammar

Categorial Grammar was introduced by Ajdukiewicz (1935) and modified and advanced by Bar-Hillel, Curry, and Lambek. In this framework, we have two basic categories NP and S, and other categories are made from the combinations of these two categories. In this grammar, every lexical item has its own *category* information in the Lexicon, and all the syntactic phenomena are described and analyzed by the functor-argument relations of these categories.

Steedman (1996, 2000) extended previous studies in Categorial Grammar and introduced Combinatory Categorial Grammar (CCG). The most important characteristic of his system is that predicate-arguments relations are projected by the combinatory rules of syntax, and other operations are based on these relations (Steedman, 2000:38). The most primary combinatory operation in Categorial Grammar is *functional application,* which is shown below in (8).

(8) Functional Application (Steedman, 1996:13, 2000:37)

a. $X / Y : f \quad Y : a \quad \rightarrow \quad X : f\,a$ (>)

b. $Y : a \quad X \backslash Y : f \quad \rightarrow \quad X : f\,a$ (<)

Here, (8a) is called a *forward functional application,* and (8b) is a *backward functional application.* $f$ is the translation of the functor category, and $a$ stands for that of the argument category. The schema in (8) illustrates how syntactic operations, i.e., category combinatorics, can be related with its semantic interpretations.

The other operations that we will use in this paper are *functional composition* and *type raising,* which is shown in (9) and (10) respectively. Here, the semantic interpretations are added to category combinatorics as in (8).

(9) Functional Composition (Steedman, 1996:43, 2000:55)[4]

a. $X / Y : f \quad Y / Z : g \quad \rightarrow_{\mathbf{B}} \quad X / Z : \lambda x.f(gx)$ (> **B**)

b. $X / Y : g \quad Y \backslash Z : f \quad \rightarrow_{\mathbf{B}} \quad X \backslash Z : \lambda x.g(fx)$ (> **B**)

[4]Steedman (1996, 2000) didn't contain all the semantic interpretations to all of the rules in *functional composition.* Semantic interpretations in (9) here are added with help of Kang (2001:84-85).

c. $Y \backslash Z : f \quad X \backslash Y : g \rightarrow_B X \backslash Z : \lambda x.g(fx)$ ($<$ **B**)
d. $Y / Z : g \quad X \backslash Y : f \rightarrow_B X / Z : \lambda x.f(gx)$ ($<$ **B**)

(10) Type Raising (Steedman, 1996:37, 2000:44)
a. $X : a \rightarrow_T T / (T \backslash X) : \lambda f.fa$ ($>$ **T**)
b. $X : a \rightarrow_T T \backslash (T / X) : \lambda f.fa$ ($<$ **T**)

(9) and (10) also demonstrates the relations between syntactic operations and semantic interpretations. (9a) and (9b) are called *forward functional compositions,* and (9c) and (9d) are *backward functional compositions.* Likewise, (10a) is called a *forward type raising,* whereas (10b) is a *backward type raising.*

**4.2. A CCG-like System (Lee 2003)**

The system that this paper develops is a CCG-like system, which has been introduced in Lee (2003). It is basically the incorporation of storage mechanisms into Steedman's CCG. This system is similar to Steedman's system in that surface combinatorics triggers other operations, especially Question-handling algorithms in this paper. It is different from Steedman's in that it uses attribute-value ordered pairs (**avop**) in (11) to describe syntactic dependencies of constituents. These five attributes are explained in (12).

(11) Structure of Attribute-Value Ordered Pair (**avop**)
<PHON, CAT, TRANS, NPS, SLASH>

(12) Five Attributes
a. PHON
( i ) phonological/morphological form
( ii ) concatenates a word to a stream of words
b. CAT
( i ) has categorial information

(ii) such as S, NP, S\NP, and so on

c. TRANS

(i) semantic interpretation

(ii) based on Montagovian semantics

d. NPS (NP Index Store)

(i) something like a Cooper-storage

(ii) has indices of NPs

e. SLASH

(i) similar to that of HPSG

(ii) necessary to deal with crossover phenomena

The operations on the CAT values trigger operations on all the other **avop** values. The semantic interpretation of Questions, i.e., the TRANS value, is also calculated by the result of *functional application*, *functional composition*, or *type raising*.

