WORD-ORDER RELAXATIONS & RESTRICTIONS WITHIN A DEPENDENCY GRAMMAR

Martin Plátek Tomáš Holan
Vladislav Kuboň
Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
`platek@ksi.mff.cuni.cz`, `holan@ksvi.mff.cuni.cz`
`vk@ufal.mff.cuni.cz`

Karel Oliva
OeFAI
Schottengasse 3
A-1010, Wien, Austria,
`karel@ai.univie.ac.at`

Abstract

This paper presents some new results on relaxations and restrictions of word-order within dependency grammar (DG). The notions of dependency and word-order are separated in order to obtain an infinite scale of classes of gradually relaxed languages, starting with the context-free class. A linguistically motivated type of grammars, the *proper DGs*, is defined. At the end, the paper discusses the relevance of degree of word-order relaxation for parsing complexity.

1 Introduction and basic notions

This paper is a substantially shortened version of the technical report [3], where all the details (incl. motivations, formal background, proofs, etc.) are to be found. This technical report is a continuation of papers [1], [2] (linguistic issues) and [4] (formal considerations).

The notion of word-order relaxation within a dependency grammar (DG) means that besides the usual (projective) interpretations of a dependency grammar other (non-projective) interpretations are also considered. A possible approach is put forward in the following definitions.

Definitions. *A dependency grammar (DG)* is a tuple $G = (T, N, S_t, P)$, where T is the set of terminals, N is the set of nonterminals, $V = N \cup T$, $S_t \subseteq N$ is the set of starting symbols, and P is the set of rewriting rules in the following forms:

a) $A \rightarrow_X BC$, where $A \in N$, $B, C \in V$, $X \in \{L, R\}$ b) $A \rightarrow B$, where $A \in N$, $B \in V$.

The letter L (R) in the subscripts of the rules of the type a) means that the first (second) symbol on the right-hand side of the rule is considered *dominant*, and the other *dependent*.

If a rule has only one symbol on its right-hand side, we consider this symbol to be *dominant*.

For a reduction, a rule is applied as follows: the dependent symbol (if any) is deleted and the dominant one is rewritten by the symbol occurring on the left-hand side. The rules $A \rightarrow_L BC$, $A \rightarrow_R BC$ can be applied for a reduction of a string z for any of the occurrences of symbols B, C in z, where B precedes C in z (not necessarily immediately).

The reduction history is recorded in a *DR-tree* (delete-rewrite-tree). For a grammar G and a string w, this tree is obtained by interpreting the rules of the grammar as local trees (trees of depth one) from which the DR-tree is then combined, cf. Fig. 1. The direction of the edge connecting the node with its mother reflects the nature of the daughter: if the daughter is dominant, the edge is vertical, if it is dependent, the edge is oblique.

The notion of DR-tree Tr over a string can be understood also as a derivation of a dependency tree (D-tree) $dT(Tr)$. Such a D-tree is achieved by collapsing each strictly vertical path (sequence of vertical edges) into a single node marked by the terminal symbol from the bottom of this path, and by keeping the oblique edges intact (this means that all edges in any D-tree are oblique) For a clarifying example, cf. Fig.1. Note that both kinds of trees can contain crossing edges. Note also that the number of crossings in a D-tree must be less than or equal to the number of crossings in the respective DR-tree; in fact, it is even possible to have a DR-tree with crossing branches inducing a D-tree without any crossing.

Example 1. Let $G = (T_1, N_1, \{A_1\}, P_1)$, $T_1 = \{a_1, a_2, b_1, b_2\}$, $N_1 = \{A_1, A_2, B_1, B_2\}$, $P_1 = \{A_1 \to_R a_1 B_1, B_1 \to_L b_1 A_2,\ A_2 \to_R a_2 B_2, B_2 \to_L b_2 A_1 | b_2\}$.
The left part of Fig.1. displays a DR-tree Tr parsed by G for the input sentence $a_1 a_1 a_2 a_2 b_1 b_2 b_1 b_2$.

