A Computational Approach to Definite NPs*

Yong-hun Lee
(University of Illinois at Urbana-Champaign)

Lee, Yong-hun. 2003. **A Computational Approach to Definite NPs.** *Korean Journal of English Language and Linguistics* 3-1, 89-108.
As pronouns are resolved with their antecedents, definite NPs may enter into the anaphora-antecedent relations with indefinite NPs. This paper is to provide faster and more efficient computational algorithms by which definite NPs are resolved effectively, For this purpose, this paper extends Chierchia's Binding Theory in Categorial Grammar, and definite NPs are resolved with their antecedents by similar algorithms that are used to reflexive resolution. In these algorithms, the relations between indefinite NP and definite NP are represented with λ-expressions, and definite NPs are resolved with their antecedent by λ-conversions.

Key Words: computational algorithms, anaphora-antecedent relations, definite NPs

1. Introduction

As pronouns require antecedents, definite NPs may also enter into the anaphora-antecedent relations with indefinite NPs. (1) demonstrates an example.[1]

(1) **A cat$_k$** came. **The cat$_k$** mewed.

(1) has two sentences. The first sentence has an indefinite NP

---

*The system introduced in this paper is computationally implemented in JAVA as a partial fulfillment of Lee (in prep.). I hope to thank Peter Lasersohn for useful comments on earlier drafts of this paper.

[1] In the examples below, relevant expressions are marked by boldfacing. Relevant co-reference relations between a definite NP and its antecedent are represented by subscribed indexes.

a cat, and the second sentence has a definite NP *the cat*. As the co-indexing in (1) indicates, a definite NP *the cat* in the second sentence refers to an indefinite NP *a cat* of the first sentence.

The pattern of the definite NP *the cat* and the indefinite NP *a cat* is similar to those of pronouns and their antecedents, in that the former refers to the latter. (2) illustrates this fact.

(2) a. **John**$_i$ loves **Mary**$_j$. **He**$_i$ loves **her**$_j$.
    b. **John**$_i$ loves **Mary**$_j$. **She**$_j$ loves **him**$_i$.

In (2a), *he* and *her* in the second sentence refer to *John* and *Mary* in the first sentence respectively. In (2b), *she* and *him* in the second sentence refer to *Mary* and *John* in the first sentence respectively. As comparison of (1) and (2) implies, the relations between definite NPs and indefinite NPs can be captured by similar mechanisms that are used to resolve the pronouns with their antecedents. In (1), the definite NP *the cat* of the second sentence is resolved with the indefinite NP *a cat* of the first sentence. The relation between *the cat* and *a cat* in (1) are similar to the relations between pronouns and their antecedents in (2). Consequently, resolution algorithms for definite NPs can be developed similarly.

The goal of this paper is to develop faster and more efficient *computational* algorithms for definite NPs in English. Here, the terminology *computational* has dual meaning. One is *operations on representations* à la O'Grady (1998, 1999), and the other is *computation implementations*. Accordingly, computational efficiency is also crucial in addition to theoretical discussions. This paper does not provide specific implementational algorithms, but it presupposes computational implementations. Specific implementational algorithms are included in Lee (in prep.).

This paper is organized as follows. Section 2 discusses two types of previous approaches to definite NPs. Section 3 introduces

Categorial Grammar, Steedman's (1996, 2000) Combinatory Categorial Grammar (CCG), and Binding Theory in the framework. Section 4 introduces a CCG-like system, and develops definite NP resolution algorithms based on this CCG-like system. Section 5 summarizes this paper.

2. Two Types of Approaches to Definite NPs

Previous approaches to *definiteness* can be divided into two types. One is *uniqueness approach* (Russell 1905, Montague 1974) and the other is *familiarity approach*. (Heim 1982, 1983).

In Montague Grammar, Russell's *uniqueness approach* to *definiteness* is adopted, and *a*(*n*) and *the* are translated as in (3).

(3) Translation of *a* and *the* (Dowty et al., 1981:195)
  a. a(n) : $\lambda P[\lambda Q \exists x[P\{x\} \wedge Q\{x\}]]$
  b. the : $\lambda P[\lambda Q \exists y[\forall x[P\{x\} \leftrightarrow x=y] \wedge Q\{y\}]]$

According to these translations, two sentences in (1) can be interpreted as in (4).

(4) Semantic Interpretations of Two Sentences in Uniqueness Approach
  a. A cat came : $\exists x_1[\text{cat}'\{x_1\} \wedge \text{come}'\{x_1\}]$
  b. The cat mewed : $\exists y[\forall x_2[\text{cat}'\{x_2\} \leftrightarrow x_2=y] \wedge \text{mew}'\{y\}]$

Here, note that all the variables in (4) are bound by quantifiers $\forall$ or $\exists$.

Heim (1982, 1983) took a little different approach, and explained *definiteness* by *novelty* vs. *familiarity*, rather than *uniqueness*. That is, an indefinite NP introduces a *novel* entity into the discourse context, whereas a definite NP refers to an entity familiar to us, i.e., the entity that has already been introduced into the discourse context.