In this paper, the index types of NPs are classified as in (13), and this information represents the type of NP.

(13) Index Types of NPs

a. $n_{+name}$ : proper nouns

b. $n_{+pron}$ : pronominals

These index types are stored in NPS or SLASH, and play important role to identify whether the given sentence is a Declarative, a Yes-No Question, or a Wh-Question.

## 5. Analyzing Questions in the CCG-like System

### 5.1. Declaratives and Questions

Before we start to analyze Yes-No Questions and Wh-Questions in the CCG-like system, we have to think about the followings: (i) how we can capture the semantic interpretation of Declaratives and Questions, and (ii) how we can represent them semantically.

First, let's see how a Declarative (3a) is analyzed in the CCG-like system. It will be useful to compare the analyses of Declaratives with those of Questions. (14) is the analysis for (3a). Here, the superscripted bracketed numbers in (14a) refers to phonological/morphological forms of each lexical item.5)

(14) a. [1]John [2]loves [3]Mary.

b. <[1]+[2]+[3],S,love′(*j*,*m*),NPS:{$1_{+name}$,$2_{+name}$},SLASH: ∅ >

<[1],$NP_1$,*j*,NPS:{$1_{+name}$},SLASH: ∅ ><[2]+[3],S\\$NP_1$,love′(*m*),NPS:{1,$2_{+name}$},SLASH: ∅ >

<[2],(S\\$NP_1$)/NP,love′,NPS:{1},SLASH: ∅ ><[3],NP,*m*,NPS:{$2_{+name}$},SLASH: ∅ >

The analysis proceeds as follows. First, *loves* is combined with *Mary*, whose category are ($S \backslash NP_1$)/NP and NP respectively. Because ($S \backslash NP_1$)/NP is the functor and NP is the argument, a *forward functional application* in (8a) is applied.

(8) Functional Application (Steedman, 1996:13, 2000:37)

| | | | | | |
|---|---|---|---|---|---|
| a. | $X / Y : f$ | $Y : a$ | $\rightarrow$ | $X : f\ a$ | (>) |
| b. | $Y : a$ | $X \backslash Y : f$ | $\rightarrow$ | $X : f\ a$ | (<) |

The result of *functional application* becomes $S \backslash NP_1$ and its interpretation is calculated as *love′*(*m*). The PHON value is calculated just by concatenating two constituent values. NPS and SLASH values for the category $S \backslash NP_1$ are calculated by the union of each value of the constituents. Then, *John* combines with *loves Mary*, whose categories are $NP_1$ and $S \backslash NP_1$ respectively. Now, since $S \backslash NP_1$ is the functor and $NP_1$ is the argument, a *backward functional application* in (8b) is applied. The category value becomes S, and its semantic interpretation is

5Here 1 in ($S \backslash NP_1$)/NP means the index of subject NP must be 1. It is a tool for subject-predicate agreement.

calculated as *love'(m)(j)*, where *love'(m)* comes from S\$NP_1$ and *j* from $NP_1$. *love'(m)(j)* becomes *love'(j,m)* by Relational Notations (Dowty et al., 1981:164). By these processes, we can analyze a Declarative sentence (14a), which is identical to (3a), successfully in the CCG-like system.

Now, let's move to semantics of Questions. Karttunen derived the meaning of Questions from that of Declaratives using the Proto-Question Rule in (15).

(15) Proto-Question Rule (Karttunen, 1977:13)
If $\Phi \in P_t$, then $?\Phi \in P_Q$.
If $\Phi$ translates to $\Phi'$, then $?\Phi$ translates to $\lambda p[{}^{\vee}p \wedge {}^{\wedge}p = \Phi']$

As you can see, $\Phi$ is a t-type expression (i.e., a Declarative), while $?\Phi$ is a Q-type expression (i.e., a Question). If $\Phi'$ is the translation of $\Phi$, then $?\Phi$ is translated into $\lambda p[{}^{\vee}p \wedge {}^{\wedge}p = \Phi']$. Along with this rule, he analyzed the sentence (16) as in (17).