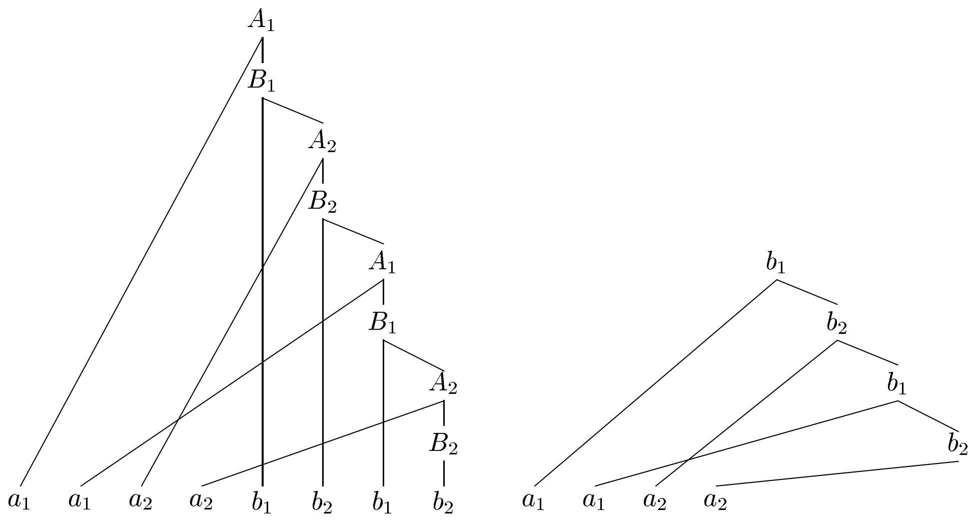

Figure 1: DR-, D-tree on $a_1 a_1 a_2 a_2 b_1 b_2 b_1 b_2$

The right part of Fig. 1. displays the D-tree dTr contracted from Tr.

Definitions. For the purpose of definition of coverage, let us first associate each terminal symbol within a string with a number marking its position counted from left, and call this number the *horizontal index* of the terminal symbol.

For any node u of a tree Tr, we shall define its *coverage Cov(u,Tr)* as the set of horizontal indices of all terminal nodes from which a bottom-up path leads to u.

Let there be a node u of a DR-tree (D-tree) Tr such that $Cov(u,Tr) = \{i_1, i_2, \ldots, i_n\}$, $i_1 < i_2 \ldots i_{n-1} < i_n$, $1 \le j < n$ and $i_{j+1} - i_j > 1$. We say that the pair (i_j, i_{j+1}) forms a gap in the $Cov(u,Tr)$ (or that the $Cov(u,Tr)$ contains the gap (i_j, i_{j+1})).

Let Tr be a DR-tree (D-tree), u be a node of Tr, and $Cov(u,Tr)$ its coverage. The symbol *DR-Ng*(u,Tr) (*D-Ng*(u,Tr)) represents the number of gaps in $Cov(u,Tr)$.
DR-Ng(Tr) (*D-Ng*(Tr)) denotes the maximum from {DR-Ng(u,Tr); $u \in Tr$} ({D-Ng(u,Tr); $u \in Tr$}). We say that *DR-Ng*(Tr) (*D-Ng*(Tr)) is the *node-gaps complexity of* Tr.

A tree is projective if the number of gaps of any its node is equal to 0 (zero). If this is not so, the tree is non-projective. In order to measure non-projectivity we use the measures *DR-Ng* or *D-Ng*.

Example 2. We stick to the *DR-tree* Tr from the previous example. The coverages of non-terminal nodes are (in the top-down order) as follows: $\{1,2,3,4,5,6,7,8\}$, $\{2,3,4,5,6,7,8\}$,$\{2,3,4,6,7,8\}$ $\{2,4,6,7,8\}$, $\{2,4,7,8\}$, $\{4,7,8\}$, $\{4,8\}$, $\{8\}$. Hence *DR-Ng*$(Tr) = 2$, because the number of gaps in the coverages does not exceed 2, and, e.g., $\{2,4,6,7,8\}$ contains the gaps $(2,4), (4,6)$.

