A D-TREE GRAMMARS ACCOUNT FOR ROMANIAN CASES OF FRONTING

ANCA DINU

Abstract. In this article we provide examples of D-Tree Grammars analysis for Romanian phrases which can not be correctly accounted for by plane TAG. We show that such cases are not isolated in Romanian: the case of questions with object fronting, multiple wh-words fronting and preposition phrase fronting (which actually ends in second position). We argue that DTG is a suitable framework for Romanian, both because they are linguistically well-motivated (they can be lexicalized, the elementary trees can be constructed based on linguistic evidence, the derivation tree is semantically relevant, etc) and because of their capability of accounting for diffcult Romanian syntactic construction of the type we present in this article.

1. INTRODUCTION

In this paper we give an account for some Romanian cases of extraction in the framework of D-Tree Grammars (DTG), a Tree Adjoining Grammars (TAG) related formalism. Systematic analysis of extraction within the TAG formalism are proposed in Kroch, Joshi (1986) and Kroch (1989) and included in the development of the XTAG project for English (XTAG Research group, 1995) and the FTAG project for French (Abeillè 1991, Abeillé 2001, Candito 1999). An alternative description of extractions in TAG is given in (Kahane *et al*., 2000).

The rest of the paper is structured as it follows. In section 2 we present the DTG formalism, its original motivation and give some linguistic examples. In section 3, some cases of Romanian extraction are discussed. We claim that in Romanian certain cases of extraction from wh-relative closes behave in a similar manner as their Kashmiri counterpart, which was a part of the original motivation for the introduction of DTG. We also provide linguistic examples of Romanian extraction of prepositional phrase, that are correctly analyzed by DTG, but cannot be analyzed by TAG. Section 4 is dedicated to the conclusions.

2. D-TREE GRAMMARS

D-Tree Grammars (DTG) are introduced in Rambow *et al.* (1995). An interesting related formalism, D-tree substitution grammar, is introduced in Owen

RRL, **LII**, *1–2*, p. 223–229, Bucureşti, 2007

Rambow *et al.* (2001). DTG are designed to overcome some limitations of TAG, while preserving its advantages, notably the lexicalization and the extended domain of locality (Joshi 1999): each elementary structure (i.e. tree structure) can be associated to a lexical item, whose properties (subcategorization, agreement, word order variation, etc.) can be locally stated within this structure. TAG have two problems that DTG overcome. The first one is that TAG treat the operations of modification and complementation in a heterogeneous manner. The modification (operation that adds a modifier, i.e an element which is not subcategorized by the head of the phrase) is handled by a special case of adjunction, where the adjoined tree is of depth 1 and the result is adding a leftmost or rightmost daughter to the node to which the *adjoin* operation is performed. The complementation (operation that adds a subcategorized argument to the syntactic phrase head) is handled both by substitution (in simple subategorization cases) and by the adjoining operation (in cases where parts of the relative clause have to remain above the main clause which is adjoined into its own clausal complement, like for example in Fig. 1, were we have a case of object fronting).

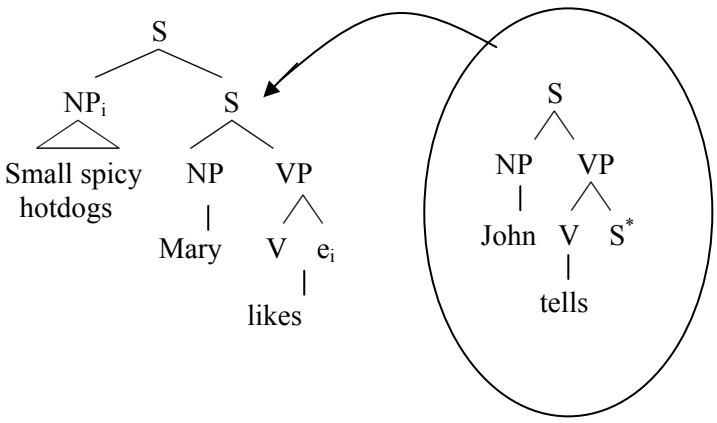

Fig. 1 – Obtaining the TAG derived tree for the proposition *Small spicy hotdogs John tells Mary likes.*

The second problem of TAG is a consequence of the first one. The use of substitution and adjunction in a linguistically heterogeneous manner implies that the directionality of the edges of derivation trees does not provide a good representation for the dependency structure of the phrase (i.e. the predicate-argument and modification structure). In Fig. 2 one sees that in the derivation tree the verb *tells* depends on the verb *likes*, instead of the other way around, as *likes* is a complement of *tells*. Also, the adjective small depends on the other adjective *spicy*, although *small* is a modifier of *hotdogs*. The DTG derivation tree for this phrase is instead semantically meaningful.

As we will see, DTG provide an elegant solution to both these problems.

