Deriving the Predicate-Argument Structure for a Free Word Order Language *

Cem Bozsahin
Department of Computer Engineering
Middle East Technical University
06531 Ankara, Turkey
bozsahin@ceng.metu.edu.tr

Abstract

In relatively free word order languages, grammatical functions are intricately related to case marking. Assuming an ordered representation of the predicate-argument structure, this work proposes a Combinatory Categorial Grammar formulation of relating surface case cues to categories and types for correctly placing the arguments in the predicate-argument structure. This is achieved by treating case markers as type shifters. Unlike other CG formulations, type shifting does not proliferate or cause spurious ambiguity. Categories of all argument-encoding grammatical functions follow from the same principle of category assignment. Normal order evaluation of the combinatory form reveals the predicate-argument structure. The application of the method to Turkish is shown.

1 Introduction

Recent theorizing in linguistics brought forth a level of representation called the Predicate-Argument Structure (PAS). PAS acts as the interface between lexical semantics and d-structure in GB (Grimshaw, 1990), functional structure in LFG (Alsina, 1996), and complement structure in HPSG (Wechsler, 1995). PAS is the sole level of representation in Combinatory Categorial Grammar (CCG) (Steedman, 1996). All formulations assume a prominence-based structured representation for PAS, although they differ in the terms used for defining prominence. For instance, Grimshaw (1990) defines the thematic hierarchy as:

Agent > Experiencer > Goal / Location / Source > Theme

whereas LFG accounts make use of the following (Bresnan and Kanerva, 1989):

Agent > Beneficiary > Goal / Experiencer > Inst > Patient / Theme > Locative.

As an illustration, the predicate-argument structures of the agentive verb *murder* and the psychological verb *fear* are (Grimshaw, 1990, p.8):

murder	(x	(y))
	Agent	Theme
fear	(x	(y))
	Exp	Theme

To abstract away from language-particular case systems and mapping of thematic roles to grammatical functions, I assume the Applicative Hierarchy of Shaumyan (1987) for the definition of prominence:

Primary Term > Secondary Term >
Tertiary Term > Oblique Term.

Primacy of a term over another is defined by the former having a wider range of syntactic features than the latter. In an accusative language, subjects are less marked (hence primary) than objects; all verbs take subjects but only transitive verbs take objects. Terms (=arguments) can be denoted by the *genotype* indices on NPs, such as NP_1, NP_2 for primary and secondary terms.[1] An NP_2 would be a direct object (NP_{acc}) in an accusative language, or an ergative-marked NP (NP_{erg}) in an ergative language. This level of description also simplifies the formulation of grammatical function changing; the primary term of a passivized predicate (PASS p) is the secondary term of the active p. I follow Shaumyan and Steedman (1996) also in the ordered representation of the PAS (1). The reader is referred to (Shaumyan, 1987) for linguistic justification of this ordering.

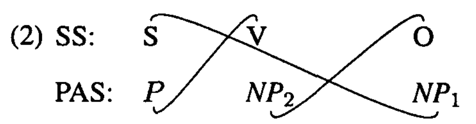
(1) Pred... <Sec. Term><Primary Term>

Given this representation, the surface order of

* Thanks to Mark Steedman for discussion and material, and to the anonymous reviewer of an extended version whose comments led to significant revisions. This research is supported by TUBITAK (EEEAG-90) and NATO Science Division (TU-LANGUAGE).

[1] Shaumyan uses T^1, T^2, but we prefer NP_1, NP_2 for easier exposition in later formulations.

constituents is often in conflict with the order in the PAS. For instance, English as a configurational SVO language has the mapping:

(2)


However, in a non-configurational language, permutations of word order are possible, and grammatical functions are often indicated not by configurations but by case marking. For instance, in Turkish, all six permutations of the basic SOV order are possible, and Japanese allows two verb-final permutations of underlying SOV. The relationship between case marking and scrambling is crucial in languages with flexible word order. A computational solution to the problem must rely on some principles of parsimony for representing categories and types of arguments and predicates, and efficiency of processing.