(16) Who dates Mary?

(17) a. ?$he_i$ dates Mary $\lambda p[{}^{\vee}p \wedge {}^{\wedge}p = \text{date}'(x_i,m)]$
b. who $\lambda P \exists x P\{x\}$
c. Who dates Mary? $\lambda p \exists x[{}^{\vee}p \wedge {}^{\wedge}p = \text{date}'(x,m)]$

From a Declarative sentence *he dates Mary*, we can derive a Proto-Question form in (17a). Here $he_i$ is translated into $x_i$. A wh-word *who* is translated as in (17b). Then, (17b) combines with (17a). By applying $\lambda$-conversion, we can derive the semantic interpretation in (17c).

The semantic interpretation in (17c), however, has ${}^{\vee}$-operator, ${}^{\wedge}$-operator, and brace notation. Those are operators in *intensional logic*, whose computational implementations are very difficult. Thus, in this paper, we will develop a little different mechanisms, which are easy to computational implementations.

First, we need to think about how Yes-No Questions and Wh-Questions are different in their semantic interpretations. Traditionally, especially in Propositional Approaches, semantic interpretation of Yes-No Questions is known to be *a set of two propositions, one of which is true and the other is false,* whereas that of Wh-Questions is said to be *a set of proposition or a proposition.* In this paper, semantics of Questions will be represented by *propositions* rather than *a set of propositions,* as in Groenendijk & Stokhof (1982, 1984). The semantic interpretations of Yes-No Questions and Wh-Questions in the CCG-like system are represented as in (18).

(18) Semantic Interpretations of Questions in the CCG-like System
 a. Yes-No Questions : Two propositions, one of which is true and the other is false.
 b. Wh-Questions : A set of entities that satisfied the propositions.

The true proposition and the false one will be connected by $\vee$ (logical *or*), rather than we put them into a set. But, the semantics of Wh-Questions will be represented by $\lambda$-expressions.

Now, let's think about the semantic interpretation of the *who.* This paper gets the basic idea from that of *some,* whose interpretation is shown in (19a). (Dowty et al., 1981:108)

(19) Semantic Interpretation of *some* and *someone*

| | | |
|---|---|---|
| a. some | : | $\lambda Q[\lambda P \exists x[Q(x) \wedge P(x)]]$ |
| b. someone | : | $\lambda P \exists x[\text{person}'(x) \wedge P(x)]$ |
| c. anyone | : | $\lambda P \exists x[\text{person}'(x) \wedge P(x)]$ |
| d. who | : | $\lambda P \lambda \text{x}[\text{person}'(x) \wedge P(x)]$ |

From (19a), we can derive the semantic interpretation of *someone* as in (19b), replacing Q with *person'*. From (19b), we can derive the semantic interpretation of *anyone* as in (19c). As you can find, the semantic interpretations of *someone* and *anyone* are identical. Their difference comes from syntactic distributions. That is, *someone* is used in positive Declarative sentences and *anyone* is used in negative Declaratives and Questions.[6] Once again, from (19c), we can obtain the semantic interpretation of *who* as in (19d). As you can observe, we can obtain the semantic interpretation of the wh-pronoun *who* from the indefinite pronoun *anyone*, just by replacing ∃ with λ. You will find how these two pronouns are different from the following sections.

### 5.2. Yes-No Questions

Now, let's start our analyses from Yes-No Questions. As we saw in Section 3, we can derive a Yes-No Question in (3b) from a Declarative (3a). (20) is the analysis for (3b).