In the *familiarity approach*, the structure of (1) is assumed to be in (5). Here, T stands for *text*, i.e., discourse context, and e_1 and e_2 are the traces left after NP_1 and NP_2 are moved out.

(5) Structure of (1) in Familiarity Approach

$[_T\ [_S\ [_{NP1}\ \text{a cat}]\ e_1\ \text{came}]\ [_S\ [_{NP2}\ \text{the cat}]\ e_2\ \text{mewed}]\ ]$

Based on this structure, two sentences in (1) are translated as in (6).

(6) Semantic Interpretations of Two Sentences in Familiarity Approach

a. A cat came : $[\text{cat}'(x_1) \wedge \text{come}'(x_1)]$

b. The cat mewed : $[\text{cat}'(x_2) \wedge \text{mew}'(x_2)]$

In (6), note that two variables x_1 and x_2 are free in the translations. That is, both variables are not bound by any quantifier in these translations. In order to overcome this problem, Heim supposed that there is an existential quantifier $\exists$ in the T level, and this quantifier binds all the free variables. That is, the final semantic interpretation of (1) in Heim's theory becomes as in (7).

(7) Semantic Interpretation of (1) in Familiarity Approach

$\exists x_1[\text{cat}'(x_1) \wedge \text{come}'(x_1) \wedge \text{mew}'(x_1)]$

In (7), the variables x_1 is bound by $\exists$ that is located in the T level.

Let's compare two approaches, i.e., *uniqueness approach* and *familiarity approach*, from theoretical and implementational perspectives. Their semantic interpretations are repeated below again.

(4) Semantic Interpretations of Two Sentences in Uniqueness Approach

a. A cat came : $\exists x_1[\text{cat}'\{x_1\} \wedge \text{come}'\{x_1\}]$
b. The cat mewed : $\exists y[\forall x_2[\text{cat}'\{x_2\} \leftrightarrow x_2 = y] \wedge \text{mew}'\{y\}]$

(6) Semantic Interpretations of Two Sentences in Familiarity Approach
a. A cat came : $[\text{cat}'(x_1) \wedge \text{come}'(x_1)]$
b. The cat mewed : $[\text{cat}'(x_2) \wedge \text{mew}'(x_2)]$

If we would utilize semantic interpretations in (4), we would have theoretical and implementational problems. The theoretical problem comes from *uniqueness* of *definiteness*. Since a definite NP refers to a unique entity, it would raise a problem if the referred entity is an empty set or the definite NP may refer to entities.

The semantic interpretations in (4) also raise some problems in the implementations. In (4), the relations between *a cat* and *the cat* can be captured by manipulating the variables x_1 and x_2 in semantic representations. But, these manipulations have the following problems. First, the semantic representation in (4) is complicated, and it is difficult to implement these representations computationally, because the quantifiers $\forall$ and $\exists$ are difficult to represent with computer keyboards. Second, semantic interpretations in (4) are more complicated than those of (6). More complicated representations make implementational algorithms more complex, increasing time and space complexity. It would be better if we can avoid this inefficiency.

Consequently, this paper adopts Heim's *familiarity approach* to *definiteness*, and makes use of the semantic interpretations in (6). Because semantic interpretation in (6) has free variables, we have Existential Closure so that these free variables can be bound as in (7). By putting resolution algorithms between (6) and Existential Closure in (7), this paper avoids the problem that weird operations are performed across quantifiers.

3. Binding Theory in Categorial Grammar

Categorial Grammar was first introduced by Ajdukiewicz (1935) and later modified and advanced by Bar-Hillel, Curry, and Lambek. In this framework, we have two basic categories *n* and *s*, and other categories come from the combinations of these two categories. All the syntactic phenomena are described and analyzed by the functor-argument relations of the constituents.

Steedman (1996, 2000) extended previous studies in Categorial Grammar and developed Combinatorial Categorial Grammar (CCG). The most important characteristic of his system is that predicate-argument relations are projected by the combinatory rules of syntax, and other operations are based on these relations (Steedman, 2000:38). The most fundamental combinatory rule is *functional application*, which are delineated in (8). Here, f is the semantic interpretation of the functor category, and a is that of the argument.

(8) Functional Application (Steedman, 1996:13, 2000:37)
a. $X \backslash Y : f \quad Y : a \rightarrow X : f\,a$ (>)
b. $Y : a \quad X \backslash Y : f \rightarrow X : f\,a$ (<)

Chierchia applied Categorial Grammar to explain Binding phenomena in English, and he described syntactic constraints of reflexives and pronominals as follows.

(9) Binding in Categorial Grammar (Chierichia, 1988:134)
a. A reflexive must be bound to an F-commanding argument in its minimal NP or S domain.
b. A non-reflexive pronoun must not be co-indexed with anything in its minimal NP or S domain.
where F-command is simply c-command at

function-argument structure.[2]

Agreement in *number* and *gender* must hold between pronouns and their antecedents, so that pronouns can refer to their antecedents. The constraint for checking agreement is stated in (10a). FT(*n*) in (10b) has three information: *n* is the index of the NP, *gndr* is gender, and *nmbr* is number (Chierchia, 1988:132).

