

Dependencies as Categories

Alexander Dikovsky

LINA, Université de Nantes

2 rue de la Houssinière

BP 92208 F 44322 Nantes Cedex 3,

France

Alexandre.Dikovsky@lina.univ-nantes.fr

Abstract

Categorial Dependency Grammars (CDG) introduced in this paper make clear-cut distinction between local and distant word driven dependencies. The former are treated in classical categorial grammar terms. The latter are defined in terms of polarized valencies, whose neutralization is controlled by the simple principle “choose the first available” (**FA**). Being very expressive, CDGs are analyzed in polynomial time. Besides this, CDGs represent a convenient frame for relating dependency grammar with linguistic semantics.

1 Introduction

Dependency grammars (DGs) are formal grammars assigning *dependency trees* (DTs) to well-formed sentences. A DT of a sentence is a labelled arrows tree whose nodes are the words of the sentence. A rather formal description of DGs and DG syntax was given by L. Tesnière (Tesnière, 1959). The first exact definitions are due to D. Hays (Hays, 1960) and H. Gaifman (Gaifman, 1961). The basic syntactic principle behind the DGs is quite different from that of syntagmatic grammars. They are designed for and more adapted to definitions of binary relations between wordforms (*syntactic dependencies*), than to definitions of sentence constituents. Meanwhile, historically the first DGs define in fact both. The Hays-Gaifman’s DGs are lexicalized. They assign grammatical categories to words and *position the subordinates with respect to their governors*. In this manner, they define not only the binary relations “governor \rightarrow subordinate”, whose union forms a tree, but also (due to the order given) the projections of words on the sentence, which form a system of constituents with the projected words serving as the constituents’ *heads*. From the 70ies, it is known ((Gladkij, 1966; Robinson, 1970)) that this link between the two

structures is reversible: a selection of one immediate head per constituent induces a unique DT by the following induction: $C \subsetneq C' \Rightarrow \text{root}(\text{ImmHead}(C')) \rightarrow^* \text{root}(\text{ImmHead}(C))$. This structural “equivalence” produced an illusion that DTs are *byproduct of head selection in constituent structures*. So all syntagmatic grammars with head selection (e.g., LFG (Kaplan and Bresnan, 1982) and HPSG (Pollard and Sag, 1994)) may in a way be considered as DGs. Formally, such “extension” to DG concerns TAGs (Joshi et al., 1975) and categorial grammars (CGs) (Bar-Hillel, 1953; Lambek, 1958)) as well. However, this analogy is very superficial. It was soon realized that *dependency* and *precedence* are rather independent. For instance, the head driven DT above are always *projective*: the projections of all words fill continuous segments. Meanwhile, discontinuous non-projective dependencies are inevitable in languages. They often mark communicative structure (e.g. topicalization) and special constructions encoding complex semantic relations (e.g. clefting, subject or object extraction in pied-piping, etc.). This distinction led to many propositions, both in terms of order constraints (e.g., (Maruyama, 1990; Bröker, 1998; Duchier and Debusmann, 2001)) and in structure sharing terms like lifting (e.g., (Lombardo and Lesmo, 1998; Kahane et al., 1998)). In logical type based grammars, like CGs, adequate extension to DG presupposes one more distinction: that of *semantic functionality* and of *syntactic subordinacy*, which are opposite, for instance, for verb and noun modifiers. Multimodal extensions of CGs take proper account of these distinctions (cf. (Moortgat and Morrill, 1991; Morrill, 1994; Kruijff, 2001)).

The overwhelming majority of DGs are *head driven* in the above sense. At the same time, some linguistic theories, e.g. “Meaning-Text Theory” (Mel’čuk, 1997), “Word Grammar” (Hudson, 1984) use a very different per-

