The Derivational Generative Power of Formal Systems or Scrambling is Beyond LCFRS*

Institute for Research in Cognitive Science, University of Pennsylvania

Technical Report IRCS-92-38

Tilman Becker[†], Owen Rambow[‡], Michael Niv[‡]

† Universität des Saarlandes
Graduiertenkolleg Informatik
Im Stadtwald
D-W6600 Saarbrücken 11
GERMANY
tilman@cs.uni-sb.de

‡ University of Pennsylvania
Dept. of Computer and Information Science
Philadelphia, PA 19104
USA
{rambow, niv}@linc.cis.upenn.edu

1 Introduction

Scrambling is a word-order phenomenon which is "doubly unbounded" in that more than one element can move, and movement can be unbounded. [Becker et al., 1991] argue that scrambling is beyond TAG by assuming that elementary trees express a complete predicate-argument structure. In this paper, we formalize this notion by defining the "derivational generative capacity" of a formal system as the generative capacity given certain restrictions on the derivation. Using this notion, we show that LCFRS cannot derive scrambling.

*A version of this paper was presented at the third Mathematics of Language Workshop in Austin, Texas, November 1992. Becker was supported by a fellowship of the "Graduiertenkolleg Informatik" which is funded by the Deutsche Forschungsgemeinschaft. Niv and Rambow were supported by DARPA N00014-90-J-1863, ARO DAAL03-89-C-0031, NSF IRI 90-16592, Ben Franklin 91S.3078C-1. We would like to thank Aravind Joshi and Yves Schabes for helpful discussions.

2 Linguistic Facts

Scrambling can be seen as the leftward movement of arguments (nominal, prepositional or clausal) in languages such as German, Korean, Japanese and Hindi. In this paper, we will give data only from German. The following example is from [Haider, 1991].

(1) ...daß [eine hiesige Firma] [meinem Onkel] [die Möbel] [vor drei Tagen]
...that a local company (NOM) my uncle (DAT) the furniture (ACC) three days ago
[ohne Voranmeldung] zugestellt **hat**
without advance warning delivered has
...that a local company delivered the furniture to my uncle three days ago without advance warning

As Haider points out, "any permutation of these five elements (5! = 120) is grammatically well-ordered". If we assume the base order is as given in (1), we conclude that scrambling of more than one element is possible.

If there are embedded clauses, scrambling can move elements out of the embedded clause ("long-distance" scrambling). (In German, scrambling can never proceed out of tensed clauses. This matter does not affect this analysis.)

(2) a. ...daß bisher noch niemand [PRO den Kühlschrank zu reparieren] versprochen **hat**
...that so far yet no-one (NOM) the refrigerator (ACC) to repair promised has
...that so far, no-one has promised to repair the refrigerator

b. ...daß [den Kühlschrank]$_i$ bisher noch niemand [t$_i$ zu reparieren] versprochen **hat**
...that the refrigerator (ACC) so far yet no-one (NOM) to repair promised has
...that so far, no-one has promised to repair the refrigerator

There is no bound on the number of clause boundaries over which an element can scramble:

(3) ...daß [den Kuehlschrank]$_i$ bisher noch niemand
...that the refrigerator (ACC) so far yet no-one (NOM)
[[t$_i$ zu reparieren] zu versuchen] versprochen hat.
to repair to try promised has
...that so far no-one has promised to try to repair the refrigerator.

Furthermore, an element scrambled (long-distance or not) from one clause does not preclude the scrambling of an element from another clause:

(4) ...daß [dem Kunden]$_i$ [den Kuehlschrank]$_j$ bisher noch niemand t$_i$ [[t$_j$ zu reparieren]
...that the client (DAT) the refrigerator (ACC) so far yet no-one (NOM) to repair

zu versuchen] versprochen hat.
to try promised has
...that so far no-one has promised the client to try to repair the refrigerator.

We thus conclude that scrambling in German is "doubly unbounded" in the following sense:

- There is no bound on the distance over which each element can scramble.
- There is no bound on the number of unbounded dependencies that can occur in one sentence.

A similar generalization appears to be licensed for other scrambling languages. For example, [Saito, 1992] exhibits a Japanese sentence in which two elements have undergone scrambling across a clause boundary (CP).