(10) Agreement between Antecedent and Pronouns

a. $FT(n) \approx FT(m)$: The features associated with *n* are non-distinct from those associated with *m*.

b. $$FT(n) = \begin{bmatrix} n \\ \text{gndr} \\ \text{nmbr} \end{bmatrix}$$

For example, the FTs of three different NPs *John*, *himself*, and *her* can be stated as follows.[3]

(11) John_1 himself_2 her_3

$$FT(1) = \begin{bmatrix} 1 \\ \text{male} \\ 3 \end{bmatrix} \quad FT(2) = \begin{bmatrix} 2 \\ \text{male} \\ 3 \end{bmatrix} \quad FT(3) = \begin{bmatrix} 3 \\ \text{female} \\ 3 \end{bmatrix}$$

---

[2] This condition has different predictions for the sentences in (i) and (ii). It rules in (i), but rules out (ii) (Chierchia, 1988:135).

(i) Mary showed the men each other.
(ii) *Mary showed each other the men.

Binding Theory before GB (Chomsky, 1981) says that both sentences are grammatical because tripartite structure is possible for (i) and (ii). In those tripartite structures, *each other* and *the men* c-command each other. But, according to Larson's analyses (1988) with VP-shells, (i) and (ii) can be clearly distinguished from each other, because *the men* c-commands *each other* in (i), but the latter does not c-command the former in (ii). This is also pointed out in Lee (2001) and Lee (2002).

[3] Note that *John*, *himself*, and *her* stand for an R-expression, a reflexive, and a pronominal, respectively.

Chierchia introduced resolution algorithms for pronouns based on the combinatorics, and they are enumerated in (12).

(12) Chierchia's Algorithms (1988:138-9)[4]

a. $\underset{0}{\text{TV}} + \underset{1}{\text{NP}} \Rightarrow \underset{2}{\text{IV}}$ (here and throughout integers will be used as names for the categories mentioned in the rules)

conditions: (i) $\text{LPS}(0) \cap \text{LPS}(1) = \emptyset$ non-coreference
(ii) $\text{SLASH}(2) \cap (\text{LPS}(1) \cup \text{LPS}(2)) = \emptyset$ crossover
(iii) $\text{SLASH}(2) = \text{SLASH}(0) \cup \text{SLASH}(1)$ slash-percolation
(iv) $\text{LPS}(2) = \text{LPS}(0) \cup \text{LPS}(1)$ LPS-percolation

b. $\underset{0}{\text{S/NP}_n} + \underset{1}{\text{NP}_n} \Rightarrow \underset{2}{\text{S}}$

conditions: (i) $\text{LPS}(2) = \emptyset$[5] A-opacity boundary
(ii) $\text{SLASH}(2) \cap (\text{LPS}(1) \cup \text{LPS}(2)) = \emptyset$ crossover
(iii) $\text{SLASH}(2) = \text{SLASH}(0) \cup \text{SLASH}(1)$ slash-percolation
(iv) $n_{+\text{refl}} \notin \text{LPS}(0) \cup \text{LPS}(1)$ reflexive

c. $\underset{0}{\text{IV/IV}} + \underset{1}{\text{IV}} \Rightarrow \underset{2}{\text{IV}}$

conditions: (i) $\text{LPS}(2) = \text{LPS}(0)$ A-opacity boundary
(ii) $\text{SLASH}(2) \cap (\text{LPS}(1) \cup \text{LPS}(2)) = \emptyset$ crossover
(iii) $\text{SLASH}(2) = \text{SLASH}(0) \cup \text{SLASH}(1)$ slash-percolation
(iv) $n_{+\text{refl}} \notin \text{LPS}(1)$ reflexive

d. Reflexives

(i) $\underset{n[+\text{refl}] \in \text{LPS}}{A} \Rightarrow \underset{n \notin \text{LPS}}{A}$
(ii) conditions: (a) A = IV, TV (b) $\text{FT}(A) \approx \text{FT}(n)$
(iii) translation: $\lambda x_n\, [A'(x_n)]$

According to these algorithms, two pronouns *himself* and *him* are resolved as in (13) and (14). Here note that the translations

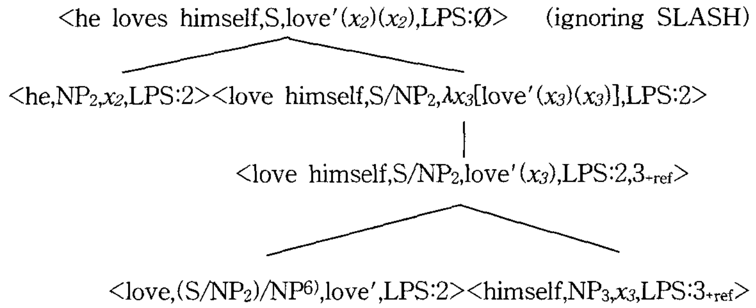
---

[4] LPS (Local Pronoun Store) stores indices for pronouns and their antecedents, and SLASH is similar to that of HPSG.