spective of syntactic dependency, to which we will refer as *word driven*. Word-driven DG is closer to the original idea of Tesnière. Basically, it describes dependencies as binary relations between words in the language *without addressing head scopes*. This leads to far-reaching consequences for grammatical classification of words and dependency typology (cf. (Jackendoff, 1977; Keenan and Comrie, 1977) and a word driven DG of English in (Mel'čuk and Pertsov, 1987)). Formal definitions of word driven DGs are few in number. Historically the first was (Gladkij and Mel'čuk, 1971). It proposed a concept of tree generating DG and started a mathematical research into subfamilies of this very general class of grammars (see a review in (Dikovsky and Modina, 2000)). More recent propositions are Link Grammars (Sleator and Temperly, 1993) expressing only projective DTs and Polarized DGs (Dikovsky, 2001). The existing word driven DG definitions are operational and make no link with logic. In particular, there is no definition based on word driven dependency types. In this paper, we propose such a definition in terms of classical CGs. The Categorical Dependency Grammars (CDGs) we propose have two particularities: a specific type system inspired by (Mel'čuk and Pertsov, 1987) and a simple dependency calculus for projective and discontinuous dependencies.

2 Syntactic types

A word driven dependency $G(\text{overnor}) \xrightarrow{D} S(\text{ubordinate})$ encodes an irreflexive antisymmetric antitransitive binary relation on words with intuitive meaning “ G licenses S ” defined by constraints on lexical and grammatical features, precedence constraints, pronominalization constraints, etc. concerning the words G and S and sometimes also close context words (see (Iordanskaja and Mel'čuk, 2000) for a detailed presentation). Let us see a fragment of the definition in (Mel'čuk and Pertsov, 1987) of the *modificative* dependency in English:

$$\begin{array}{c} X_\alpha \\ \downarrow \\ \circ \\ Y_\beta \end{array} \text{modif}^* \Leftrightarrow Y_\beta + \dots + X_\alpha \quad \text{WHERE}$$

if $Y_\beta \prec Z \prec X_\alpha$ then

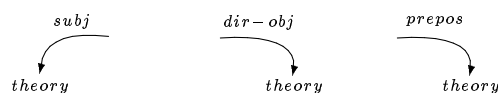
(if $Y_\beta \xrightarrow{D} Z$ then $Z = \text{'ENOUGH'}$ or
($D = \text{coordin}$ and $Z = \text{'OR'}$ | 'AND')))

otherwise if $X_\alpha \xrightarrow{(\text{det}|\text{quant})} Z$ then
exists $Y_\beta \prec U + Z : (U = \text{'SUCH'}$ | 'WHAT')
and $Z = \text{'A'}$ | 'AN') and
if $X_\alpha = (\text{Num})$ then
(forall) $T : \text{not } T \xrightarrow{\text{quant}} X_\alpha$
etc.)...
and $\alpha = (N, \text{not pron}), (\text{Num}), \text{'ONE'}$
and $\beta = (A, \text{not det}, \text{not pred!}, \text{not postpos!}),$
(V)_{ppres}, ($V, \text{not postpos!}$)_{pass}

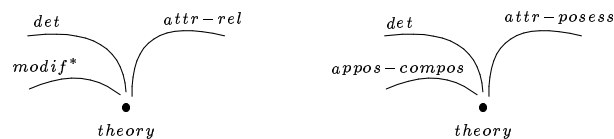
This complex formula is encoded by the dependency name *modif*. The corresponding dependency is the set of pairs of wordforms satisfying the formula in their occurrences.

Suppose that we have to describe syntactic type of the word *theory* in terms of such dependencies. Then we are to look at this word from two different points of view.

For *theory* as *dependent*, we must find all possible incoming dependencies:

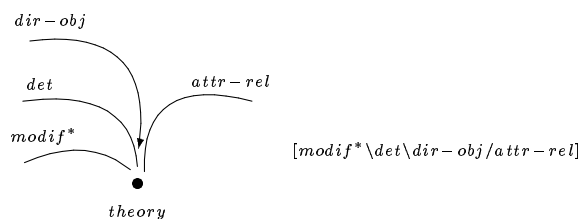


For *theory* as *governor*, we must find all possible dependencies outgoing from it:



In so doing, we must of course take into account the precedence of beginnings (ends) of the arrows with respect to the word. This we can do using the classical CG type constructors. The fundamental difference compared with CG is that now a primitive type D corresponds to the *incoming* dependency D and in complex types, e.g. $D \setminus \alpha$ assigned to a word w , D corresponds to *the beginning* of dependency D .