(20) a. [1]Does [2]John [3]love [4]Mary?

b.

$$\langle[1]+[2]+[3]+[4], S_Q, [love'(j,m) \vee \neg love'(j,m)], NPS{:}\{1_{+name}, 2_{+name}\}, SLASH{:}\varnothing\rangle$$

$$\langle[1], S_Q/S_R, \lambda f[f \vee \neg f], NPS{:}\varnothing, SLASH{:}\varnothing\rangle \langle[2]+[3]+[4], S_R, love'(j,m), NPS{:}\{1_{+name}, 2_{+name}\}, SLASH{:}\varnothing\rangle$$

$$\langle[2], NP_1, j, NPS{:}\{1_{+name}\}, SLASH{:}\varnothing\rangle \langle[3]+[4], S_R\backslash NP_1, love'(m), NPS{:}\{1, 2_{+name}\}, SLASH{:}\varnothing\rangle$$

$$\langle[3], (S_R\backslash NP_1)/NP, love', NPS{:}\{1\}, SLASH{:}\varnothing\rangle \langle[4], NP, m, NPS{:}\{2_{+name}\}, SLASH{:}\varnothing\rangle$$

The processes up to *John love Mary* is identical with those in (14b). The only difference is that the category value for *John loves Mary* in (14b) is S, whereas that for *John love Mary* in (20b) is $S_R$. Note that *John loves Mary* in (14b) has tense and

---

[6]Someone may say that those different syntactic distributions come from the meaning of *some* vs. *any*, and ask how we can handle those meaning differences in the semantic interpretations. In this paper, I will leave this problem open to further research.

agreement information, whereas *John love Mary* in (20b) does not. S is the category for the sentences whose tense and agreement are fully implemented. $S_R$ in (20b) is the category for the sentences where tense and agreement are not expressed, i.e., S root. After we proceed up to *John love Mary*, then we combine *Does* with *John love Mary*. Note that the semantic interpretation of *Does* is defined as $\lambda f[f \vee \neg f]$, which is a <t,t>-type expression. This *Does* makes a declarative sentence into a Yes-No Question. After *Does* combines with *John love Mary* by a *forward functional application* in (8a), the semantic interpretation of the whole sentence *Does John love Mary?* is calculated as $[love'(j,m) \vee \neg love'(j,m)]$. Note that two propositions are connected by logical *or*, one of which is true and the other is false, whether $love'(j,m)$ is true or $\neg love'(j,m)$ is true.[7]

Now, let's move on to (4) and (5). These sentences have indefinite pronoun *anyone*. (4b) and (5b) are the Yes-No Questions for (4a) and (5a) respectively. We will make use of the semantic interpretation in (19c) for *anyone*. The analyses for (4a) and (5a) will be similar to (20), though they will not be shown in this paper. The only difference is that *someone* in (19b) is used instead of *John* or *Mary* in the sentence. Basic mechanisms are identical.

Now, let's see how (4b) and (5b) can be analyzed in our system. First, (21) is the analysis for (4b).

(21) a. [1]Does [2]anyone [3]love [4]Mary?

b. <[1]+[2]+[3]+[4], $S_Q$, $[\exists x_1[person'(x_1) \wedge love'(x_1,m)] \vee \neg \exists x_1[person'(x_1) \wedge love'(x_1,m)]$, NPS:$\{1_{+pron}, 2_{+name}\}$, SLASH: ∅>

<[1], $S_Q/S_R$, $\lambda f[f \vee \neg f]$, NPS: ∅, SLASH: ∅><[2]+[3]+[4], $S_R$, $\exists x_1[person'(x_1) \wedge love'(x_1,m)]$, NPS:$\{1_{+pron}, 2_{+name}\}$, SLASH: ∅>

<[2], $S_R/(S_R\backslash NP_1)$, $\lambda P \exists x_1[person'(x_1) \wedge P(x_1)]$, NPS:$\{1_{+pron}\}$, SLASH: ∅><[3]+[4], $S_R\backslash NP_1$, $love'(m)$, NPS:$\{1, 2_{+name}\}$, SLASH: ∅>

<[3], $(S_R\backslash NP_1)/NP$, love', NPS:{1}, SLASH: ∅><[4], NP, *m*, NPS:$\{2_{+name}\}$, SLASH: ∅>

---

[7] As one reviewer pointed out, the semantic interpretation of (20a), i.e., $[love'(j,m) \vee \neg love'(j,m)]$, looks like a tautology. This is a problem open to further research.