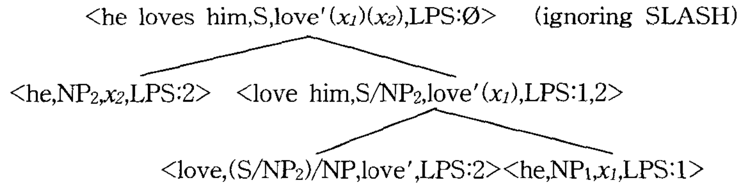
[5] As discussed in Lee (in prep.), this condition is English-specific. It must be modified to deal with long-distance reflexives.

of two sentences are different, i.e., *love'*$(x_2)(x_2)$ and *love'*$(x_1)(x_2)$.

(13) He loves himself.



(14) He loves him.



Because *himself* is a reflexive, the algorithms in (12d) is applied in the analysis in (13). By (12di), the reflexive index $3_{+\text{ref}}$ is erased from the LPS of S/NP$_2$. By (12dii), *A* becomes S/NP$_2$, which is equivalent to IV. By (12diii), the semantic interpretation changes *love'*(x_3) into λx_3[*love'*$(x_3)(x_3)$], where *A'* is *love'*(x_3). The reflexives *himself* is resolved with *he*, when S/NP$_2$ *love himself* meets NP$_2$ *he*. These operations do not occur in (14) because *him* is a pronominal, which results in different semantic interpretations between a reflexive *himself* and a pronominal *him*.

4. Resolving Definite NPs in Categorial Grammar

---

[6] Here 2 in (S/NP$_2$)/NP means the index of subject NP must be 2. It is a tool for subject-predicate agreement.

4.1. A CCG-like System (Lee, 2001, 2002)7)

The system that this paper develops is a CCG-like system, which has been introduced in Lee (2001, 2002). It is basically an incorporation of Chierchia's ideas into Steedman's Combinatory Categorial Grammar (CCG). This system is similar to Steedman's system in that surface combinatorics triggers other operations, especially definite NP resolution algorithms in this paper. It is different from Steedman's in that it makes use of attribute-value ordered pairs (**avop**) in (15) to describe syntactic dependencies of constituents. The six attributes are explained in (16). Here, note a CCG-like system uses NPS rather than Chierchia's LPS. NPS is different from LPS in that it is available beyond S nodes.

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

(16) Six 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. AGR
(i) agreement feature
(ii) index, type, gender, and number
d. TRANS
(i) semantic interpretation

---

[7] A different, but related, version of CCG-like system is developed by Park (2001a, 2001b) and Lee & Park (2001). Their focus is on how Korean Case markers can be handled in Categorial Grammar. Because this paper concerns only definite NP resolution, it will not illustrate how Case markers can be dealt with in Categorial Grammar. For a theoretical approach and its importance, see Park (2001a, 2001b). For its computational implementation, see Lee & Park (2001).

  (ii) based on Montagovian semantics
- e. NPS (NP Index Store)
  - (i) something like a Cooper-storage
  - (ii) has indices of NPs
- f. SLASH
  - (i) similar in HPSG, except that it deals with pronouns
  - (ii) necessary to deal with crossover phenomena

The *functional application* on the CAT values, i.e., *categories*, triggers operations on TRANS and NPS values, and all the definite NPs are resolved by these operations. In (15), AGR and SLASH are parenthesised, because the values for these two attributes will be omitted from the actual representations.

4.2. Resolving Definite NPs in Categorial Grammar

Now, let's develop resolution algorithms of definite NPs within the CCG-like system. The basic idea is that definite NPs can be resolved with indefinite NPs by similar algorithms that are demonstrated in the analysis in (13).

The first thing that we have to do is to differentiate two types of determiners, i.e., *indefinite* and *definite*. In this paper, these two types of determiners are distinguished as in (17).

(17) Two Index Types of Determiners
  a. *n* [+indef] : Indefinite
  b. *n* [+def] : Definite

These indexes will be stored in NPSs with other kinds of NP indexes, and *n*[+def] will trigger resolution algorithms for definite NPs, as reflexive resolution algorithms in (12d) is initiated by *n*[+refl].

The next job is to define semantic interpretations of *a*(*n*) and *the*. This paper takes *familiarity approach*, rather than *uniqueness approach*, and represents the **avop**s of *a* and *the* as in (18).

(18) **avop** for *a*(*n*) and *the*

a. a(n) : <a,NP_i/N,$\lambda P[\lambda Q[P(x_i) \wedge Q(x_i)]]$,NPS:$i_{+indef}$,SLASH:$\emptyset$>

b. the : <the,NP_j/N,$\lambda P[\lambda Q[P(x_j) \wedge Q(x_j)]]$,NPS:$j_{+def}$,SLASH:$\emptyset$>

The *category* for determiners is NP/N. That is, determiners take the category N, such as *cat*, producing the category NP, such as *a cat* or *the cat*. Note that semantic interpretations of *a*(*n*) and *the* are identical, i.e., $\lambda P[\lambda Q[P(x_n) \wedge Q(x_n)]]$ where *n* is the index of the determiner. Note that definite NPs and indefinite NPs are differentiated by the index type, [+indef] vs. [+def]. Also note that this index is attached to the *category* NP_n/N, making this NP either *indefinite* or *definite*.