Finally, we must find all mutually compatible combinations of outgoing arrows and of an incoming arrow compatible with the features of the word *theory*. Each such combination describes a dependency type:



In this example, being assigned to *theory*, the type $[modif^*\backslash det\backslash dir - obj/attr - rel]$ admits several (possibly no) left dependents of *theory* through dependency *modif* (*modif** denotes iteration), requires a left dependent through *det*, a right dependent through *dir - obj* and the incoming dependency *attr - rel*. The position of the end of *attr - rel* must be determined through derivation. As we will see, such types are sufficient to describe projective dependencies. In order to specify the long distance discontinuous dependencies, we use, as we do in (Dikovsky, 2001), polarized dependency types. A *positive polarized type* specifies the name and the direction of an outgoing dependency. For instance, in the sentence *It was yesterday that we had the meeting*, the positive subcategory ($\nearrow it - cleft$) of the category assigned to *that* marks the beginning of the distant dependency named *it - cleft* outgoing from the clause root (see Fig. 3).

A *negative polarized type* specifies the name and the direction of an incoming dependency. For instance, the end of the discontinuous dependency *it - cleft* incoming to the expletive pronoun in the syntactic role of subject of the main sentence is defined by the negative category ($\swarrow it - cleft$) assigned to *It*. The categories ($\swarrow it - cleft$) and ($\nearrow it - cleft$) are *dual*. Together, they describe the discontinuous dependency *that* $\xrightarrow{it - cleft}$ *It*.

Now, let us define the dependency types formally.

For simplicity, in this paper we consider the primitive dependency types as elementary categories without parameters. \mathbf{C} denotes a finite set of such elementary categories. $\mathbf{C}^* =_{df} \{A^* \mid A \in \mathbf{C}\}$ denotes the set of all iterated categories.

Defining polarized categories, we distinguish between four *dependency polarities*: left and right positive \nearrow, \searrow and left and right negative \swarrow, \nwarrow . For each polarity $v \in \{\nearrow, \searrow, \swarrow, \nwarrow\}$ there is the unique *dual polarity* \check{v} : $\check{\nearrow} = \swarrow$, $\check{\searrow} = \nwarrow$, $\check{\swarrow} = \nearrow$, $\check{\nwarrow} = \searrow$. $\nearrow \mathbf{C}$, $\searrow \mathbf{C}$, $\swarrow \mathbf{C}$ and $\nwarrow \mathbf{C}$ denote the corresponding sets of polarized distant dependency categories. E.g., $\nearrow \mathbf{C} = \{\nearrow C \mid C \in \mathbf{C}\}$ is the set of *right positive* categories. $V^+(\mathbf{C}) = \nearrow \mathbf{C} \cup \searrow \mathbf{C}$ is the set of positive distant dependency categories, $V^-(\mathbf{C}) = \swarrow \mathbf{C} \cup \nwarrow \mathbf{C}$ is the set of those negative.

If we limit ourselves to these categories we cannot express adjacency of distant subordinates to a given word. E.g., in French, the negative de-

pendency category $\swarrow clitic - dobj$ of a cliticized direct object must be anchored to the auxiliary verb or to the verb in a non-analytic form. For that, we will use specially marked *anchored* negative categories: $Anc(\mathbf{C}) =_{df} \{\#(\alpha) \mid \alpha \in V^-(\mathbf{C})\}$ - our name for negative categories whose position is determined relative to some other category - whereas the negative categories in $V^-(\mathbf{C})$ will be called *loose*.

Definition 1 *The set $Cat(\mathbf{C})$ of categories is the least set verifying the conditions:*

1. $\mathbf{C} \cup V^-(\mathbf{C}) \cup Anc(\mathbf{C}) \subset Cat(\mathbf{C})$.
2. For $C \in Cat(\mathbf{C})$, $A_1 \in (\mathbf{C} \cup \mathbf{C}^* \cup Anc(\mathbf{C}) \cup \nwarrow \mathbf{C})$ and $A_2 \in (\mathbf{C} \cup \mathbf{C}^* \cup Anc(\mathbf{C}) \cup \nearrow \mathbf{C})$, the categories $[A_1 \backslash C]$ and $[C / A_2]$ also belong to $Cat(\mathbf{C})$.

By the nature of word driven dependencies, the type constructors $\backslash, /$ are associative. So every complex category can be presented in the form: $[L_k \backslash \dots L_1 \backslash C / R_1 \dots / R_m]$.