Note that the category value of *anyone* in the subject position is $S_R/(S_R\backslash NP_1)$ and that its semantic interpretation is defined as $\lambda P \exists x_3[\mathrm{person}'(x_3) \wedge P(x_3)]$. After *love* combines with *Mary, anyone* combines with *love Mary,* here by a *forward functional application* in (8a). Then, *Does* combines with *anyone love Mary* with a *forward functional application* in (8a). Note that the final semantic interpretation has an existential quantifier $\exists$, which comes from *anyone.* As in (20b), the semantic interpretation of (4b) is represented by two propositions are connected by logical *or,* one of which is true and the other is false.

Now, let's move to (5b). (22) is the analysis for (5b).

(22) a. [1]Does [2]John [3]love [4]anyone?

b. $<[1]+[2]+[3]+[4],S_Q,[\exists x_2[\mathrm{person}'(x_2) \wedge \mathrm{love}'(j,x_2)] \vee \neg \exists x_2[\mathrm{person}'(x_2) \wedge \mathrm{love}'(j,x_2)]],\mathrm{NPS}:\{1_{+name},2_{+pron}\},\mathrm{SLASH}:\varnothing>$

$<[1],S_Q/S_R,\lambda f[f \vee \neg f],\mathrm{NPS}:\varnothing,\mathrm{SLASH}:\varnothing><[2]+[3]+[4],S_R,\exists x_2[\mathrm{person}'(x_2) \wedge \mathrm{love}'(j,x_2)],\mathrm{NPS}:\{1_{+name},2_{+pron}\},\mathrm{SLASH}:\varnothing>$

$<[2],NP_1,j,\mathrm{NPS}:\{1_{+name}\},\mathrm{SLASH}:\varnothing><[3]+[4],S_R\backslash NP_1,\lambda x_1 \exists x_2[\mathrm{person}'(x_2) \wedge \mathrm{love}'(x_1,x_2)],\mathrm{NPS}:\{1,2_{+pron}\},\mathrm{SLASH}:\varnothing>$

$<[3],(S_R\backslash NP_1)/NP,\mathrm{love}',\mathrm{NPS}:\{1\},\mathrm{SLASH}:\varnothing><[4],NP,\lambda P \lambda x_1 \exists x_2[\mathrm{person}'(x_2) \wedge P(x_1,x_2)],\mathrm{NPS}:\{2_{+pron}\},\mathrm{SLASH}:\varnothing>$

Note that the category value of *anyone* in the object position is NP and that its semantic interpretation is $\lambda P \lambda x_1 \exists x_2[\mathrm{person}'(x_2) \wedge P(x_1,x_2)]$. Here, $x_1$ is the place for the subject NP. Every thing else needs no further explanations.

### 5.3. Wh-Questions

Now, let's move to Wh-Questions in (6). From (6a), we can derive two Wh-Questions (6b) and (6c). The subject is questioned in (6b) whereas the object is questioned in (6c). Here, we will make use of the semantic interpretation in (19d) for *who.*

(23) is the analysis for the sentence (6b).

(23) a. [1]Who [2]loves [3]Mary?

b. <[1]+[2]+[3],$S_Q$,$\lambda x_1$[person′($x_1$) $\wedge$ love′($x_1$,$m$)],NPS:{$1_{+pron}$,$2_{+name}$},SLASH:{$1_{+pron}$}>

<[1],$S_Q/(S_R\backslash NP_1)$,$\lambda P \lambda x_1$[person′($x_1$) $\wedge$ P($x_1$)],NPS:{1},SLASH:{$1_{+pron}$}><[2]+[3],$S_R\backslash NP_1$,love′($m$),NPS:{1,$2_{+name}$},SLASH: ∅>