In a categorial formulation, grammatical functions of preverbal and postverbal NPs in (2) can be made explicit by type shifting[2] the subject to $S/(S\backslash NP_1)$ and the object to $(S\backslash NP_1)\backslash((S\backslash NP_1)/NP_2)$. These categories follow from the order-preserving type shifting scheme (Dowty, 1988):

(3) $NP \Rightarrow T/(T\backslash NP)$ or $T\backslash(T/NP)$

To resolve the opposition between surface order and the PAS in a free word order language, one can let the type shifted categories of terms proliferate, or reformulate CCG in such a way that arguments of the verbs are sets, rather than lists whose arguments are made available one at a time. The former alternative makes the spurious ambiguity problem of CG parsing (Karttunen, 1989) even more severe. Multiset CCG (Hoffman, 1995) is an example of the set-oriented approach. It is known to be computationally tractable but less efficient than the polynomial time CCG algorithm of Vijay-Shanker and Weir (1993). I try to show in this paper that the traditional curried notation of CG with type shifting can be maintained to account for Surface Form↔PAS mapping without leading to proliferation of argument categories or to spurious ambiguity.

Categorial framework is particularly suited for this mapping due to its lexicalism. Grammatical functions of the nouns in the lexicon are assigned by case markers, which are also in the lexicon. Thus, grammatical function marking follows naturally from the general CCG schema comprising rules of application (A) and composition (B). The functor-argument distinction in CG helps to model prominence relations without extra levels of representation. CCG schema (Steedman (1988; 1990)) is summarized in (4). Combinator notation is preferred here because they are the formal primitives operating on the PAS (cf. (Curry and Feys, 1958) for Combinatory Logic). Application is the only primitive of the combinatory system; it is indicated by juxtaposition in the examples and denoted by · in the normal order evaluator (§4). B has the reduction rule $\mathsf{B}fga \geq f(ga)$.

(4)
$$\begin{array}{llll}
X/Y{:}\,f & Y{:}\,a & \Rightarrow_{\mathsf{A}>} & X{:}\,fa \\
Y{:}\,a & X\backslash Y{:}\,f & \Rightarrow_{\mathsf{A}<} & X{:}\,fa \\
X/Y{:}\,f & Y/Z{:}\,g & \Rightarrow_{\mathsf{B}>} & X/Z{:}\,\mathsf{B}fg \\
Y\backslash Z{:}\,g & X\backslash Y{:}\,f & \Rightarrow_{\mathsf{B}<} & X\backslash Z{:}\,\mathsf{B}fg \\
X/Y{:}\,f & Y\backslash Z{:}\,g & \Rightarrow_{\mathsf{B}_{\times}>} & X\backslash Z{:}\,\mathsf{B}fg \\
Y/Z{:}\,g & X\backslash Y{:}\,f & \Rightarrow_{\mathsf{B}_{\times}<} & X/Z{:}\,\mathsf{B}fg
\end{array}$$

2 Grammatical Functions, Type Shifting, and Composition

In order to derive all permutations of a ditransitive construction in Turkish using (3), the dative-marked indirect object (NP_3) must be type shifted in 48 (4!2) different ways so that coordination with the left-adjacent and the right-adjacent constituent is possible. This is due to the fact that the result category T is always a conjoinable type, and the argument category T/NP_3 (and $T\backslash NP_3$) must be allowed to compose with the result category of the adjacent functor. However, categories of arguments can be made more informative about grammatical functions and word order. The basic principle is as follows: The category assigned for argument n must contain all and only the term information about NP_i for all $i \leq n$. An NP_2 type must contain in its category word order information about NP_1 and NP_2 but not NP_3. This can be generalized as in (5):

(5) Category assignment for argument n:

$$C(n) = \begin{cases} T_r/T_a \text{ or } T_r\backslash T_a \\ NP_n \end{cases}$$

[2] aka. type raising, lifting, or type change

T_a = Lexical category of an NP_n-governing element (e.g., a verb) in the language whose highest genotype argument is NP_n.

T_r = The category obtained from T_a by removing NP_n.