E.g., $[\#(\swarrow clitic - dobj) \backslash subj \backslash S / aux]$ is one of possible categories of an auxiliary verb, which defines it as the host word for a cliticized direct object, requires the local subject dependency on its left, and on its right, the local outgoing dependency *aux*.

3 Grammar definition

Definition 2 *A categorial dependency grammar (CDG) is a system $G = (W, \mathbf{C}, S, \delta)$, where W is a finite set of words, \mathbf{C} is a finite set of elementary categories containing the selected root category S , and δ - called lexicon - is an assignment of finite sets of categories in $Cat(\mathbf{C})$ to words in W (i.e. $\delta(a) \subset Cat(\mathbf{C})$ for $a \in W$).*

The language and DT language generated by a DCG are defined by a calculus of local and polarized dependencies. In this calculus, most specific are the rules for polarized dependencies. These rules establish a distant dependency between two words with dual polarized categories, if the corresponding negative category is *loose*. The anchored negative dependency valencies serve only to anchor a distant subordinate to a host word. As soon as the correct position of the subordinate is identified by the anchored dependency rule, its category becomes loose and so available to the governor. So the anchored dependency marker serves as an exponential in this resource sensitive calculus.

In the definition below, indexed Γ denote strings of categories. An occurrence of an el-

elementary category C in a derived string of categories corresponds to a DT D of category C . $r(D)$ denotes the root of D . For space reasons, we present only the rules for left constructors. The rules for right constructors are similar.

Definition 3 *Provability relation* \vdash

Local dependency rule:

L. $\Gamma_1 C[C \setminus \alpha] \Gamma_2 \vdash \Gamma_1 \alpha \Gamma_2$.

If C is the category of D_1 and $[C \setminus \alpha]$ is that of D_2 , then α becomes the category of the new DT

$$D_1 \cup D_2 \cup \{r(D_1) \xleftarrow{C} r(D_2)\}.$$

Iterated dependency rules:

I. $\Gamma_1 C[C^* \setminus \alpha] \Gamma_2 \vdash \Gamma_1 [C^* \setminus \alpha] \Gamma_2$.

If C is the category of D_1 and $[C^* \setminus \alpha]$ is that of D_2 , then $[C^* \setminus \alpha]$ in the consequence becomes the category of the new DT

$$D_1 \cup D_2 \cup \{r(D_1) \xleftarrow{C} r(D_2)\}.$$

Ω . $\Gamma_1 [C^* \setminus \alpha] \Gamma_2 \vdash \Gamma_1 \alpha \Gamma_2$ ¹.

Anchored dependency rule:

A. $\Gamma_1 \#(\alpha) [\#(\alpha) \setminus \beta] \Gamma_2 \vdash \Gamma_1 \alpha \beta \Gamma_2$, $\#(\alpha) \in \text{Anc}(\mathbf{C})$.

Distant dependency rule:

D. $\Gamma_1 (\sphericalangle C) \Gamma_2 [(\sphericalleftarrow C) \setminus \alpha] \Gamma_3 \vdash \Gamma_1 \Gamma_2 \alpha \Gamma_3$.

The rule applies if there are no occurrences of subcategories $\sphericalangle C$, $\#(\sphericalangle C)$ and $\sphericalleftarrow C$ in Γ_2 .

If $\sphericalangle C$ is the category of D_1 and $[(\sphericalleftarrow C) \setminus \alpha]$ is that of D_2 , then α becomes the category of the new DT

$$D_1 \cup D_2 \cup \{r(D_1) \xleftarrow{C} r(D_2)\}.$$

\vdash^* is the reflexive-transitive closure of \vdash .

Comments: 1. $\mathbf{L}^1, \mathbf{L}^r$ are standard elimination rules present in all kinds of categorial grammars. The rigid distribution constraints encoded by elementary word driven dependencies make impossible type raising and symmetrical introduction rules.

2. The iteration rules $\mathbf{I}^1, \mathbf{I}^r, \mathbf{\Omega}^1, \mathbf{\Omega}^r$ induce different realizations of the same iterated modifier type. In fact, the iteration marker also serves as an exponential in this calculus.

3. $\mathbf{D}^1, \mathbf{D}^r$ are rules establishing distant valencies between the ‘‘closest’’ dual loose dependency valencies. This is another important particularity of this calculus implied by the nature of word driven dependencies and the

underlying concept of polarity.