(Note that the "clause union" analysis of infinitival constructions that has been proposed in the literature [Evers, 1975] does not affect the argument that we present in this paper, since our analysis is based on predicate-argument structure, which must be derivable even under a "clause-union" analysis.)

3 LCFRS

Linear Context-Free Rewriting Systems (LCFRS) were introduced in [Weir, 1988] in order to provide a general framework for defining any so-called mildly context-sensitive formalism, such as Tree Adjoining Grammars (TAG; see Section 4.1).

The definition of an LCFRS comprises two parts: First, a generalized context-free grammar (GCFG) constructs a set of (context-free) derivation trees. Each context-free production rule is associated with a function symbol. Second, a unique yield function is associated with each production rule of the GCFG. The yield of a node is a tuple of strings. The yield function constructs the yield of the node from the yields of the nodes of the right-hand side along with a bounded collection of new terminals.

Formally, an LCFRS G is an ordered pair (C, E), where C is a GCFG and E a set of equations defining yield functions. The GCFG is a quadruple, $C = (V, S, F, P)$, where V is a set of variables (nonterminals), S a distinguished variable, P a set of productions of the form $A \rightarrow f(A_1, \ldots, A_n)$, and F a set of such function symbols. If k is the maximum number of components in the yield of any nonterminal of G (its "fanout"), we will say that G is an LCFRS-k grammar.

4 The Notion of Derivational Generative Capacity

4.1 Weak, Strong, and Derivational Generative Capacity

The issue of formal complexity of natural language has been discussed in terms of two formal properties: weak and strong generative capacity. [Shieber, 1985] and [Miller, 1991] argue that the set of sentences of Swiss German and Swedish, respectively, cannot be generated by a context-free grammar (CFG). The arguments rely on the weak generative capacity of the CFG. [Bresnan et al., 1983] and [Miller, 1991] argue that the set of syntactic structures of Dutch and Swedish, respectively, cannot be generated by a CFG and a Tree Adjoining Grammar (TAG), respectively. These arguments rely on the strong generative capacity of the systems in question. In this section, we introduce a third way of arguing about the formal adequacy of formal systems, the "derivational generative capacity" of a system.

The starting point for our definition is the observation that while the specifics of a syntactic analysis of a particular construction may be subject to controversy (such as the issue of verb raising in Dutch), the (semantic) predicate-argument structure of sentences is fairly uncontroversial. Furthermore, what characterizes many syntactic phenomena in natural languages is the way in which the predicate-argument structures map onto the surface strings. For example, a crucial difference between Dutch and German clausal embedding is the fact that Dutch exhibits cross-serial dependencies between verbs and subcategorized nouns, while the dependencies are nested in German. This statement holds independently of whether or not verb raising is assumed to occur in Dutch and/or in German. However, in the past such insight has not been exploited for the formal study of natural language, since most arguments discuss the adequacy of CFGs, and in a context-free framework the mapping between predicate-argument structure and syntactic structure cannot be expressed easily.

However, formalisms with different types of rules are now being explored for linguistic description. These include Tree Adjoining Grammars (TAG) (see [Joshi, 1987] for formal issues and [Kroch, 1987] for the linguistic relevance). In TAG, the elementary structures are trees. The domain of locality of the elementary structures is extended beyond that of the context-free rule. In a TAG, it is easy to localize predicate-argument relations within the elementary structures of the grammar. We will therefore assume that the elementary structures include exactly one lexical head and (positions for) its complements. (This restriction is called "co-occurrence constraint" in [Becker et al., 1991].) It follows that semantically related symbols are contributed by the same elementary structure in the grammar, i.e. they are generated in the same derivation step. Imposing predicate-argument relations among terminal symbols of a string therefore corresponds to imposing specific restrictions on the possible derivations of the string. If we consider the predicate-argument structure to be an essential part of the description of a natural language, then we may impose restrictions on the derivation structures in a particular system.

We define a system's "derivational generative capacity" in terms of its ability to generate sets of derivation structures. Notice that derivational generative capacity is orthogonal to strong generative capacity: it is possible that a system can generate a given set of strings with some given constraints on the derivation (and thus on the derivation structure), but not with some constraints on the derived structures, and vice versa. However, as in the case of strong generative capacity, if a grammar can derive a set of strings with some constraints on their derivation, then of course it can derive those strings without the constraints. In the following, we will present a formalization of constraints on derivations, which we will then use to represent predicate-argument relations in scrambled strings in German.