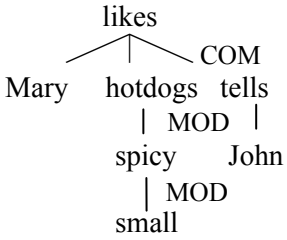

Fig. 2 – TAG derivation tree for the proposition *Small spicy hotdogs John tells Mary likes.*

There are also well known cases in which parts of the clausal complement are required to be placed within the structure of the adjoined tree, that TAG simply can not analyze, but the DTG are able to provide correct analysis for.

The idea behind the DTG is to design two operations that cleanly map to the complementation and modification operations, preserving the dominance relations, which exist between the adjoining tree constituents (nods) after the tree is adjoined. We now shortly remind the DTG formalism, as it was introduced in Rambow *et al.* (1995).

A *d-tree*[1] is a tree that has two types of edges: domination edges (d-edges) and immediate domination edges (i-edges).

During the derivation any number of nods can be inserted between two nods linked by a d-edge (preserving the dominance relation between them), whereas two nodes linked by an i-edge cannot be rescinded and remain in a mother-daughter relation throughout the derivation. D-edges and i-edges are not distributed arbitrarily in d-trees. For each internal node, either all of its daughters are linked by i-edges, or it has a single daughter attached to it by a d-edge. It follows that a d-tree containing *n* d-edges can be decomposed into *n + 1* components containing only i-edges. D-edges are represented by dashed lines and i-edges are represented by continuous lines.

A *DTG* is a construction G = (V_N, V_T, S, D), where V_N and V_T are a set of non-terminal symbols and a set of terminal symbols, respectively, with $V_N \cap V_T \neq \emptyset$, $S \in V_N$ is a distinguished initial symbol and *D* is a finite set of elementary d-trees.

The two operations that handel the complementation and modification in DTG are subsertion (*sub*stitution + in*sertion*) and sister-adjoining, respectively. When the d-tree α is subserted in the d-tree β, a component from the frontier of α is substituted at a node on the frontier of β and all other components of α (that are above this component) are inserted into d-edges of β, above the substitution site or above the root of β, thus respecting the dominance relation between the inserted components. Whenever a component $\alpha(i)$ of α is inserted into a d-edge of β between the nods η_1 and η_2, two new d-edges are created, one between η_1 and the root of $\alpha(i)$ and the other one between the node on the frontier of $\alpha(i)$ that was linked to a d-edge (that corresponds to the foot node in simple TAG terms) and η_2.

[1] *d* meens *dominance*

In another TAG related formalism called Multi-component TAG (where trees are grouped into sets which must be adjoined together) that was designed to extend the range of possible analysis, there is no way to state that two trees from a set must be in a dominance relation in the derived tree, even though the syntactic relations are invariably subject to c-command and dominance constraints. However, the MCTAG with Domination Links (Becker *et al.* 1991), which are systems that allow for the expression of dominance constraints, cannot be given linguistically meaningful interpretation of the derivation structures.

If a d-tree α is sister-adjoined in a node η of the d-tree β, the resulted d-tree γ is formed out of β in which α was added as the rightmost or leftmost daughter of η. An i-edge is created between η and α. It is possible to sister-adjoin more than one d-tree in a single node.

To avoid overgeneration, constraints on subsertion and sister-adjoining are needed.

Subsertion-adjoining trees (SA) are partial derivation structures that represent the dependency relation between the elementary structures. They record only the positions where substitution and sister-adjoining are done, but not the places where insertion is performed.

Let G be a DTG. One recursively defines the sets $T_i(G)$ as being sets of d-trees whose SA trees have the depth at most i. $T_0(G) = D$ (consists of all elementary d-trees). All elementary d-trees are marked as substitutable. Obviously the SA tree for a tree $\alpha \in T_0(G)$ is formed by a single node with the label α. $T_i(G) = T_{i-1}(G) \cup \{ \gamma \mid \gamma$ obtained by subsertion of sister-adjuncion of $\gamma_1, \gamma_2, ..., \gamma_k$ into α, $\alpha \in D$, $\gamma_1, \gamma_2, ..., \gamma_k \in T_{i-1}(G)\}$, where only the components marked as substitutable could have been substituted and only new components of γ that came from α are marked as substitutable in γ. It is done so, in order not to allow substitution (as part of the subsertion operation) of the same element more than once.

Let $\tau_1, \tau_2, \ldots \tau_k$ be the SA trees for the d-trees $\gamma_1, \gamma_2, ..., \gamma_k$. The SA tree τ for the d-tree γ has the root labeled by α; the root's daughters are $\tau_1, \tau_2, \ldots \tau_k$. There are two cases of labeling the edge between the root and its daughter τ_i.