Definition 4 A DT D is assigned by a CDG $G = (W, \mathbf{C}, S, \delta)$ to a sentence w (denoted $G(D, w)$) if D is defined as DT of category S in a proof $\Gamma \vdash^* S$ for some $\Gamma \in \delta(w)$.

The DT-language generated by G is the set of DTs $\Delta(G) = \{D \mid \exists w \in W^+ G(D, w)\}$.

The language generated by G is the set of sentences $L(G) = \{w \in W^+ \mid \exists D G(D, w)\}$.

This definition is correct in the following sense:

Proposition 1 For each CDG G , $G(D, w)$ implies that D is a DT on w .

4 Expressivity

DCGs are more expressive than CFGs. Here is an example of a CDG generating a non-context-free language.

Example 1 Let $G_0 = (W_0, \mathbf{C}_0, S, \delta_0)$, with $\delta_0 : a \mapsto [\beta \setminus \alpha], [\alpha \setminus \alpha]$, $b \mapsto [\alpha_1 \setminus D/A]$, $d_1 \mapsto \alpha$, $d_2 \mapsto [\alpha \setminus \beta_1 \setminus S/D]$, $d_3 \mapsto D$, $c \mapsto [D \setminus A]$, where $\alpha = \#(\sphericalangle B)$, $\alpha_1 = (\sphericalleftarrow B)$, $\beta = \#(\sphericalangle C)$ and $\beta_1 = (\sphericalleftarrow C)$. For instance, Fig. 1 presents a derivation of $d_1 a^3 d_2 b^3 d_3 c^3$.

Proposition 2 $L(G_0) = \{d_1 a^n d_2 b^n d_3 c^n \mid n > 0\}$.

In Figures 1-3, two meeting continuous slanting lines correspond to one application of rules **L** or **I**, two meeting dashed slanting lines correspond to one application of rule **A**, and right-angled dashed lines connect categories to which rule **D** is applied. Anchored categories must first be made loose by applying the rule **A**. Only then a distant dependency can be introduced by application of the rule **D**. E.g., in Fig. 2, first **A** is applied to C_3 and C_4 , and then **D** applies to C_3 and C_7 .

Using CDGs - and this is their important advantage - one can describe discontinuous syntactic dependencies caused by topicalization, extraction, and other movement operations in a uniform and concise manner. E.g., let us see how can be described PP-movement (Fig. 2) and it-clefting (Fig. 3).

In the proof in Fig. 2, we use the categories $C_1 = [det \setminus subj/attr - rel] \in \delta(person)$, $C_2 = [\#(\sphericalangle prepos) \setminus attr - rel/wh - rel] \in \delta(whom)$, $C_3 = [subj \setminus wh - rel/inf - obj] \in \delta(must)$, $C_4 = [(\sphericalleftarrow prepos) \setminus inf - obj] \in \delta(refer)$, $C_5 = [subj \setminus S/n - copul] \in \delta(is)$.

¹The DTs rest unchanged when no instruction.

