Combinatory Categorial Grammar for Korean

Sung-Kook Han
Dept. of Computer Eng.
Won Kwang University

Chan-Gon Park
Dept. of Computer Science
Chung Ju University

ABSTRACT

A commutative productive category is proposed to the current CCG for the syntactic analysis of free word order languages like Korean. The introduction of this sort of category is quite natural for categorial lexicon and functional operations. We present the theorical basis of productive category and examine the linguistic availability through typical syntactic structures of Korean.

1. INTRODUCTION

Classical categorial grammar formulated by Ajukiewicz(1935) and Bar-Hill(1953) in the decades before generative grammar become the main stream of syntax theory, is based on the idea that the expression of logical and natural languages can be described entirely in terms of the defined categories. In recent years there have been the emerging interests in categorial grammar by means of Steedman(1982, 1987) and Dowty(1988). One recent approach on categorial grammar called Combinatorial Categorial Grammar (CCG) proposes the operations of functional composition and type raising in analysis of noncaonical structures such as Wh-dependecies and nonidentical phrase conjunction. CCG argues that a wide varity of extractions, discontinuous and coordinate structures can be captured by adding operations of functional composition and type raising to CG. On the assumption that syntactic categories directly reflect the semantics of the entity in a type-driven system, CCG can make it possible to combine the syntactic functions and semantic intepretation in one combinatorial category.

Since the functional operations have strict ordering properties, the adaption of the current CCG to free word order languages can not avoid the unusual type raising operation and may be hard to extend the domain of categories. In this paper, we will consider what CCG has to prepare for the free word order languages. We will propose a new type of category called commutative productive category for this facility. We also investigate the linguistic implication concerning this type of category.

2. OVERVIEW OF CCG

CCG consists of two components: categorial lexicon that assigns syntactic categories to lexical elements and a set of combinatory rules related to Curry's combinatory logic.

2.1 Categorial Lexicon

Some syntactic categories, such as N, the category of nouns, are atomic symbols to represent a certain category. The categories may be more complex by combining with atomic symbols to form the so-called functor categories. Functor categories, for example, which combine with arguments to their right will be a category of the form X/Y, which is viewed as a function from categories of type Y to categories of type X. Thus, for example, determiners are NP/N an transitive verbs are VP/NP, respectively. In addition to these, other functor categories which combine with their arguments to the left can be defined by the use of backward slash like X\Y, denoting a combining function from Y into X to the leftward. As the the examples of the leftward combining functor categories, adverbial phrases like "quickly" can be represented in the form of VP\VP and predicate phrases like "arrive" can be represented as S\NP.

Both types of function may have more than on argument, and may mix the two types of slash which represent the direction of application of functions. For example, the tensed transitive verb like "eat" will be (S\NP)/NP. This can be regard as a second order function from NP to another function S NP wich is itself a function from NP to S. All functor categories are unary or curried in this sense. The combinatory rules can be applied to the combination of these functor categories.

2.2 Combinatorial rules

We can derive the two functional application rules in terms of the directionality given to functor categories, which we usually denote as "fa>" (forward functional application) and "fa<" (backward functional application).

(1)
a. Forward functional application (fa>)
```
    X/Y    Y  => X
```
b. Backward functional application (fa<)
```
    X\Y    Y  => X
```

The rules of functional application in (1) can be used for analysis of sentences as shown in (2).

(2)
```
   the  man      ate    the  cake
   NP/N   N   (S\NP)/NP NP/N  N
   ---------fa>         --------fa>
      NP                   NP
             ----------------fa>
                   S\NP
      --------------------fa<
                 S
```

To solve the coordination of contiguous strings that do not constitute consrituients, CCG proposes the grammar to allow certain operations on functions which compose two functor categories. Functions may compose under the rules of function composition. Several functional composition rules are derived in terms of the directionality of categories.

(3) Functional composition
```
a.    X/Y    Y/Z  =>  X/Z (>fc)
b.    X/Y    Y\Z  =>  X\Z (>fx)
c.    Y\Z    X\Y  =>  X\Z (<fc)
d.    Y/Z    X\Y  =>  X/Z (<fx)
```

The following example in (4) shows how functional composition is applied to analyze coordinate sentence.