4.2 A Formalization of Derivational Constraints

Definition An "indexed string" over an alphabet T is a finite string w over T, along with a mapping from the symbols (i.e., symbol tokens) of w to the natural numbers. A string w that is equipped with such a mapping is written as w^{ind}, and the mapping is represented by superscripts in brackets on the symbols of w.

Definition Let w^{ind} be an indexed string and G a grammar in grammar formalism GF. We will say that G derives w^{ind} if there is a derivation of w in G (as defined by GF) in which all symbols bearing the same index are contributed in the same derivation step, and any derivation step contributes only symbols bearing the same index. This definition extends in the obvious way to sets of indexed strings.

Here are some examples:

1. In a CFG all symbols bearing the same index (and no others) must be generated by the same application of a single context-free rule.
2. In a TAG, all symbols bearing the same index (and no others) must be generated by adjunction (or substitution) of the same elementary tree.
3. In an Multi Component-TAG [Weir, 1988], all symbols bearing the same index (and no others) must be generated by multi-component adjunction of the same elementary tree set.
4. In an LCFRS, G derives the indexed string w^{ind} if symbols bearing the same index (and no others) are contributed during the application of the same production. Put differently, the yield function associated with a production rule p is a composition of the strings contributed by the symbols of the right-hand side of p and the symbols of w^{ind} corresponding to the derivation step (and no others).

We now can define a formal representation for a subset of scrambled German sentences as a set of indexed strings. We will assume that each verb has exactly one overt nominal argument.

Recall that the indices are not part of the terminal alphabet, which is simply $\{n, v\}$.

$$\text{SCR} = \{\sigma(n^{[0]}, \ldots, n^{[m]})v^{[0]} \ldots v^{[m]} | m \geq 0 \text{ and } \sigma \text{ a permutation}\}$$

The string language SCR is the language $\{n^m v^m | m > 0\}$. Each index in SCR pairs exactly one n and one v, reflecting the fact that the n is an argument of the v. This means that the elementary structures of the grammar should be clause-sized, i.e., contain a verb and (a position for) its argument, and nothing else.

5 The Proof

We will show by contradiction that there can be no LCFRS that can derive SCR. The idea of the proof is as follows: We assume that there is some LCFRS-k grammar G that derives SCR. Then we construct a new grammar G' that introduces k' subscripts into the terminal alphabet of G'. This allows us to interpret the derivational generative capacity in terms of weak generative capacity and from this point on, we only deal with string languages. From G' we derive by intersecting with a regular language and by applying a homomorphism a LCFRS-k grammar G''' that derives COUNT-k' $= \{a_1^n, \ldots, a_{k'}^n | n > 0\}$. Since we choose k' to be sufficiently larger than k, it is known that COUNT-k' cannot be derived by any LCFRS-k grammar. Thus we have a contradiction.

In detail:

1. Suppose there is an LCFRS-k grammar $G = (C, E)$ with $C = (V, S, F, P)$ that derives SCR. The alphabet T of G is $\{n, v\}$. Let $k' = 2k + 1$. We construct a new LCFRS-k grammar $G' = (C', E')$ over the alphabet $T' = \{n_0, \ldots, n_{k'-1}, v_0, \ldots, v_{k'-1}\}$. $C' = (V, S, F', P')$, where $F' = \{f_i^{(j_1,\ldots,j_l)} | f \in F \text{ and } 0 \leq i, j_1, \ldots, j_l < k'\}$ and

$$P' = \{A^j \longrightarrow f_i^{(j_1,\ldots,j_l)}(A_1^{j_1}, \ldots, A_l^{j_l}) | (A \longrightarrow f(A_1, \ldots, A_l)) \in P \text{ and } 0 \leq i, j, j_1, \ldots, j_l < k'\}.$$

For every equation $e \in E$ associated with $f \in F$, there are $k' * (k')^l = (k')^{l+1}$ equations $e_0^{(0,\ldots,0)}, \ldots, e_{k'-1}^{(k'-1,\ldots,k'-1)}$ in E', associated with $f_i^{(j_1,\ldots,j_l)}$, respectively, where $e_i^{(j_1,\ldots,j_l)}$ is the same as e (for $0 \leq i, j_1, \ldots, j_l < k'$), except that all terminal symbols introduced bear the subscript i. We have:

$$L(G') = \{\sigma(n_{i_0}, \ldots, n_{i_m})v_{i_0} \ldots v_{i_m} | m \geq 0, i_0, \ldots, i_m \in \{0, \ldots, k'-1\}, \text{ and } \sigma \text{ a permutation}\}$$

It is clear that in every string in $L(G')$, for any i, $0 \leq i \leq k'-1$, the number of instances of n_i is equal to the number of instances of v_i.