<[2],$(S_R\backslash NP_1)/NP$,love′,NPS:{1},SLASH: ∅><[3],NP,$m$,NPS:{$2_{+name}$},SLASH: ∅>

Note that the category value of *Who* is $S_Q/(S_R\backslash NP_1)$ and that its semantic interpretation is defined as $\lambda P\lambda x_1$[person′($x_1$) $\wedge$ P($x_1$)]. After *loves* combines with *Mary*, *who* combines with *loves Mary*, here by a *forward functional application* in (8a). Note that SLASH has a index for the wh-pronoun *who*. This difference distinguishes Wh-Questions from Yes-No Questions. In the analyses for Declaratives and Yes-No Questions, this value was empty, i.e., ∅. Also, note that the semantic interpretation of a Wh-Question (6b) is represented by a set of entities that satisfied the proposition [person′($x_1$) $\wedge$ *love*′($x_1$,$m$)].

Now, let's move to the analysis for (6c). (24) illustrates the analysis.

(24) a. [1]Who(m) [2]does [3]John [4]love?

b. <[1]+[2]+[3]+[4],$S_Q$,$\lambda x_1$[person′($x_1$) $\wedge$ love′($j$,$x_1$)],NPS:{$1_{+pron}$,$2_{+name}$},SLASH:{$1_{+pron}$}>

<[1]+[2]+[3]+[4],$S_Q$,$\lambda x_1$[person′($x_1$) $\wedge$ $\lambda x_3$[love′($j$,$x_3$)]($x_1$)],NPS:{$1_{+pron}$,$2_{+name}$},SLASH:{$1_{+pron}$,$3_{+pron}$}>

<[1],$S_Q/(S_Q\backslash NP_1)$,$\lambda P\lambda x_1$[person′($x_1$) $\wedge$ P($x_1$)],NPS: ∅,SLASH:{$1_{+pron}$}><[2]+[3]+[4],$S_Q\backslash NP$,$\lambda x_3$[love′($j$,$x_3$)],NPS:{$2_{+name}$},SLASH:{$3_{+pron}$}>

<[2],$(S_Q\backslash NP)/(S_R/NP)$,$\lambda$PP,NPS: ∅,SLASH: ∅><[3]+[4],$S_R/NP$,$\lambda x_3$[love′($j$,$x_3$)],NPS:{$2_{+name}$},SLASH:{$3_{+pron}$}>

<[3],$S_R/(S_R\backslash NP_2)$,$\lambda$PP($j$),NPS:{$2_{+name}$},SLASH: ∅><[4],$(S_R\backslash NP_2)/NP$,$\lambda x_3$[love′($x_3$)],NPS:{2},SLASH:{$3_{+pron}$}>

<[3],$NP_2$,$j$,NPS:{$2_{+name}$},SLASH: ∅>

As you can see, The analysis in (24) is very different from the others. Let's see how this analysis proceeds step by step. First, *John* goes though a *forward type raising* in (10a).

(10) Type Raising (Steedman, 1996:37, 2000:44)

a. $X : a \rightarrow_T T / ( T \backslash X ) : \lambda f.fa$ (> **T**)

b. $X : a \rightarrow_T T \backslash ( T / X ) : \lambda f.fa$ (< **T**)

Its category is changed from $NP_2$ to $S_R/(S_R \backslash NP_2)$, where $S_R$ corresponds to T and $NP_2$ to X. The semantic interpretation is also changed from *j* to $\lambda P.P(j)$. On the other hand, the SLASH value for *love* contains $\{3_{+pron}\}$, and semantic interpretation is defined as $\lambda x_3[love'(x_3)]$. It is the index for missing objective NP. This index will be used later. Then, *John* combines with *love* by a *forward functional composition* in (9a).