(4)

```
 I   cooked   and    might     eat  the beans.
NP (S\NP)/NP conj (S\NP)/VP VP/NP     NP
                  ---------------->fc
                      (S\NP)/NP
    ----------------------------coord
             (S\NP)/NP
             --------------------------->fa
                        S\NP
  -----------------------------<fa
             S
```

Function composition is useful for the description of some syntactic structures such as non-constituient coordination and gapping in English.

In some sentences such as topicalization, functional application and composition rules are insufficient to account for these structures. Sometimes we need an operation is used to map arguments into functions over functions which take such arguments. This operation is called "type raising" and can be roughly summarized as follows: an argument of type X can be replaced by a function whose domain is the functions which map objects of type X to objects of type Y and whose codomain is object of type Y(Lee 1989). The type raising is widely used in Montague Grammar and we can derive universal type raising rules for CCG.

(5) Type raising rules

```
a.   X  =>  Y/(Y\X)    (>T)
b.   X  =>  Y\(Y\X)    (<Tx)
c.   X  =>  Y\(Y/X)    (<T)
d.   X  =>  Y/(Y/X)    (>Tx)
```

Type raising is somewhat difficult to understand. In the analysis of sentence in (6), we use type raising for topicalized NP.

(6)

```
 Apple        he        said     John      hates!
  NP          NP      (S\NP)/S    NP     (S\NP)/NP
------->Tx --------->T          -------->T
S/(S/NP)   S/(S\NP)             S/(S\NP)
           ------------------->fc
                S/S
                ---------------------->fc
                       S/(S\NP)
                       ------------------------->fc
                                 S/NP
----------------------------------------->fc
                 S
```

Sometimes this sort of type raising operation seems to be absurd though type raising is generally accepted in many linguistic liternatures. This also provokes serious nondeterministic problems in parsing a sentence. It is inevitable, however, in the free order languages like Korean to frequently use type raising operation in the course of syntactic analysis.

3. AN EXTENSION OF CCG

In this chapter we will propose a category called commutative productive category based on typed lambda calculus. We will show how this type of categories can be applied to free order languages. We will also examine the linguistic implication of this type through the analysis of typical syntactic structures.

3.1 A commutative productive category

Let us consider the types in typed lambda-calculus. Types are defined inductively as follows.

1) A, B, C, are types
2) A -> B is type if A, B are types.
3) A X B is type if A, B are types.
4) A + B is types if A, B are types.

1) and 2) are the minimal requirements for types in typed lambda calculus. 3) and 4) are productive and additive categories we will discuss for the extension of CCG. 3) and 4) are in standard typed lambda calculus, and so these can be easily embedded in CCG. We will pay more attention to introducing productive categoty. For the additive category, however, we will give an example to show the possibility of additive category for CCG.

It is not difficult to introduce productive categories into CCG. As we know, the type construction has the close relationship to natural deduction. Let us review some typical examples of natural deduction.

```
     [x : A] -
        .      |
        .      |
     M : B   -
   -------------------
   ( lambda x:A . M) : A -> B
```

This deduction process implies that we can deduce B from A. Using beta reduction, we can obtain B as follows:

```
   ( lambda x:A . M) : A -> B     N : A
 ----------------------------------------
   ( lambda x:A . M)N : B == M[N/x] : B
```

The same concept of natural deduction can be applied to productive categories as follows:

```
     x : A        Y : B
   ----------------------
        (x,y) : A X B
```

This deduction means that we have prove A $\wedge$ B if we have the proof A and B. The introduction of productive categories allows us using Cartesian Closed Category(CCC), which is a categorial model of typed lambda calculus. One of the interesting results from CCC is

(A X B) -> C is isomorphic to A -> (B -> C).