If τ_i was subserted into α and α' is the root of τ_i, then only the components of α' were marked as substitutable in γ_i. It follows that $\exists\, j$ s.a. α' was substituted into α at a node n. The label of the edge between the root and its daughter τ_i is then (j, n).

If τ_i was sister-adjoined into α at a node n, then the label of the edge between the root and its daughter τ_i is (d, n), with $d \in \{left, right\}$.

The set of the d-trees generated by a DTG grammar G, denoted by $T(G)$ consists of the d-trees γ, obtained from the d-trees γ', $\gamma' \in T_i(G)$, $i \geq 0$, γ' has its root labelled by S and the frontier in V_T^*, by removing all the d-edges from γ'. A d-edge can be removed only if the labels of its endings are identical.

For a DT Grammar G, if the labels of two nodes linked by a d-edge are different in a derived d-tree, then this d-edge can not be removed and so the d-tree is not in $T(G)$.

The language generated by a DTG grammar *G*, denoted by *L(G)* is the set of strings on the frontier of the d-trees in *T(G)*.

3. ROMANIAN EXTRACTION CASES

Due to the free word order of Romanian, there are a number of phrases, which are rare, but grammatical, that cannot be analyzed by the TAG formalism. DTG prove to be in return a powerful enough tool for such an analysis. This comes at the expense of a greater computational complexity (compared to TAG), though the Early type parser for DTG introduced in Vijay-Shanker *et al.* (1995) works in polynomial time. The worst-case running time of this algorithm is $O(n^{4k+3})$, where n is the length of the input stream and k is the total number of d-edges in the grammar.

Part of the motivation for developing DTG was to get the word order right in cases where parts of the clausal complement are required to be placed within the structure of the adjoined tree. The example given in Rambow *et al.* (1995) is the analysis of a Kasmiri phrase, which cannot be produced by a plane TAG. It turns out that the same phrase in Romanian:

Ion ce crede că fac?

Ion what$_{ACC}$ believes that do$_{1st\ pers,sg}$?

What does Ion believe that I do?

can be analyzed in a similar manner and that TAG fail to analyze it. In Fig. 3 one can see the DTG analysis of this phrase.

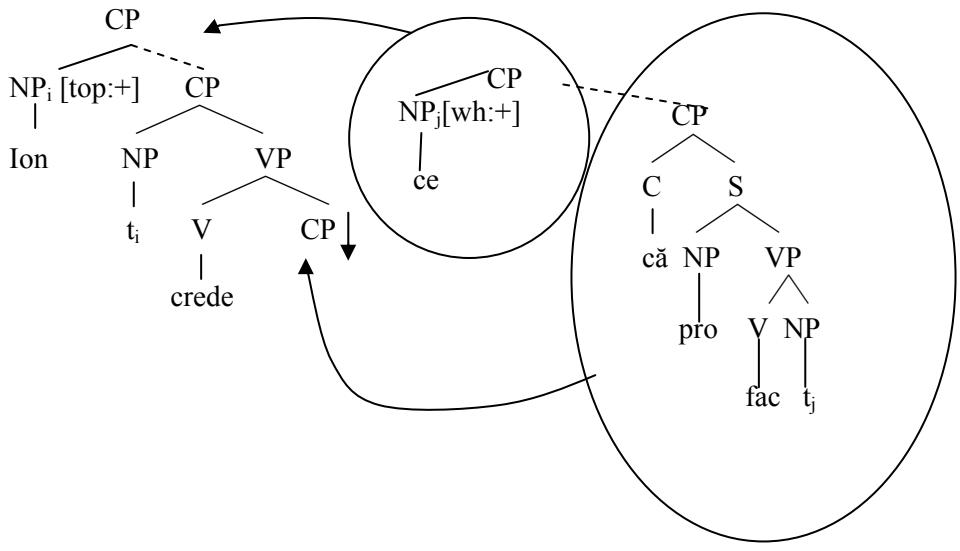

Fig. 3 – DTG derivation tree for the proposition *Ion ce crede că fac?*

In Leahu (1998) some other cases of phrases in Romanian which can not be analyzed by TAG are discussed, namely the multiple wh-fronting.

To show that such cases are not isolated in Romanian, we provide another examnple that DTG succeed and TAG fail to analyze. It is the case of preposition phrase fronting, which actually ends in second position (see Fig. 4), as in:

Ion în acest pat crede că doarme.
Ion in this bed believes that sleeps$_{3rd\ pers,\ sg}$.
It is in this bed Ion believes that he sleeps.