Case markers in Turkish are suffixes attached to noun groups.[3] The types of case markers in the lexicon can be defined as:

(6) Lexical type assignment for the case marker (*-case*) encoding argument n:

$$\textit{-case} := \mathcal{C}(n) \colon \mathcal{T}(\mathcal{C}(n))x \backslash N \colon x$$

where $\mathcal{T}(C)$ denotes the semantic type for category C:

(7) a. $\mathcal{T}(NP_n) = \mathsf{I}$ (lower type for NP_n)

b. $\mathcal{T}(C) = \mathsf{T}$ (if C is a type shifted category as in (3))

c. $\mathcal{T}(C) = \mathsf{BBT}$ (if C is a type shifted and composed category)

(5) and (6) are schemas that yield three lexical categories per *-case*: one for lower type, and two for higher types which differ only in the directionality of the main function due to (5). For instance, for the accusative case suffix encoding NP_2, we have:

$$\begin{aligned} \text{-ACC} &:= NP_2 \colon \mathsf{I}x \backslash N \colon x \\ &:= ((S|NP_1)/(S|NP_1|NP_2)) \colon \mathsf{T}x \backslash N \colon x \\ &:= ((S|NP_1)\backslash(S|NP_1|NP_2)) \colon \mathsf{T}x \backslash N \colon x \end{aligned}$$

Type shifting alone is too constraining if the verbs take their arguments in an order different from the Applicative Hierarchy (§1). For instance, the category of Turkish ditransitives is $S|NP_1|NP_3|NP_2$. Thus the verb has the wrapping semantics $\mathsf{C}v'$ where C is the permutator with the reduction rule $\mathsf{C}fga \geq fag$. Type shifting an NP_3 yields $(S|NP_1|NP_2)/(S|NP_1|NP_2|NP_3)$ in which the argument category is not lexically licensed. (5) is order-preserving in a language-particular way; the result category always corresponds to a lexical category in the language if the argument category does too.

For arguments requiring a non-canonical order, we need type shifting and composition (hence the third clause in (7)):

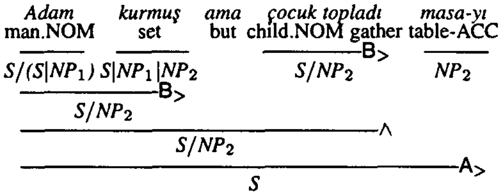
$$NP_3 \colon x \overset{\mathsf{T}}{\Rightarrow} (S|NP_1)/(S|NP_1|NP_3) \colon \mathsf{T}x \overset{\mathsf{B}}{\Rightarrow} (S|NP_1|NP_2)/(S|NP_1|NP_3|NP_2) \colon \mathsf{B}(\mathsf{T}x) = \mathsf{BBT}x$$

Once syntactic category of the argument is fixed, its semantics is uniquely determined by (7).

The combinatory primitives operating on the PAS are I (7a), T (7b–c), and B (7c). T has the reduction rule $\mathsf{T}af \geq fa$, and $\mathsf{I}f \geq f$. The use of T or B signifies that the term's category is a functor; its correct place in the PAS is yet to be determined. I indicates that the term is in the right place in the partially derived PAS.

According to (5), there is a unique result-argument combination for a higher type NP_3, compared to 24 using (3). (5) differs from (3) in another significant aspect: T_r and T_a may contain directionally underspecified categories if licensed by the lexicon. Directional underspecification is needed when arguments of a verb can scramble to either side of the verb. It is necessary in Turkish and Warlpiri but not in Japanese or Korean. The neutral slash | is a lexical operator; it is instantiated to either \ or / during parsing. A crucial use of underspecification is shown in (8). SV composition could not follow through if the verbs had backward-looking categories; composition of the type shifted subject and the verb in this case would only yield a backward-looking $S\backslash NP_2$ by the schema (4).

(8)



'The man had set the table but the child is cleaning it.'

The schema in (5) makes the arguments available in higher types, and allows lower (NP_n) types only if higher types fail (as in NP_2 in (8)). There are two reasons for this: Higher types carry more information about surface order of the language, and they are sufficient to cover bounded phenomena. §3 shows how higher types correctly derive the PAS in various word orders. Lower types are indispensable for unbounded constructions such as relativization and coordination. The choice is due to a concern for economy. If lower types were allowed freely, they would yield the correct PAS as well:

[3] As suggested in (Bozsahin and Gocmen, 1995), morphological and syntactic composition can be distinguished by associating several attachment calculi with functors and arguments (e.g., affixation, concatenation, clitics, etc.)