We will discuss how we can apply this property to CCG at later. Now, let's extend this productive category to have commutative property. By the definition, A X B and B X A are identical but positions. In CCG, however, A X B and B X A can produce different results. For instance,

C / A X B == C / B / A
C / B X A == C / A / B

but C / A X B and C / B X A are isomorphic. Thus, we define {A, B} as either A X B or B X A. Then an extended CCG by adding commutative product categories will be defined as follows:

1) {A}, {B} and {C} are types. If {A} is singleton, {A} can be written as A.
2) {A}/{B}, {A} {B} are types if {A} and {B} are types.
3) {A, B} are types if {A} and {B} are types.

Before we precede linguistic applications, let us review the proposed rules in CCG from the point of view of productive categories. Instead of presenting the entire rules, we will show some typical examples for the clear description of the underlying concepts of commutative productive categories.

(7)

Functional Application:
{A}/{Y, Z}   {Y} => {A}/{Z}
Functional Composition:
{A}/{B}   {B}/{C} => {A}/{C}
{A}/{B, D}   {B}/{C} => {A}/{D}/{C}
Functional production:
{A}   {B} => {A, B}
Distributive Law:
{A/C}   {B/C} => {A, B}/{C}
{A, B}/{C} => {A/C} {B/C}

All the results in CCG can be extended in this manner. The sematic interpretation will be also extended in this fashion. Every function in CCG has to be curried form, but we can accept many-arity functions as well as curried one. Now, we will demonstrate how these newly proposed rules can be applied to free word-order languages like Korean.

3.2 Application of commutative productive categories

Under CCG that use the communtative productive categories, a categorial lexicon will be as follows:

John : N	-ka(NOM) : NPn N	juta : S\{NPa, NPn, NPd}
Mary : N	-lul(ACC) : NPa N	mekta : S\{NPn, NPa}
sakwa : N	-eykey(DAT) : NPd N	
chaik : N		

Note that categories can be uniquely assigned to linguistic elements by using commutative productive categories. This determinism is very useful for the description of syntactic properties and may not be achived in current CCG .

3.2.1 Word order variation

The scrambling phenomenon shown in Korean cab be easily explained in terms of productive categories.

(8)

```
 John-i      Mary-eykey    sakwa-lul        juta.
 N  NPn N    N  NPd N      N  NPa N     S\{NPa, NPd, NPn}
 --------FA  --------FA    ---------FA
  NPn          NPd           NPa
                          ------------------------------FA
                                    S\{NPd, NPn}
              ------------------------------FA
                            S\{NPn}
 ----------------------------FA
                 S
```

We can verify that any order-variated sentences can be describe without adding other mechanisms such as type raising. This analysis is also consistent with linguistic implication. It is known that the current CCG brings about the addendant spurious ambiguity problem on account of its category assignment and type raising(Wittenbug 1988). This problem is more serious in case of free order languages. The productive category can resolve this problem without sacrificing the formal properties of CCG.

The semantic interpretation in this derivation will be given according to usual functional application. The predicate "ju-ta", for example, is represented with the abstraction operator "lambda" as follows:

juta := lambda[x:NPn, y:NPd, z:NPa]. (GIVE x y z) : S

Thus we can get the following semnatic interpreation for the above sentence.

(GIVE (NOM John) (DAT Mary) (ACC sakwa))

We have to remind that scarmbling reminiscent the commutative operations.

3.2.2 Coordination

One of the merits of CCG is of the analysis of coordination(Steedman 1989). Functional composition makes it possible to analyze the varity of coordonationmore efficiently than any other formalisms do. The productive categoty is applied to nonidentical constituient coordination including word order variation. We will give some typical examples to show availability of productive categoty.

(9)

a.

```
John-i   norai-lul      pulu        -ko   Mary-ka  piano-lul  nynjuha-ta.
NPn        NPa     S\{NPn, NPa}    COORD   NPn       NPa     S\{NPn, NPa}
-------------------------------          ------------------------------
           S                                          S
          ---------------------------------------------------
```

```
                                  S
b.
  John-i   sakwa-lul  kuriko  bae-lul    Mary-ka       mek-ta.
  NPn         NPa     COORD    NPa          NPn    S\{NPn, NPa}
   --------------               ---------------
      {NPn,  NPa}               (NPa,  NPn}
      ----------------------------------------
                  {NPa,  NPn}
                  ------------------------------------------------
                                          S
```

We can derive syntactic structures by using functional production of productive categories. Any order variations occurred in coordination is also consistently derived.