Now that we have all the tools that are necessary to analyze the sentences in (1), let's see how these sentences are represented and how this sentence is analyzed in the CCG-like system.

(1) **A** $\mathbf{cat_k}$ came. **The** $\mathbf{cat_k}$ mewed.

As in the analyses in (13) or (14), each constituent of the two sentences in (1) are combined as in (19a) and (19b), respectively.

(19) a. <a cat came,S,[cat'$(x_1) \wedge$come'(x_1)],NPS:1_{+indef}>

<a cat,NP_1,λQ[cat'$(x_1) \wedge Q(x_1)$],NPS:1_{+indef}><came,S\\NP_1,come',NPS:1>

<a,NP_1/N,$\lambda P[\lambda Q[P(x_1) \wedge Q(x_1)]]$,NPS:$1_{+indef}$><cat,N,cat',NPS:1>

b. <the cat mewed,S,[cat'$(x_2) \wedge$mew'(x_2)]],NPS:2_{+def}>

<the cat,NP_2,λQ[cat'$(x_2) \wedge Q(x_2)$]],NPS:2_{+def}><mewed,S\\NP_2,mew',NPS:2>

<the,NP_2/N,$\lambda P[\lambda Q[P(x_2) \wedge Q(x_2)]]$,NPS:$2_{+def}$><cat,N,cat',NPS:2>

The next step is to combine the two sentences in (1). In order to combine them in discourse contexts, this paper adopts the discourse handling algorithms that are developed in Lee (2002). In addition to category combinatorics in (12a) and (12b), we have discourse rules in (20) to combine sentences in the discourse contexts.

(20) Discourse Rules (Lee 2002:235)

1. D → D/S S
2. D → S

When the first sentence *A cat came* comes into the discourse context, by Discourse Rule 2, the CAT value of the **avop** in (19a) is changed from S into D as in (21).

(21) Conversion of *A cat came* by Discourse Rule 2

a. <a cat came,S,[cat′(x_1)$\wedge$come′(x_1)],NPS:1$_{+\mathrm{indef}}$>

$$\downarrow$$

b. <a cat come,D,[cat′(x_1)$\wedge$come′(x_1)],NPS:1$_{+\mathrm{indef}}$>

When the next sentence *The cat mewed* enters into the discourse context, we may apply Discourse Rule 1 to combine this sentence with *A cat came*. As we can find in (21), however, the **avop** in (21) cannot be applicable directly, because Discourse Rule 1 has the category D/S whereas the **avop** in (21) has the category D. Accordingly, the **avop** in (21) has to be converted so that Discourse Rule 1 can be available. Because CAT and TRANS values are closely connected in the CCG-like system, conversion of CAT value accompanies that of TRANS value. These two conversions are described in (22).

(22) CAT and TRANS Conversion (Lee 2002:235)

a. CAT : D ⇒ D/S

b. TRANS : $\alpha \Rightarrow \lambda\phi[\alpha\wedge\phi]$

Here, ϕ acts a placeholder for the next proposition. It is a tool that takes basic ideas of Groenendijk and Stokhof's Dynamic Montague Grammar (1990) and Dynamic Predicate Logic (1991). According to (22), the **avop** in (21) is converted as in (23).

(23) Conversion of *A cat came* for Discourse Rule 1

a. $<\text{a cat came,D,[cat}'(x_1)\wedge\text{come}'(x_1)\text{],NPS:}1_{+\text{indef}}>$

$$\downarrow$$

b. $<\text{a cat come,D/S,}\lambda\phi\text{[[cat}'(x_1)\wedge\text{come}'(x_1)]\wedge\phi\text{],NPS:}1_{+\text{indef}}>$

Now, two sentences in (1) can be combined by Discourse Rule 1. For the CAT value, S in D/S of *A cat came* is cancelled out with the S of *The cat mewed.* For the TRANS value, two sentences are combined as in (24).