(9)

$$\begin{array}{lllll}
\text{S} & \text{IO} & \text{DO} & \text{V} & \\
\hline
NP_1{:}\,\mathsf{I}s' & NP_3{:}\,\mathsf{I}i' & NP_2{:}\,\mathsf{I}o' & DV{:}\,\mathsf{C}v' & \\
\cline{3-4}
 & & \multicolumn{2}{r}{S|NP_1|NP_3{:}\,(\mathsf{C}v')(\mathsf{I}o')} & \mathsf{A}_< \\
\cline{2-4}
 & \multicolumn{3}{r}{S|NP_1{:}\,(\mathsf{C}v')(\mathsf{I}o')(\mathsf{I}i')} & \mathsf{A}_< \\
\cline{1-4}
\multicolumn{4}{c}{S:(\mathsf{C}v')(\mathsf{I}o')(\mathsf{I}i')(\mathsf{I}s') \geq v'\,i'\,o'\,s'} & \mathsf{A}_<
\end{array}$$

In parsing this is achieved as follows: An NP_k can only be the argument in a rule of application, and schema (5) is the only way to obtain NP_k from a noun group. Thus it suffices to check in the application rules that if the argument category is NP_k, then the functor's result category (e.g., X in X/Y) has none of the terms with genotype indices lower than k. NP_2 in (8) is licensed because the adjacent functor is S/NP_2. NP_2 in (9) is not licensed because the adjacent functor has NP_1.

For noun-governed grammatical functions such as the genitive (NP_5), (5) licenses result categories that are underspecified with respect to the genotype index. This is indeed necessary because the resulting NP can be further inflected on case and assume a genotype index. For Turkish, the type shifted category is $\mathcal{C}(5)$ $=NP_{agr}/(NP_{agr}\backslash NP_5)$. Hence the genitive suffix bears the category $\mathcal{C}(5)\backslash N$. Agreement features enforce the possessor-possessed agreement on person and number via unification (as in UCG (Calder et al., 1988)):

$$\begin{array}{lllll}
\textit{kalem} & \textit{-in} & \textit{uc} & \textit{-u} & \\
\text{pencil} & \text{-GEN.3s} & \text{tip} & \text{-POSS.3s} & \\
\hline
N{:}\,p' & \mathcal{C}(5)\backslash N{:}\,\mathsf{T} & N{:}\,t' & (NP_{agr}\backslash NP_5)\backslash N{:}\,poss & \\
\cline{1-2}\cline{3-4}
\multicolumn{2}{c}{NP_{agr}/(NP_{agr}\backslash NP_5){:}\,\mathsf{T}p' \quad \mathsf{A}_<} & \multicolumn{2}{c}{NP_{agr}\backslash NP_5{:}\,poss\,t'} & \mathsf{A}_< \\
\cline{1-4}
\multicolumn{4}{c}{NP_{agr}:\mathsf{T}p'\,(poss\,t')\geq(poss\,t')p'} & \mathsf{A}_> \\
\multicolumn{4}{c}{\text{'The tip of the pencil'}} &
\end{array}$$

3 Word Order and Scrambling

Due to space limitations, the following abbreviated categories are employed in derivations:

IV = $S|NP_1$
TV = $S|NP_1|NP_2$
DV = $S|NP_1|NP_3|NP_2$

The categories licensed by (5) can then be written as IV/TV and $IV\backslash TV$ for NP_2, TV/DV and $TV\backslash DV$ for NP_3, etc. (10a–b) show the verb-final variations in the word order. The bracketings in the PAS and juxtaposition are left-associative; $(fa)b$ is same as fab.

(10) a.

$$\begin{array}{llll}
\textit{Mehmet} & \textit{kitab-ı} & \textit{oku-du} & \\
\text{M.NOM} & \text{book-ACC} & \text{read-PAST} & \\
\hline
S/IV{:}\,\mathsf{T}m' & IV/TV{:}\,\mathsf{T}b' & TV{:}\,r' & \\
\cline{2-3}
 & \multicolumn{2}{c}{IV{:}\,\mathsf{T}b'\,r'} & \mathsf{A}_> \\
\cline{1-3}
\multicolumn{3}{c}{S{:}\,\mathsf{T}m'\,(\mathsf{T}b'\,r')\geq r'\,b'\,m'} & \mathsf{A}_> \\
\multicolumn{3}{c}{\text{'Mehmet read the book.'}} &
\end{array}$$

b.