3.3.3 Long distance scrambling

The long distance scrambling observed in Korean is related with verbal endings, especially nonfinal verbal endings, of some kinds of predicate such as "malha-ta, sangkakha-ta". On account of discontiunous constituient structures, this phenomenon is hard to explain its syntactic behaviors. We can reveal the syntactic properties of long distance scrambling in terms of introducing productive category. Let us consider the following syntactic analysis.

(10)

```
sakwa-lul John-i Mary-ka Judy-eykey  ju-ess-ta       -ko  Bill-eykey malha-ssta.
 NPa      NPn     NPn         NPd  S\{NPa,NPn,NPd} S'\S     NPd  S\{S',NPn,NPd}
 -----------     -------------------------------          -------------------
 {NPa, NPn}                S\{NPa}                            S\{S', NPn}
   --------------------------------
              {S, NPn}
              ------------------------------------------
                             {S', NPn}
                             -----------------------------------------------
                                                   S
```

In this analysis the preposed argument "sakwa-lul" is treated as a "hole" in functional composition of predicate. This implies that we can confirm the dependency relation concerning the preposed areguments. The functor category S\{NPa} requests a NPa, and after functional composition with {NPa, NPn} it becomes {S, NPn} that has a extra NPn category in S. Note that we can realize the extended domain for discontinuous constituient by means of productive category.

4. CONCULSION

We propose the commutative productive category to current CCG so that we can capture word order variation and its associated phenomena within the current CCG formalsim. The introduction of this type of category is quite natural and CCC gives the theortical basis for this. We show how the productive category is available to free word order languages like Korean. We can assign the unique categoty to linguistic elemnets in categorial lexicon and avoid the unusual application of type raising. This improvement is examined through typical linguistic structures of Korean. The semantic interpretation of course can be derived from typed lambda calculus in spite of using commutative productive category. We can also realize the subcategorization frame of predicate that is important to Korean. This implies that the productive category closes to linguistic implications. It is not hard to introduce a recursive operation to deal with nominal adjuncts and other recursive structures.

ACKNOWLEDGEMENT

The orginal idea of this paper was proposed by the first author for Ji-In Kim, Young-Suk Lee and Mark Steedman, University of Pennslyvania. We intend to report this discussion for the research on CCG. We would like to thank Dr. Steedman, Dr. Joshi, Ji-In Kim, Young-Suk Lee, and especially Jin-Young Choi for their helpful discussion and criticism. Any remaining errors are ours, and ours only.

REFERENCES

1. Ades, Anthony E. and Mark J. Steedman(1982), 'On the order of words', Linguistics and Philosophy, 4, 517-558.

2. Ajdukiewicz, Kazimierz(1935), Polish Logic, edited by Storrs McCall, 1920-1939, Oxford University Press, pp. 207-231.

3. Bar-Hillel, Yehoshua(1953), 'A quasi-arithmetical notation for syntactic description', Language, 29, 47-58.

4. Dowty, David(1988), 'Type raising, functional composition, and non-constituent coordination', paper to the Conference on Categorial Grammar, Tucson, AR, June 1985

5. Lambek, Joachim(1958), 'The mathematics of sentence structure', American Mathematical Monthly, 65, 154-170.

6. Steedman, Mark J.(1987) 'Combinatory grammars and parasitic gaps', Natural Language and Linguistic Theory

7. Steedman, Mark J(1989), 'Constituency and Coordination in a Combinatory Grammar', in, Baltin, Mark, and Tony Kroch, (eds.), New Conceptions of Phrase Structure, Chicago, University Press, 201-306.

8. Szabolcsi, Anna(1987), 'On Combinatory Categorial Grammar', Proceedings of the Symposium on Logic and Language, Debrecen, Akademiai Kiado, Budapest, 151-162.

9. Young-Suk Lee etal.(1989), 'Scrambling and Coordination in Korean: A CCG analysis", ms in University of Pennsylvania, Oct. 1989