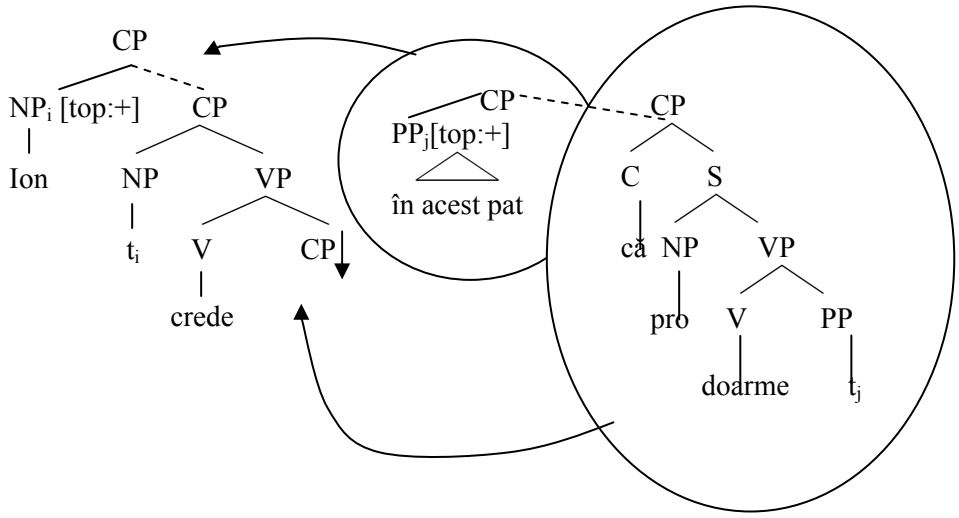

Fig. 4 – DTG derivation tree for the proposition *Ion în patul acesta crede că doarme.*

As it happens with all of the derivation trees in DTG, the derivation trees for the above sentences yield the correct representation for dependence relations, fact that provides a uniform interface to semantics.

4. CONCLUSIONS

We have provided examples of DTG analysis for Romanian phrases which cannot be accounted for by plane TAG. We showed that such cases are not isolated in Romanian: except of the Kashmiri case of questions with object fronting, which has a corresponding case in Romanian that behaves in a similar manner, there are other Romanian cases as multiple wh-fronting and preposition phrase fronting (which actually ends in second position) and probably more other cases. We argue that DTG is a suitable framework for Romanian, both because they are linguistically well-motivated (they can be lexicalized, the elementary trees can be

constructed exclusively based on linguistic evidence, the derivation tree is semantically relevant, etc.) and because of their capability of accounting for difficult syntactic construction of the type we presented in this article.

Acknowledgements: Research supported by MEdC-ANCT.

REFERENCES

Abeillé, A., 1991, *Une grammaire lexicalisée d'arbres adjoints pour le français*, Ph.D. Thesis Université Paris 7.

Abeillé, A., M. H. Candito, 2001, "FTAG: A lexicalized tree Adjoining Grammar for French", in A. Abeillé, O. Rambow, *Tree Adjoining Grammar: Formalisms, Linguistic Analysis and Processing*, Stanford, CSLI Publications, 305–29.

Becker, T., A. K. Joshi, O. Rambow, 1991, "Long Distance Scrambling and Tree Adjoining Grammars", in *EACL-91: Papers presented to the 5th Conference of the European Chapter of the Association for Computational Linguistics*, Berlin.

Candito, M. H., 1999, *Organisation modulaire et paramétrable de grammarires électroniques lexicalisées. Aplications au français et au l'italien,* Thése de Doctorat, Université Paris 7.

Sylvain, K., M-H, C., Y. de Kercadio, 2000, "An alternative descriptions of extractions in TAG", in *Proc. TAG+5*, Paris, 115–122.

Kroch, A., A. Joshy, 1986, "Analyzing extrapositions in a tree adjoining grammar", in: G. Huck, A. Ojeda (eds), *Syntax and Semantics: Discontinuous Constituents*, 107–149.

Rambow, O., K. Vijay-Shanker, D. Weir, 1995, "D-Tree Grammars", in *XTAG for English 1995, ACL 1995, Institute for Research in Cognitive Science*, University of Pennsylvania.

Rambow, O., D. Weir, K. Vijay-Shanker, 2001, "D-tree substitution grammars", in *Computational Linguistics*, 27, MIT Press Cambridge, MA, USA, 89–121.

The XTAG Research Group, 1995, "A lexicalized tree adjoining grammar for English", *Technical Report. IRCS Report 95-03*, University of Pennsylvania.

Joshi, A., 1999, "Explorations of a domain of locality: Lexicalized Tree-Adjoining Grammar", in *CLIN*, Utrecht.

Leahu, M., 1998, "Wh-dependencies in Romanian and TAG", in *Proceedings of Fourth International Workshop in TREE ADJOINING GRAMMAR*, University of Pennsylvania, Philadelphia, 92–96.

Vijay-Shanker, K., D. Weir, O. Rambow, 1995, "Parsing D-Tree Grammars", in *Proceedings of the International Workshop on Parsing Technologies*.