(24) Combination of Two Sentences (Semantic Interpretation)

a. A cat came := $[\text{cat}'(x_1)\wedge\text{come}'(x_1)]$

b. The cat mewed := $[\text{cat}'(x_2)\wedge\text{mew}'(x_2)]$

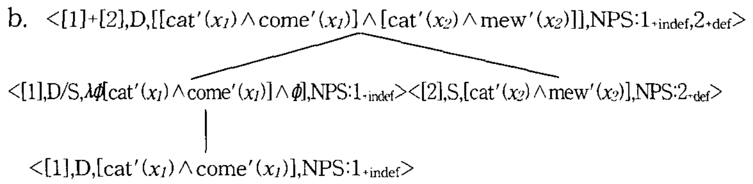
c. A cat came. The cat mewed. :=

$$[\text{cat}'(x_1)\wedge\text{come}'(x_1)] + [\text{cat}'(x_2)\wedge\text{mew}'(x_2)]$$
$$\Rightarrow \lambda\phi[[\text{cat}'(x_1)\wedge\text{come}'(x_1)]\wedge\phi]([\text{cat}'(x_2)\wedge\text{mew}'(x_2)])$$
$$\Rightarrow [[\text{cat}'(x_1)\wedge\text{come}'(x_1)]\wedge[\text{cat}'(x_2)\wedge\text{mew}'(x_2)]]$$

(25) summarizes these combination processes. Here, note the following two points: (i) two variables, i.e., x_1 and x_2, are still free and (ii) *the cat* is not connected with *a cat* yet.

(25) a. [1]A cat came. [2]The cat mewed.

b.



$<\text{[1]+[2],D,[[cat}'(x_1)\wedge\text{come}'(x_1)]\wedge[\text{cat}'(x_2)\wedge\text{mew}'(x_2)]\text{],NPS:}1_{+\text{indef}},2_{+\text{def}}>$

$<\text{[1],D/S,}\lambda\phi\text{[cat}'(x_1)\wedge\text{come}'(x_1)]\wedge\phi\text{],NPS:}1_{+\text{indef}}>$ $<\text{[2],S,[cat}'(x_2)\wedge\text{mew}'(x_2)\text{],NPS:}2_{+\text{def}}>$

$<\text{[1],D,[cat}'(x_1)\wedge\text{come}'(x_1)\text{],NPS:}1_{+\text{indef}}>$

When another sentence enters into this discourse context, Discourse Rule 1 is applied again, and the CAT and the TRANS conversions in (22) are applied to the top D node in (25). This combination processes continue recursively until all the sentences in the discourse contexts are exhausted.

Now, we have two indexes in NPS of the topmost D node, $\{1_{+\text{indef}}, 2_{+\text{def}}\}$. At this point, the index $2_{+\text{def}}$ triggers Definite NP Resolution algorithms, as $3_{+\text{ref}}$ in (13) triggers the resolution algorithms in (12d). (26) is the resolution algorithms that this paper proposes. Note that these algorithms have similar forms that are shown in (12d).

(26) Definite NPs

(i) $$\underset{j[+\text{def}]\in \text{NPS}}{A} \Rightarrow \underset{j\notin \text{NPS}}{A}$$

(ii) conditions: (a) A = S, D

(b) $\text{FT}(j) \approx \text{FT}(i)$, where $i_{[+\text{indef}]}$ is a potential antecedent

(iii) translation: $\lambda x_j[\ \ldots\ \text{P}(x_j)\ \ldots\ \text{Q}(t_i)\ \ldots\ ](t_i)$, where (i) t_i is the translation of $\text{NP}_{i[+\text{indef}]}$, (ii) $A' = [\ \ldots\ \text{P}(x_j)\ \ldots\ \text{Q}(t_i)\ \ldots\ ]$, (iii) P and Q are one-place predicates, and (iv) P = Q.

If we combine this algorithm with the category combinatorics in (12a) and (12d), Definite NP Resolution Algorithms in the CCG-like system is completed as in (27). Here, $(S\backslash NP_n)/NP$ is equal to TV and $S\backslash NP_n$ corresponds to IV in (12a).

(27) Definite NP Resolution Algorithms in English

a. $\underset{0}{(S\backslash NP_n)/NP} + \underset{1}{NP} \Rightarrow \underset{2}{S\backslash NP_n}$ (here and throughout integers will be used as names for the categories mentioned in the rules)

conditions: (i) $\text{NPS}(0) \cap \text{NPS}(1) = \emptyset$ non-coreference

(ii) $\text{SLASH}(2) \cap (\text{NPS}(1) \cup \text{NPS}(2)) = \emptyset$ crossover

(iii) $\text{SLASH}(2) = \text{SLASH}(0) \cup \text{SLASH}(1)$ slash-percolation

(iv) $\text{NPS}(2) = \text{NPS}(0) \cup \text{NPS}(1)$ NPS-percolation

b. $\underset{0}{NP_n} + \underset{1}{S\backslash NP_n} \Rightarrow \underset{2}{S}$

conditions: (i) $\mathrm{NPS}(2)=\emptyset$ A-opacity boundary
(ii) $\mathrm{SLASH}(2)\cap(\mathrm{NPS}(1)\cup\mathrm{NPS}(2))=\emptyset$ crossover
(iii) $\mathrm{SLASH}(2)=\mathrm{SLASH}(0)\cup\mathrm{SLASH}(1)$ slash-percolation
(iv) $n_{+\mathrm{refl}} \notin \mathrm{NPS}(0)\cup\mathrm{NPS}(1)$ reflexive