Observation. It holds that any D-tree which is contracted from a projective DR-tree is a projective D-tree, but there are certain types of non-projective DR-trees which are contracted into projective D-trees. Some of them are linguisticaly inadequate. This observation leads to the following definition.

Definition. A DG G is called a *properDG* if for any projective D-tree dTr generated by G there exists a projective DR-tree Tr generated by G such that $dTr = dT(Tr)$.

Example 3. Let us choose a DG G_{Ll} in the following way: $G_{Ll} = (T, N, \{C\}, P)$, where $T = \{a, b, c, d\}$, $N = \{A, B, C\}$, $P = \{A \rightarrow_L Bb, B \rightarrow_L Cc, C \rightarrow_L Aa, C \rightarrow d\}$.
We have shown in [4] that G_{Ll} is not a proper grammar. Namely it generates only projective (very simple) D-trees of depth 1, and on the other hand it generates a language, which is not context-free, thus it generates an infinite set of words corresponding to nonprojective DR-trees.

The previous definition embodies a substantial empirical claim, namely that all DGs of a natural language have to fall into the class proper grammars - i.e. the claim that grammars which lie outside this class are not linguistically adequate.

Second, as said above, we need mechanisms for expressing language-particular constraints on word-order, in particular constraints on number of gaps within a subtree (or local subtree) headed by word of a certain category. These constraints can be expressed easily as follows:

Definitions. Let $G = (T, N, P, S)$ be a DG, and Cs be a set of *gap restrictors*, i.e. pairs of the shape $[A, i]$, where $A \in N$ and $i \in Nat \cup \{0\}$. Then we say that the pair $G_{Cs} = (G, Cs)$ is a *restricted DG* (*RsD-grammar, RsDG*), and if G is a proper grammar we say that $G_{Cs} = (G, Cs)$ is a *restricted proper DG* (*prop-RsDG*). Any pair $[A, i] \in Cs$ expresses the constraint that only such DR-trees are well-formed according to the RsD-grammar G_{Cs} in which the value of the measure *DR-Ng* of any of their covering subtrees with the root-symbol A is less or equal to i.

Let $i \in (Nat \cup \{0\} \cup \{*\})$ and let us assume that $*$ is greater than any natural number. Then, for a (fixed) Cs and for a (fixed) string w we define the following:
DR-T(w, G_{Cs}, i) is the set of DR-trees generated by G over w such that the value of the measure *DR-Ng* does not exceed i for them, and at the same time the constraints from Cs are met for them (in the above sense). For $i = *$ only the constraints Cs are imposed on the set of DR-trees generated by G over w.
DR-L$(G_{Cs}, i) = \{w|$ *DR-T*$(w, G_{Cs}, i) \neq \emptyset\}$.
DR-$\mathcal{L}(i)$ denotes the class of languages *DR-L*(G_{Cs}, i), for all *RsDG*'s G_{Cs}.
DR-prop-$\mathcal{L}(i)$ denotes the class of languages *DR-L*(G_{Cs}, i), for all proper *RsDG*'s G_{Cs}.
For D-trees, the classes *D-*$\mathcal{L}(i)$ and *D-prop-*$\mathcal{L}(i)$ can be defined in a similar way.

Mind here the important difference in the nature of the two kinds of constraints. The first kind is a constraint which has to hold for the a tree globally (i.e. for all nodes of the tree). The gap restrictors are constraints which hold only for any (covering, induced) subtree of a node which is of certain category. We need to use both types of constrains in order to achieve the results on hierarchy.