3. Let LCFRL-k be the class of languages derived by LCFRS-k grammars. Since LCFRL-k is closed under intersection with regular languages [Seki et al., 1991], there is an LCFRS-k

grammar G'' that derives the language that results from the intersection of $L(G')$ with the regular language R:

$$R = n_0^* \dots n_{k'-1}^* (v_0 \dots v_{k'-1})^*$$

We obtain:

$$L(G'') = \{n_0^j \dots n_{k'-1}^j (v_0 \dots v_{k'-1})^j | j > 0\}$$

4. Since the LCFRL-k are also closed under application of homomorphisms [Seki et al., 1991], there is an LCFRS-k grammar G''' that derives the language that results from the application of the following homomorphism to L(G'') (ϵ denotes the empty string):

$$h(a) = \begin{cases} \epsilon & \text{if } a = v_i, 0 \leq i < k' \\ n_i & \text{if } a = n_i, 0 \leq i < k' \end{cases}$$

The resulting language is

$$L(G''') = \{n_0^j \dots n_{k'-1}^j | j > 0\}$$

which is COUNT-k$'$. The pumping lemma for LCFRS [Seki et al., 1991] can then be used to show that COUNT-k$'$ is not in LCFRL-k. So we have derived a contradiction.

Therefore no LCFRS can derive the Scrambling Language SCR.

6 Related Work

[Becker et al., 1991] show that scrambling is beyond TAG, and that it can be handled by two formalisms related to TAG. The argument is implicitly based on the notion of derivational generative capacity introduced here. [Radzinski, 1991] presents an argument based on weak generative capacity that the Chinese number system is beyond LCFRS. However, it is not clear that the linguistic phenomenon he considers is actually a syntactic one.

References

[Becker et al., 1991] Becker, T., Joshi, A., and Rambow, O. (1991). Long distance scrambling and tree adjoining grammars. In *Fifth Conference of the European Chapter of the Association for Computational Linguistics (EACL'91)*, pages 21–26. ACL.

[Bresnan et al., 1983] Bresnan, J., Kaplan, R., Peters, S., and Zaenen, A. (1983). Cross-serial dependencies in Dutch. *Linguistic Inquiry*, 13:613–635.

[Evers, 1975] Evers, A. (1975). *The transformational cycle in Dutch and German*. PhD thesis, University of Utrecht. Distributed by the Indiana University Linguistics Club.

[Haider, 1991] Haider, H. (1991). Argument structure: Semantic basis and syntactic effects. Class notes from the Third European Summer School in Language, Logic and Information, Saarbrücken.

[Joshi, 1987] Joshi, A. K. (1987). An introduction to Tree Adjoining Grammars. In Manaster-Ramer, A., editor, *Mathematics of Language*, pages 87–115. John Benjamins, Amsterdam.

[Kroch, 1987] Kroch, A. (1987). Subjacency in a tree adjoining grammar. In Manaster-Ramer, A., editor, *Mathematics of Language*, pages 143–172. John Benjamins, Amsterdam.

[Miller, 1991] Miller, P. (1991). Scandinavian extraction phenomena revisited: Weak and strong generative capacity. *Linguistics and Philosophy*, 14.

[Radzinski, 1991] Radzinski, D. (1991). Chinese number-names, tree adjoining languages, and mild context-sensitivity. *Computational Linguistics*, 17(3):277–300.

[Saito, 1992] Saito, M. (1992). Long-distance scrambling in Japanese. *Journal of East Asian Linguistics*, 1(1):69–118.

[Seki et al., 1991] Seki, H., Matsumura, T., Fujii, M., and Kasami, T. (1991). On multiple context-free grammars. *Theoretical Computer Science*, 88:191–229.

[Shieber, 1985] Shieber, S. B. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343.

[Weir, 1988] Weir, D. J. (1988). *Characterizing Mildly Context-Sensitive Grammar Formalisms*. PhD thesis, University of Pennsylvania.