c. Definite NPs
(i) $A \Rightarrow A$
$j[+\mathrm{def}]\in\mathrm{NPS}$ $j\notin\mathrm{NPS}$
(ii) conditions: (a) A = S, D
(b) $\mathrm{FT}(j)\approx\mathrm{FT}(i)$, where $i_{[+\mathrm{indef}]}$ is a potential antecedent
(iii) translation: $\lambda x_j[\ \ldots\ \mathrm{P}(x_j)\ \ldots\ \mathrm{Q}(t_i)\ \ldots\ ](t_i)$, where (i) t_i is the translation of $\mathrm{NP}_{i[+\mathrm{indef}]}$, (ii) $A' = [\ \ldots\ \mathrm{P}(x_j)\ \ldots\ \mathrm{Q}(t_i)\ \ldots\ ]$, (iii) P and Q are one-place predicates, and (iv) P = Q.

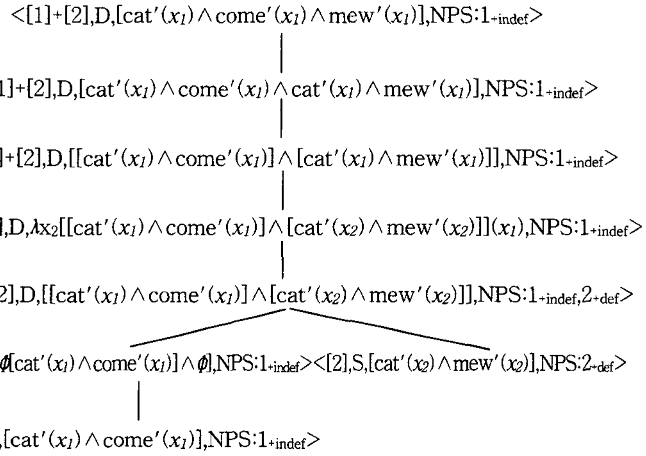
Now, let's take the analysis in (25) and see how the resolution algorithms work. After the two sentences, i.e., *A cat came* and *The cat mewed*, are combined, we have the NPS $\{1_{+\mathrm{indef}}, 2_{+\mathrm{def}}\}$. Here, $2_{+\mathrm{def}}$ triggers the resolution algorithms in (27c). Searching domain of an antecedent is the indexes stored in the NPS. Possible antecedents are selected by the following criteria: the indexes for potential antecedents are smaller than that of the definite NP. In our example, the searching domain is $\{1_{+\mathrm{indef}}\}$, because 1 is the only index that is smaller than 2.

The antecedent for the definite NP is decided by the following criteria: (i) index type of i is [+indef] and (ii) $|j-i|$ is minimum. In our example, the index $1_{+\mathrm{indef}}$ satisfies these criteria. By (27ci), $2_{+\mathrm{def}}$ is deleted from the NPS. The two conditions in (27cii) are satisfied, since A=D and $\mathrm{FT}(2)\approx\mathrm{FT}(1)$. By (27ciii), the TRANS value is changed from $[[\mathrm{cat}'(x_1)\wedge\mathrm{come}'(x_1)]\wedge[\mathrm{cat}'(x_2)\wedge\mathrm{mew}'(x_2)]]$ into $\lambda x_2[[\mathrm{cat}'(x_1)\wedge\mathrm{come}'(x_1)]\wedge[\mathrm{cat}'(x_2)\wedge\mathrm{mew}'(x_2)]](x_1)$. $\mathrm{P}(x_j)$ and $\mathrm{Q}(t_i)$ in (27ciii) are $\mathit{cat}'(x_1)$ and $\mathit{cat}'(x_2)$ in this translation. They correspond to *a cat* and *the cat* in (1), respectively. The fourth condition in (27ciii) formalizes the intuition that the common noun combined with *the* must be identical that combined with *a(n)*. This semantic interpretation now goes through a λ

-conversion, and *the cat* is resolved with *a cat* during this process. This interpretation, i.e., $[\text{cat}'(x_1) \wedge \text{come}'(x_1) \wedge \text{cat}'(x_1) \wedge \text{mew}'(x_1)]$, has the duplication of $\text{cat}'(x_1)$, and one of them is deleted. At last, we have the final result $[\text{cat}'(x_1) \wedge \text{come}'(x_1) \wedge \text{mew}'(x_1)]$. All the resolution processes are summarized in (28).

(28) a. [1]A cat came. [2]The cat mewed.

b.



Note the following two things. First, NPS of the topmost D node has no indexes whose type is [+def]. Therefore, definite NP algorithms in (27c) are not triggered anymore. Second, as the semantic interpretation indicates, *a cat* and *the cat* are related through the variable x_1.

The final step is to apply Existential Closure à la Heim (1982, 1983), for binding all the free variables in the interpretation. In our example, we have only one free variable x_1, thus only this variable has to be bound. Accordingly, the final interpretation of the sentence (1) becomes (29). Here, note that x_1 is bound with the existential quantifier $\exists$.