$$\begin{array}{llll}
\textit{kitab-ı} & \textit{Mehmet} & \textit{oku-du} & \\
\hline
IV/TV{:}\,\mathsf{T}b' & S\backslash IV{:}\,\mathsf{T}m' & TV{:}\,r' & \\
\cline{1-2}
\multicolumn{2}{l}{S/TV:\mathsf{B}(\mathsf{T}m')(\mathsf{T}b')} & & \mathsf{B}_{\times<} \\
\cline{1-3}
\multicolumn{3}{l}{S:\mathsf{B}(\mathsf{T}m')(\mathsf{T}b')r'\geq r'\,b'\,m'} & \mathsf{A}_>
\end{array}$$

(10a) exhibits spurious ambiguity. Forward composition of S/IV and IV/TV is possible, yielding exactly the same PAS. This problem is resolved by grammar rewriting in the sense proposed by Eisner[4] (1996). Grammar rewriting can be done using predictive combinators (Wittenburg, 1987), but they cannot handle crossing compositions that are essential to our method. Other normal form parsers, e.g. that of Hepple and Morrill (1989), have the same problem. All grammar rules in (4) in fact check the labels of the constituent categories, which show how the category is derived. The labels are as in (Eisner, 1996). -FC: Output of forward composition, of which forward crossing composition is a special case. -BC: Output of backward composition, of which backward crossing composition is a special case. -OT: Lexical or type shifted category. The goal is to block e.g., $X/Y\text{-FC}\ Y/Z\text{-}\{\text{FC},\text{BC},\text{OT}\} \Rightarrow_{\mathsf{B}_>} X/Z$ and $X/Y\text{-FC}\ Y\text{-}\{\text{FC},\text{BC},\text{OT}\} \Rightarrow_{\mathsf{A}_>} X$ in (10a). S/TV composition would have the label -FC, which cannot be an input to forward application. In (10b), the backward composition follows through since it has the category-label S/TV-BC, which the forward application rule does not block. We use Eisner's method to rewrite all rules in (4).

(11a–b) show the normal form parses for post-verbal scrambling, and (11c–d) for verb-medial cases.

[4] Eisner (1996, p.81) in fact suggested that the labeling system can be implemented in the grammar by templates, or in the processor by labeling the chart entries.

(11) a. *oku-du* *Mehmet* *kitab-ı*
read-PAST M.NOM book-ACC

$$\overline{TV\!: r'} \quad \overline{S/IV\!: \mathsf{T}m'} \quad \overline{IV\backslash TV\!: \mathsf{T}b'}$$
$$\overline{\qquad\qquad\qquad\qquad}^{\,\mathsf{B}_{\times>}}$$
$$S\backslash TV : \mathsf{B}(\mathsf{T}m')(\mathsf{T}b')$$
$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad}^{\,\mathsf{A}_{<}}$$
$$S : \mathsf{B}(\mathsf{T}m')(\mathsf{T}b')r' \geq r'\,b'\,m'$$

'Mehmet read the book.'

b. *oku-du* *kitab-ı* *Mehmet*

$$\overline{TV\!: r'} \quad \overline{IV\backslash TV\!: \mathsf{T}b'} \quad \overline{S\backslash IV\!: \mathsf{T}m'}$$
$$\overline{\qquad\qquad\qquad}^{\,\mathsf{A}_{<}}$$
$$IV : \mathsf{T}b'\,r'$$
$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad}^{\,\mathsf{A}_{<}}$$
$$S : \mathsf{T}m'\,(\mathsf{T}b'\,r') \geq r'\,b'\,m'$$

c. *kitab-ı* *oku-du* *Mehmet*

$$\overline{IV/TV\!: \mathsf{T}b'} \quad \overline{TV\!: r'} \quad \overline{S\backslash IV\!: \mathsf{T}m'}$$
$$\overline{\qquad\qquad\qquad}^{\,\mathsf{A}_{>}}$$
$$IV : \mathsf{T}b'\,r'$$
$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad}^{\,\mathsf{A}_{<}}$$
$$S : \mathsf{T}m'\,(\mathsf{T}b'\,r') \geq r'\,b'\,m'$$

d. *Mehmet* *oku-du* *kitab-ı*

$$\overline{S/IV\!: \mathsf{T}m'} \quad \overline{TV\!: r'} \quad \overline{IV\backslash TV\!: \mathsf{T}b'}$$
$$\overline{\qquad\qquad\qquad}^{\,\mathsf{A}_{<}}$$
$$IV : \mathsf{T}b'\,r'$$
$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad}^{\,\mathsf{A}_{>}}$$
$$S : \mathsf{T}m'\,(\mathsf{T}b'\,r') \geq r'\,b'\,m'$$