Definition. Let CF^+ be the set of context free languages without empty string. Let us take $L \in CF^+$ and $k \in \{0\} \cup Nat$. We shall say that L has the *degree of DR-relax-ability k (DRL*$(L) = k$*)* if there exists a RsDG GS such that
a) *DR-L*$(GS, 0) = L$, and
b) *DR-L*$(GS, i) \notin$ *DR-*$\mathcal{L}(i-1)$, for $i \in \{1, 2, ..., k\}$, and *DR-L*$(GS, k) =$ *DR-L*$(GS, k + j)$ for any $j \in Nat$.
We shall also say that the grammar GS has the *degree of DR-relax-ability k (DRS*$(GS) = k$*).*

2 Results

Propositions. The following holds:

a) $CF^+ = DR\text{-}\mathcal{L}(0) = DR\text{-}prop\text{-}\mathcal{L}(0)$

b) For any $j \in Nat$ there exists a prop-RsDG Gp_j such that $j = DRS(Gp_j)$.

c) $DR\text{-}\mathcal{L}(0) \subset DR\text{-}\mathcal{L}(1) \subset ... \subset DR\text{-}\mathcal{L}(n)... \subset DR\text{-}\mathcal{L}(*)$

d) $DR\text{-}prop\text{-}\mathcal{L}(0) \subset DR\text{-}prop\text{-}\mathcal{L}(1) \subset ... \subset DR\text{-}prop\text{-}\mathcal{L}(n)... \subset DR\text{-}prop\text{-}\mathcal{L}(*)$.

The proposition b) was shown in [3]. The propositions c) and d) are considered as its consequences there. The proposition c) was shown independently already in [4] by using a sequence of improper grammmars similar to the grammar from Example 3.

The previous considerations are connected with parsing complexity through the concept of coverage. Some results concerning this topic are given in the following:

Proposition. Let us denote $Nat^+ = \{0\} \cup Nat$. To any RsDG GS there exists a (sequential) algorithm Am computing for any string w an $i \in Nat^+$, such that i is the smallest element of Nat^+ for which $w \in DR\text{-}L(GS, i)$, or, if such an i does not exist, returning a message about the fact that $w \notin L(GS, *)$. Moreover, for a given $i \in Nat^+$ Am recognizes the membership $w \in L(GS, i)$ in a polynomial time, where the degree of the polynomial increases with i.

Consequences. There exists a sequential algorithm such that for any $i \in Nat^+$ and any $L \in$ DR-$\mathcal{L}(i)$, the algorithm recognizes L in a polynomial time, where the degree of the polynomial increases with i. There exists a sequential algorithm recognizing any $L \in D\text{-}prop\text{-}\mathcal{L}(0)$ in a polynomial time.

Remark. We believe that there exists an $i \in Nat^+$ for which there does not exist an algorithm recognizing every language from $D\text{-}\mathcal{L}(i)$ in a polynomial time. We have even the suspicion, due to the results from [4], that this i can be equal to 0. Further we conjecture that there exists a sequential algorithm recognizing any $L \in D\text{-}prop\text{-}\mathcal{L}(i)$ for any natural i in a polynomial time. We will try to prove this in the future.

Acknowledgement

This work has been supported in part by the grant of GAČR No. 201/99/0236 and by the grant of Fonds zur Foerderung der wissenschaftlichen Forschung, No. P13224 to OeFAI. OeFAI is supported by the Austrian Federal Ministry of Education, Science and Culture.

References

[1] T. Holan, V.Kuboň, K.Oliva, M.Plátek. *Two Useful Measures of word-order Complexity.* In: Proceedings of the Coling '98 Workshop "Processing of Dependency-Based Grammars". A. Polguere and S. Kahane (eds.), University of Montreal, Montreal, 1998, pp. 21-28.

[2] T.Holan, V.Kuboň, K.Oliva, M.Plátek. *On Complexity of word-order.* In: Traitement automatique des langues (T.A.L.), Vol. 41, No 1, 2000, pp. 243-267.

[3] M.Plátek, T.Holan, K.Oliva, V.Kuboň. *Word-Order Relaxations and Restrictions within a Dependency Grammar.* Tech. Report TR-2001-02, MFF UK, Praha, 2001.

[4] M.Plátek, T.Holan, V.Kuboň. *On Relax-ability of Word-Order by D-grammars.* In: Combinatorics, Computability and Logic. C.S. Calude and M.J. Deneen (eds.), Springer Verlag, Berlin, 2001, pp. 159-174.