(9) Functional Composition (Steedman, 1996:43, 2000:55)

a. $X / Y : f \quad Y / Z : g \rightarrow_B X / Z : \lambda x.f(gx)$ (> **B**)

b. $X / Y : g \quad Y \backslash Z : f \rightarrow_B X \backslash Z : \lambda x.g(fx)$ (> **B**)

c. $Y \backslash Z : f \quad X \backslash Y : g \rightarrow_B X \backslash Z : \lambda x.g(fx)$ (< **B**)

d. $Y / Z : g \quad X \backslash Y : f \rightarrow_B X / Z : \lambda x.f(gx)$ (< **B**)

Here $S_R$ corresponds to X, $S_R/NP_2$ to Y, and NP to Z. The result category becomes $S_R/NP$, and its semantic interpretation becomes $\lambda x_3[love'(j,x_3)]$. Then, *does* combines with *John love*. Note that its category is $(S_Q \backslash NP)/(S_R/NP)$ and that its semantic interpretation is $\lambda P.P$.[8] That is, the role of *does* here is to change the direction of combination, without its semantic change. This tool is necessary because a wh-word *who(m)* is located in front of *does John love*. Then, finally *who(m)* combines with *does John love*, by a *forward functional application* in (8a). Here, note that the SLASH value after the combination contains two indexes $\{1_{+pron}, 3_{+pron}\}$. But, because the index for the missing NP, i.e., $3_{+pron}$, have to be identical with that of *Who(m)*, $3_{+pron}$ is deleted from the SLASH

[8]As one reviewer suggested, the different semantic interpretation of *do* in Yes-No Questions and Wh-Questions may be handled with different features that cause the semantic parts. I thank the reviewer for his suggestion.

after the combination. By $\lambda$-conversion, $x_3$ is also changed into $x_1$. Along with these processes, we can successfully analyze a Wh-Question in (6c).

## 6. Advantages and Remaining Problems

### 6.1. Advantages of the CCG-like System

So far, we have analyzed Yes-No Questions and Wh-Questions in the CCG-like system. There are several advantages using the CCG-like system in analyzing Questions.

First, as you observe in the above analyses, the semantic interpretations of Questions are calculated *computationally* from the syntactic operations, i.e., from category combinatorics. When two constituents are combined into one, the schemata for *functional application*, *functional composition*, or *type raising* plays a crucial role. These schemata relate syntactic operations to their semantic interpretations. Therefore, if we make use of this characteristic, Questions are also easy to be implemented on a computer.

Second, the semantic interpretations of the Questions can be calculated *compositionally*. That is, the semantic interpretations of Questions can be calculated from those of constituents, in the similar mechanisms by which we calculate the semantic interpretations of Declaratives in Montague Grammar.

### 6.2. Some Remaining Problems

Though we can analyze English Questions in the CCG-like system, there are also some remaining problems. This section enumerates some of them.

The first problem occurs when Questions are embedded in another sentences. (25) illustrates some examples.

(25) a. Bill knows whether John loves Mary.

b. Bill knows who loves Mary.
c. Bill knows whom John loves.

Many scholars, especially those in Propositional Approaches to Questions, started their studies from these kinds of sentences. But, this paper started from the direct Questions, not from the indirect Questions. Then, the next job will be how we analyze the sentences in (25) in the CCG-like system. There are also some questions: (i) whether the semantic interpretation of indirect Questions is different from that of direct Questions or not, and (ii) if different, how we can capture the differences in the semantic interpretations.

There is another problem that is related to the first one.[9] Usually, when indirect Questions are embedded into other sentences, as in (25), they have to be t-type expressions. Yes-No Questions in our analyses raise no problem. They are t-type expressions, because they refer to two propositions connected by logical *or*, one of which is true and the other is false. But, Wh-Questions raise a problem. Because they refer to a set of entities that satisfied the propositions, they are <e,t>-type expressions, rather than t-type expressions. Therefore, in order to analyse the sentences such as (25b) or (25c), we need another operations that covert <e,t>-type expressions into t-type expressions, which are similar to Groenendijk & Stokhof (1982, 1984).

There is one more problem. Some scholars say that Questions, especially Wh-Questions, are related with focus reading. If so, we have to adopt ideas of Krifka (1991) or Rooth (1985, 1992) in the analyses of Wh-Questions. Then, the analyses for Wh-Questions will be changed from those that we had in this paper.