Controlled lexical redundancy of higher types, e.g., having both (and only) IV/TV and $IV\backslash TV$ licensed by the lexicon for an NP_2, does not lead to alternative derivations in (10–11). Assume that A/B $B\backslash C$, where A/B and $B\backslash C$ are categories produced by (5), gives a successful parse using the output $A\backslash C$. $A\backslash B$ $B\backslash C$ and $A\backslash B$ B/C are not composable types according to (4). The other possible configuration, A/B B/C, yields an A/C which looks for C in the other direction. Multiple derivations appear to be possible if there is an order-changing composition over C, such as C/C (e.g., a VP modifier IV/IV). (12) shows two possible configurations with a C on the right. (12b) is blocked by label check because A/C-FC $C \Rightarrow_{\mathsf{A}_{>}} A$ is not licensed by the grammar. If C were to the left, only (12a) would succeed. Similar reasoning can be used to show the uniqueness of derivation in other patterns of directions.

(12) a. $C/C \;\; A/B \;\; B\backslash C \;\; C$

$$\overline{\qquad\qquad}^{\,\mathsf{B}_{\times>}}$$
$$A\backslash C\text{-FC}$$
$$\overline{\qquad\qquad\qquad}^{\,\mathsf{B}_{\times<}}$$
$$A/C\text{-BC}$$
$$\overline{\qquad\qquad\qquad\qquad}^{\,\mathsf{A}_{>}}$$
$$A\text{-OT}$$

b. $C/C \, A/B \;\; B/C \;\; C$

$$\overline{\qquad\qquad}^{\,\mathsf{B}_{>}}$$
$$A/C\text{-FC}$$
$$\text{——***——}\,\mathsf{A}_{>}$$

Constrained type shifting avoids the problem with freely available categories in Eisner's normal form parsing scheme. However, some surface characteristics of the language, such as lack of case marking in certain constructions, puts the burden of type shifting on the processor (Bozsahin, 1997). Lower type arguments such as NP_2 pose a different kind of ambiguity problem. Although they are required in unbounded constructions, they may yield alternative derivations of local scrambling cases in a labelled CCG. For instance, when NP_2 is peripheral in a ditransitive construction and the verb can form a constituent with all the other arguments ($S\backslash NP_2$ or S/NP_2), the parser allows NP_2. This is unavoidable unless the parser is made aware of the local and nonlocal context. In other words, this method solves the spurious ambiguity problem between higher types, but not among higher and lower types. One can try to remedy this problem by making the availability of types dependent on some measures of prominence, e.g., allowing subjects only in higher types to account for subject-complement asymmetries. But, as pointed out by Eisner (1996, p.85), this is not spurious ambiguity in the technical sense, just multiple derivations due to alternative lexical category assignments. Eliminating ambiguity in such cases remains to be solved.

4 Revealing the PAS

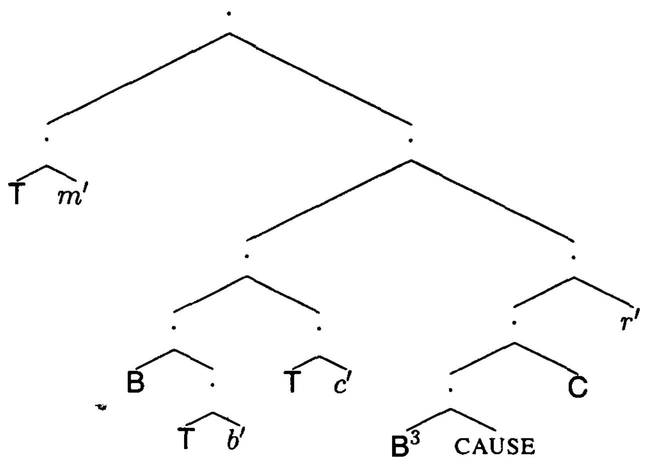
The output of the parser is a combinatory form. The combinators in this form may arise from the CCG schema, i.e., the compositor B, and the substitutor S (Steedman, 1987). They may also be projected from the PAS of a lexical item, such as the duplicator W (with the reduction rule $\mathsf{W}fa \geq faa$) for reflexives, and $\mathsf{B}^{n+1}\mathsf{C}$ for predicate composition with the causative suffix. For instance, the combinatory form for (13a) is the expression (13b).