(29) Final Interpretation of Sentence (1)

$$\exists x_1[\text{cat}'(x_1) \wedge \text{come}'(x_1) \wedge \text{mew}'(x_1)]$$

Conclusively, along with the discourse handling algorithms and the definite NP resolution algorithm in (27), we can process the sentence (1) and resolve definite NPs in the discourse contexts successively.

Now, we succeed to analyze the sentence (1). This sentence has one indefinite NP and one definite NP. But, the definite NP resolution algorithm in (27c) can be applied more than one time in order to capture indefinite NP-definite NP relation. For example, two definite NPs in (30) can be resolved with the same algorithms. Compare the patterns of (30) with those of (2).

(30) a. **A cat**$_k$ came. **A dog**$_l$ came. **The cat**$_k$ mewed.
The dog$_l$ barked.
b. **A cat**$_k$ came. **A dog**$_l$ came. **The dog**$_l$ barked.
The cat$_k$ mewed.

(2) a. **John**$_i$ loves **Mary**$_j$. **He**$_i$ loves **her**$_j$.
b. **John**$_i$ loves **Mary**$_j$. **She**$_j$ loves **him**$_i$.

As we can find, (30a) corresponds to (2a), and (30b) to (2b). Two pairs of sentences in (30) have two indefinite NP-definite NP relations: (i) *a cat* - *the cat* and (ii) *a dog* - *the dog*. But, the resolution algorithms are the same, whether they are applied to *the cat* or *the dog*. The algorithms in (27c) are triggered twice. One is by *the cat* and the other is *the dog*. Though this paper does not analyze these sentences, two definite NPs in (30) is resolved by the same resolution algorithms, i.e., by (27c).

5. Conclusion

This paper examined the indefinite NP-definite NP relations in English, and developed resolution algorithms for definite NPs.

The algorithms were developed by combining Chierchia's Binding Theory and Steedman's CCG, which is named a CCG-like system.

In this CCG-like system, the indefinite NP-definite NP relations are represented by λ-expressions, and definite NP are resolved with its antecedent indefinite NP ultimately by λ-conversions. I hope that this research can give us an opportunity to understand behaviors of definite NPs properly.

References

Ajdukiewicz, K. 1935. Die Syntaktische Konnexität. *Studia Philisophica* 1, 1-27.

Chierchia, G. 1988. Aspects of a categorial theory of binding. In R. Oehrle et al., eds., *Categorial Grammars and Natural Language Structures*. 125-151. Dordrecht: R. Reidel.

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

Groenendijk J. and M. Stokhof. 1990. Dynamic Montague grammar. In L. Kalman and P. Polos, eds., *Papers from the Second Symposium on Logic and Language*, 3-48. Budapest : Akademiai Kiado.

Groenendijk J. and M. Stokhof. 1991. Dynamic predicate logic. *Linguistics and Philosophy* 14, 39-100.

Heim, I. 1982. *The Semantics of Definitive and Indefinite Noun Phrase*. Ph.D. Dissertation. University of Massachusetts at Amherst.

Heim, I. 1983. File change semantics and the familiarity theory of definiteness. In R. Bauerle, C. Schwarze, and A. Stechow, *Meaning, Use, and Interpretation of Language*, 164-189. Berlin: de Guyter.

Jurafsky, D. and J. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle, NJ : Prentice Hall.

Lee, Y.-H. 2001. Implementation of pronoun readings in English: A categorial grammar approach. *Korean Journal of English Language and Linguistics* 1, 609-627.

Lee, Y.-H. 2002. A computational approach to discourse pronominals. *The Journal of Studies in Language* 18, 223-42.

Lee, Y.-H. in prep. *Anaphora Resolution Algorithms with Category: English and Korea*. Ph.D. Dissertation. Under Progress. University of Illinois at Urbana-Champaign.

Lee, Y.-H. and C. Park. 2001. An analysis of Korean multiple accusative constructions and its computational implementation. A Paper Presented at the Annual Conference of Linguistic Society of Korea.Kyung-Joo, Korea.

Montague, R. 1974. *Formal Philosophy: Selected Papers of Richard Montague*. ed. by R. Thomason. New Haven: Yale University Press.

O'Grady, W. 1998. Korean case: A computational approach, *Proceedings of 11th International Conference on Korean Linguistics*. Honolulu.

O'Grady, W. 1999. Computing Korean. A Paper Presented at the POSCO Conference. University of Texas at Austin.

Park, C. 2001a. Semantic and combinational valences for Korean case markers and COMPs: A bi-directional analysis and two types of Functor-Argument relations. to appear in *Japanese-Korean Linguistics* XI.

Park, C. 2001b. Two types of valence and two types of Functor-Argument relations. *Harvard Studies in Korean Linguistics* IX, 619-629.

Russell. B. 1905. On referring. *Mind* 59, 320-344.

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 Linguistics
University of Illinois at Urbana-Champaign
4088 Foreign Language Building
707 S. Mathews Ave.
Urbana, IL 61801
U.S.A.
Phone: (217) 367-7261
E-mail: ylee@uiuc.edu

received: 2002. 12. 14.
accepted: 2003. 2. 26.