---

[9]This problem is also pointed out by Prof. Mean-Young Song (Dongguk University) and Prof. Minpyo Hong (Myongji University). I thank these two professors for their helpful comments.

## 7. Conclusion

In this paper, we examined Questions in English, and analyzed Yes-No Questions and Wh-Questions in the CCG-like system. We saw that Yes-No Questions and Wh-Questions can be analyzed successfully in the CCG-like system. We also found that the semantic interpretations of Yes-No Questions and Wh-Questions could be calculated *compositionally* from those of the constituents.

In sum, this paper provides *computational* algorithms for English Questions, by which we can effectively handle and implement Yes-No Questions and Wh-Questions. I hope this study can give us an opportunity to understand Questions in English.

### References

Ajdukiewicz, K. 1935. Die syntaktisch konnexität. *Studia Philisophica* 1, 1-27.

Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton.

Chomsky, N. 1977. On Wh-Movement. In P. Culicover, T. Wasow, and A. Akmajian, eds., *Formal Syntax,* 71-132. New York: Academic Press.

Chomsky, N. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, N. 1986. *Barriers*. Cambridge, MA: MIT Press.

Chomsky, N. 1995. *The Minimalist Program*. Cambridge, MA: MIT Press.

Dowty, D., R. Wall, and S. Peters. 1981. *Introduction to Montague Semantics*. Dordrecht: R. Reidel.

Gazdar, G., K. Klein, G. Pullum, and I. Sag. 1985. *Generalized Phrase Structure Grammar*. Oxford: Blackwell.

Groenendijk, J. and M. Stokhof. 1982. Semantic analysis of Wh-complements. *Linguistics and Philosophy* 5, 175-233.

Groenendijk, J. and M. Stokhof. 1984. Studies on the semantics of questions and the pragmatic of answers. Doctoral Dissertation. University of Amsterdam.

Hamblin, C. 1973. Questions in Montague English. *Foundations of Language* 10, 41-53.

Hong, M. 1999. Semantics of questions. In Kang et al., *Formal Semantics and Description of English,* 417-457. Written in English. Seoul: Hanshin Publishing Co.

Kang, B.-M. 2001. *Categorial Grammar: English Morphology, Syntax, and Type-theoretic Semantics*. Written in Korea. Seoul: Korea University.

Karttunen, L. 1977. Syntax and semantics of questions. *Linguistics and Philosophy* 1, 3-44.
Krifka, M. 1991. A compositional semantics for multiple focus constructions. *Proceedings from Semantics and Linguistic Theory* (SALT), 127-58.
Lee, Y.-h. 2003. Anaphora resolution algorithms with category: English and English. Doctoral Dissertation. University of Illinois at Urbana-Champaign.
Lee, Y.-h. 2004. A computational approach to questions: Examples from Korean. Paper submitted to the Korean Generative Grammar Circle.
Pollard, C. and I. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago: Univ. of Chicago Press.
Rooth, M. 1985. Association with focus. Doctoral Dissertation. University of Massachusetts at Amherst.
Rooth, M. 1992. A theory of focus interpretation. *Natural Language Semantics* 1, 75-116.
Sag, I. and T. Wasow. 1999. *Syntactic Theory: A Formal Introduction*. Stanford, CA: CSLI.
Steedman, M. 1996. *Surface Structure and Interpretation*. Cambridge, MA: MIT Press.
Steedman, M. 2000. *The Syntactic Process*. Cambridge, MA: MIT Press.

Yong-hun Lee
Dept. of English Language and Literature
Chungnam National University
220 Gung-dong, Yuseong-gu
Daejeon, 305-764, Korea
Phone: 042) 821-5331

Dept. of English Language and Literature
Hannam University
133 Ojeong-dong, Daedeok-gu
Daejeon, 306-791, Korea
Phone: 042) 629-7320
E-mail: ylee@cnu.ac.kr

received: January 30, 2004
accepted: May 27, 2004