(13) a. *Adam* *çocuğ-a* *kitab-ı*
man.NOM child-DAT book-ACC
$:m'$ $:c'$ $:b'$

oku-t-tu
read-CAUS-PAST
$:\mathsf{B}^3\textsc{caus}\mathsf{C}r'$

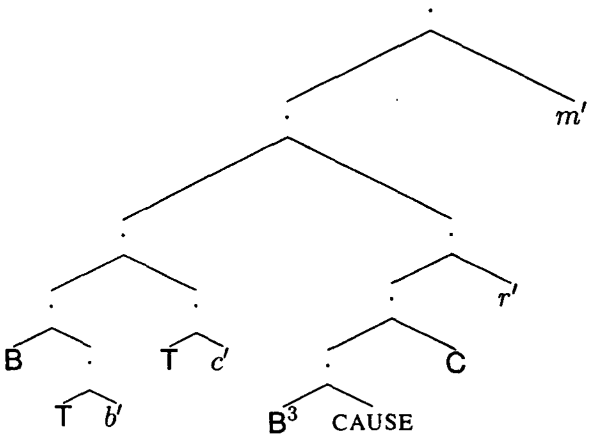
'The man had the child read the book.'

b. $\mathsf{T}{\cdot}m'\cdot(\mathsf{B}{\cdot}(\mathsf{T}{\cdot}b')\cdot(\mathsf{T}{\cdot}c')\cdot(\mathsf{B}^3{\cdot}\textsc{cause}{\cdot}\mathsf{C}{\cdot}r')) \equiv$



Although B works in a binary manner in CCG to achieve abstraction, it requires 3 arguments for full evaluation (its *order* is 3). Revealing the PAS amounts to stripping off all combinators from the combinatory form by evaluating the reducible expressions (redexes). $\mathsf{B}fg$ is not a redex but $\mathsf{B}fga$ is. In other words, the derivations by the parser must saturate the combinators in order to reveal the PAS, which should contain no combinators. PAS is the *semantic normal form* of a derivation.

The sequence of evaluation is the normal order, which corresponds to reducing the leftmost-outermost redex first (Peyton Jones, 1987). In tree-theoretic terms, this is depth-first reduction of the combinator tree in which the rearrangement is controlled by the reduction rule of the leftmost combinator, e.g., $\mathsf{T}m'\,X{\geq}Xm'$ where X is the parenthesized subexpression in (13b). Reduction by T yields:



Further reductions eventually reveal the PAS:

$$\mathsf{B}\cdot(\mathsf{T}\cdot b')\cdot(\mathsf{T}\cdot c')\cdot(\mathsf{B}^3\cdot\textsc{cause}\cdot\mathsf{C}\cdot r')\cdot m' \geq \quad (1)$$

$$\mathsf{T}\cdot b'\cdot(\mathsf{T}\cdot c'\cdot(\mathsf{B}^3\cdot\textsc{cause}\cdot\mathsf{C}\cdot r'))\cdot m' \geq \quad (2)$$

$$\mathsf{T}\cdot c'\cdot(\mathsf{B}^3\cdot\textsc{cause}\cdot\mathsf{C}\cdot r')\cdot b'\cdot m' \geq \quad (3)$$

$$\mathsf{B}^3\cdot\textsc{cause}\cdot\mathsf{C}\cdot r'\cdot c'\cdot b'\cdot m' \geq \quad (4)$$

$$\textsc{cause}\cdot(\mathsf{C}\cdot r'\cdot c'\cdot b')\cdot m' \geq \quad (5)$$

$$\textsc{cause}\cdot(r'\cdot b'\cdot c')\cdot m' \quad (6)$$

By the second Church-Rosser theorem, normal order evaluation will terminate if the combinatory form has a normal form. But Combinatory Logic has the same power as λ−calculus, and suffers from the same undecidability results. For instance, WWW has no normal form because the reductions will never terminate. Some terminating reductions, such as $\mathsf{CII}b{\geq}b\mathsf{I}$, has no normal form either. It is an open question as to whether such forms can be projected from a natural language lexicon. In an expression $X{\cdot}Y$ where X is not a redex, the evaluator recursively evaluates to reduce as much as possible because X may contain other redexes, as in (5) above. Recursion is terminated either by obtaining the normal form, as in (6) above, or by equivalence check. For instance, $(\mathsf{C}\cdot(\mathsf{I}\cdot a)\cdot b)\cdot Y$ recurses on the left subexpression to obtain $(\mathsf{C}\cdot a\cdot b)$ then gives up on this subexpression since the evaluator returns the same expression without further evaluation.