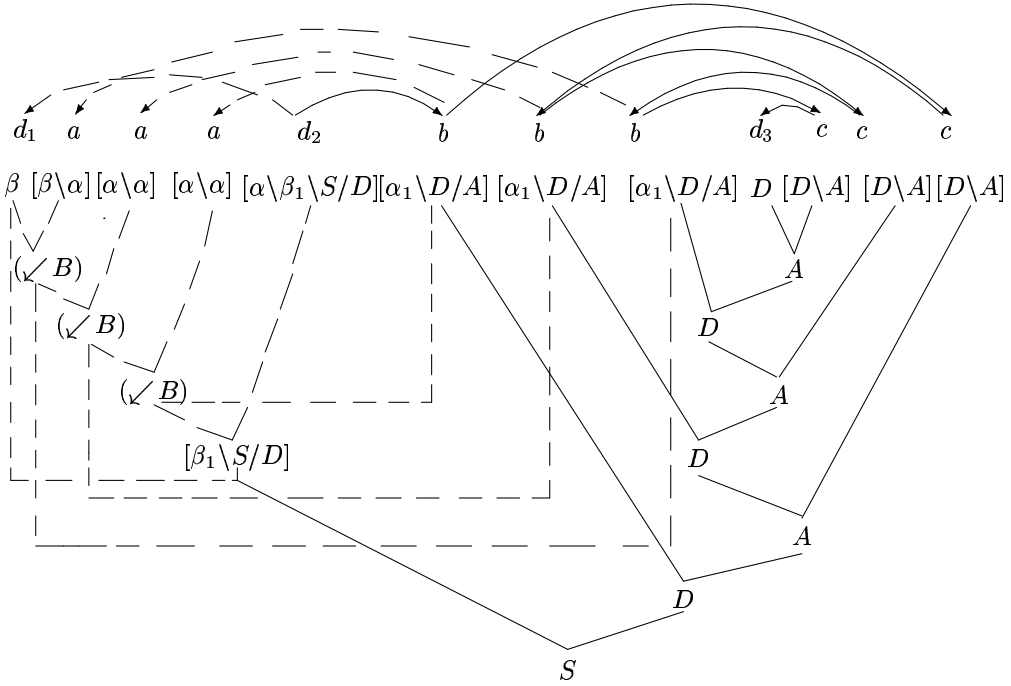


Fig. 1. A proof of $G_0(D, d_1 a^3 d_2 b^3 d_3 c^3)$.

In this proof, the rule **L** applied to *det* and C_1 gives the dependency *the* \xleftarrow{det} *person* of category $[subj/attr-rel]$. **L** applied to *subj* and C_3 gives the dependency *you* \xleftarrow{subj} *must* of category $[wh-rel/inf-obj]$. **A** applied to $\#(\surd prepos)$ and C_2 gives $(\surd prepos)[attr-rel/wh-rel]$. Now we can eliminate $(\surd prepos)$ applying **D** to it and C_4 . This reduces C_4 to *inf-obj* and gives the distant dependency *to* $\xleftarrow{wh-rel}$ *refer*. Finally, **L** is applied four times.

The proof in Fig. 3 uses categories: $C'_1 = [\#(\surd it-cleft) \setminus S/subj-cleft/circ] \in \delta(was)$, $C'_2 = [(\surd it-cleft) \setminus subj-cleft/th-rel] \in \delta(that)$, $C'_3 = [subj \setminus th-rel/obj] \in \delta(had)$, $C'_4 = [det \setminus obj] \in \delta(meeting)$.

The weak expressive power of CDGs is not completely explored. Meanwhile, some observations can already be made.

Let $G_{(m)}$, $m \geq 1$, be the CGD defined by:

$d_0 \mapsto [S/D_0]$ and $d_{m+1} \mapsto D_m$,
 $a_0 \mapsto [D_0/D_0/(\surd A_m)/\dots/(\surd A_1)]$,
for $0 < i \leq m$, $d_i \mapsto [D_{i-1}/\#(\surd A_i)]$,
and $a_i \mapsto [\#(\surd A_i)/\#(\surd A_i)], [\#(\surd A_i)/D_i]$.

Then the following proposition holds:

Proposition 3 For each $m > 0$, $L(G_{(m)}) = \{d_0 a_0^n d_1 a_1^n \dots d_m a_m^n d_{m+1} \mid n \geq 0\}$.

This means that not all languages in $\mathcal{L}(CDG)$ are generated by TAGs. On the other hand, we suppose that the copy language $\{ww \mid w \in \{a, b\}^+\}$ is not generated by CDGs.

The CDGs are sensitive to the following structural dependency measure.

Definition 5 Let D be a DT of a sentence $w = a_1 \dots a_n$. For a space i between the words a_i and a_{i+1} , $1 \leq i < n$, we define the distant dependencies thickness in i (denoted $dth(D, i)$) as the number of distant dependencies $(a_k \xleftarrow{d} a_l)$, $(a_k \xrightarrow{d} a_l)$ in D covering i (i.e. such that $k < i < l$ for some k, l and d). $dth(D) =_{df} \max\{dth(D, i) \mid 1 \leq i < |D|\}$ and $dth(G) =_{df} \max\{0, \min\{dth(D) \mid G(D, w)\} \mid w \in L(G)\}$.

E.g., $dth(G_0) = \infty$ (same for all $G_{(m)}$, $m \geq 1$). For natural languages, this measure is seemingly bounded by a small constant (2 or 3). In both

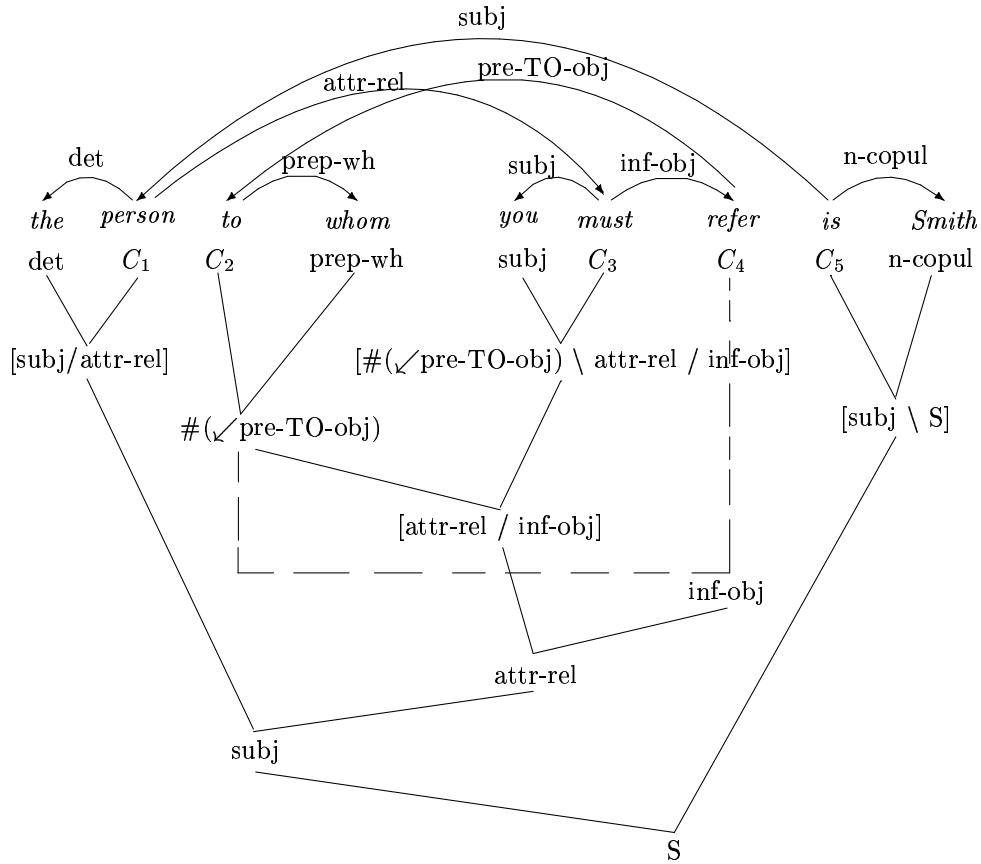


Fig. 2. Example of PP-movement.

examples above, $dth(D) = 1$. Meanwhile, the following theorem is a consequence of Theorem 2 in (Dikovskiy, 2001).

Theorem 1 *If for a CDG G , the measure $dth(G)$ is bounded by a constant, then $L(G)$ is context-free.*

This result confirms once more that relevant is the strong and not the weak expressive power.

5 Complexity

It turns out that correctness of sentences with respect to a CDG can be expressed in terms of two *independent* tests: the first in terms of only local (projective) dependencies, and the second in terms of neutralizability of distant dual polarized dependencies. This fact ensures efficient parsing algorithms.

In this short paper we don't describe the notions and technical details underlying efficient

analysis of CDGs (see (Dekhtyar and Dikovskiy, 2004)) and only announce several facts.

Theorem 2 (i) *There is an algorithm **pars** which parses CDGs in time $\mathbf{O}(n^5)$.*
(ii) *If a CDG G has bounded distant dependencies thickness $dth(G) < const$, then the algorithm **pars** parses G in time $\mathbf{O}(n^3)$.*

The analysis algorithm is Earley-type with items including current state of counters controlling well-pairing of loose polarized dependencies. When no polarized dependencies are used in the grammar or if their thickness is bounded, the items are counter-less (or bounded-counter).