5 Conclusion

If an ordered representation of the PAS is assumed as many theories do nowadays, its derivation from the surface string requires that the category assignment for case cues be rich enough in word order and grammatical function information to correctly place the arguments in the PAS. This work shows that these categories and their types can be uniquely characterized in the lexicon and tightly controlled in parsing. Spurious ambiguity problem is kept under control by normal form parsing on the syntactic side with the use of labelled categories in the grammar. Thus, the PAS of a derivation can be determined uniquely even in the presence of type shifting. The same strategy can account for deriving the PAS in unbounded constructions and non-constituent coordination (Bozsahin, 1997).

Parser's output (the combinatory form) is reduced to a PAS by normal order evaluation. Model-theoretic interpretation can proceed in parallel with derivations, or as a post-evaluation stage which takes the PAS as input. Quantification and scrambling in free word order languages interact in many ways, and future work will concentrate on this aspect of semantics.

References

Alex Alsina. 1996. *The Role of Argument Structure in Grammar*. CSLI, Stanford, CA.

Cem Bozsahin and Elvan Gocmen. 1995. A categorial framework for composition in multiple linguistic domains. In *Proceedings of the Fourth International Conference on Cognitive Science of NLP*, Dublin.

Cem Bozsahin. 1997. Grammatical functions and word order in Combinatory Grammar. ms.

Joan Bresnan and Jonni M. Kanerva. 1989. Locative inversion in Chichewa: A case study of factorization in grammar. *Linguistic Inquiry*, 20:1–50.

Jonathan Calder, Ewan Klein, and Henk Zeevat. 1988. Unification categorial grammar. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest.

Haskell B. Curry and Robert Feys. 1958. *Combinatory Logic I*. North-Holland, Amsterdam.

David Dowty. 1988. Type raising, functional composition, and non-constituent conjunction. In Richard T. Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*. D. Reidel, Dordrecht.

Jason Eisner. 1996. Efficient normal-form parsing for combinatory categorial grammar. In *Proceedings of the 34th Annual Meeting of the ACL*, pages 79–86.

Jane Grimshaw. 1990. *Argument Structure*. MIT Press, Cambridge, MA.

Mark Hepple and Glyn Morrill. 1989. Parsing and derivational equivalence. In *Proceedings of the 4th EACL*, Manchester.

Beryl Hoffman. 1995. *The Computational Analysis of the Syntax and Interpretation of "Free" Word Order in Turkish*. Ph.D. thesis, University of Pennsylvania.

Lauri Karttunen. 1989. Radical lexicalism. In Mark Baltin and Anthony Kroch, editors, *Alternative Conceptions of Phrase Structure*. Chicago University Press.

Simon L. Peyton Jones. 1987. *The Implementation of Functional Programing Languages*. Prentice-Hall, New York.

Sebastian Shaumyan. 1987. *A Semiotic Theory of Language*. Indiana University Press.

Mark Steedman. 1987. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory*, 5:403–439.

Mark Steedman. 1988. Combinators and grammars. In Richard T. Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*. D. Reidel, Dordrecht.

Mark Steedman. 1990. Gapping as constituent coordination. *Linguistics and Philosophy*, 13:207–263.

Mark Steedman. 1996. *Surface Structure and Interpretation*. MIT Press, Cambridge, MA.

K. Vijay-Shanker and David J. Weir. 1993. Parsing some constrained grammar formalisms. *Computational Linguistics*, 19:591–636.

Stephen Wechsler. 1995. *The Semantic Basis of Argument Structure*. CSLI, Stanford, CA.

Kent Wittenburg. 1987. Predictive combinators. In *Proceedings of the 25th Annual Meeting of the ACL*, pages 73–79.