Corollary 1 *There is an algorithm **pars**^p which parses projective CDGs in time $\mathbf{O}(n^3)$.*

6 Conclusion

The Categorical Dependency Grammars introduced in this paper can be used in practice as

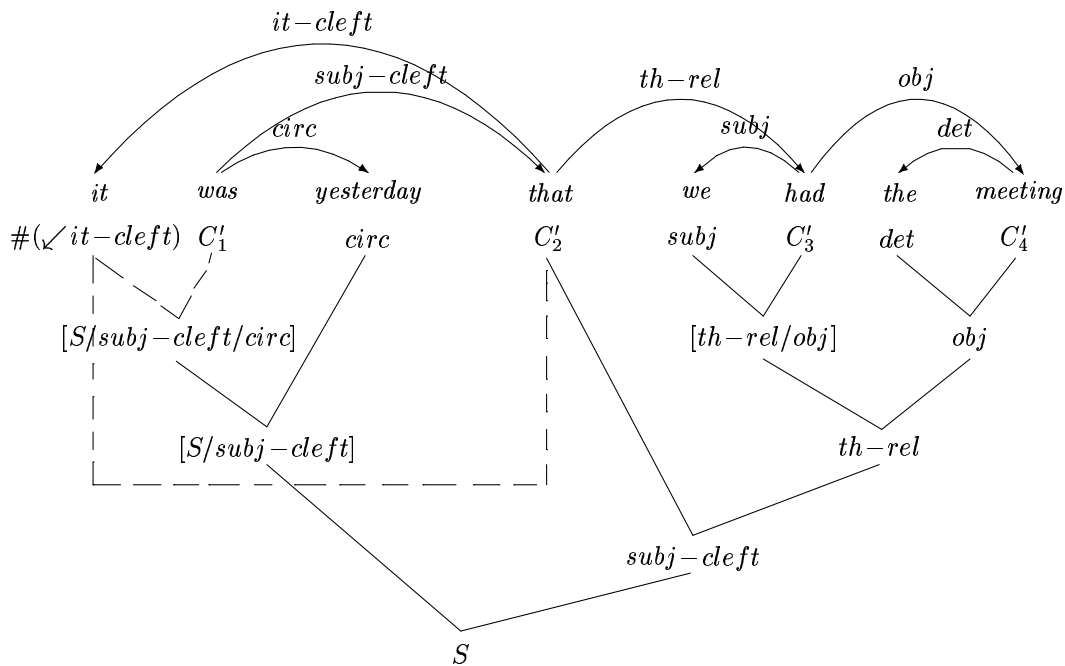


Fig. 3. Example of it-clefting.

low level DGs into which should be compiled for parsing high level word driven DGs. They can be easily extended to grammars using bounded depth feature structures as primitive categories and feature unification and propagation through dependencies. Using anchored categories, it is possible to express a variety of linear order constraints. At the same time, the CDGs have efficient parsing algorithms, comparable with or more fast than those for other dependency grammars expressing unlimited distant dependencies (cf. (Lombardo and Lesmo, 1998; Neuhaus and Bröker, 1997; Kahane et al., 1998)). A very important advantage of CDGs is their type-driven style definition which fits well the standard methods of constructing formal semantics.

7 Acknowledgments

The author is grateful to Denis Béchet, Annie Foret and Michael Dekhtyar for helpful comments and fruitful discussions of this work.

References

- Yehoshua Bar-Hillel. 1953. A quasi-arithmetical notation for syntactic description. 29(1):47–58.
- Norbert Bröker. 1998. Separating surface order and syntactic relations in a dependency grammar. In *Proc. COLING-ACL*, pages 174–180, Montreal.
- Michael Dekhtyar and Alexander Dikovsky. 2004. Categorical dependency grammars. In *Proc. of Intern. Conf. Categorical Grammars (CG2004)*, pages 76–91, Montpellier, France.
- Alexander Dikovsky and Larissa Modina. 2000. Dependencies on the other side of the curtain. *Traitement Automatique des Langues (TAL)*, 41(1):79–111.
- Alexander Dikovsky. 2001. Grammars for local and long dependencies. In *Proc. of the 39th Intern. Conf. ACL’2001*, pages 156–163. ACL & Morgan Kaufman.
- Denis Duchier and Ralf Debusmann. 2001. Topological dependency trees: A constraint-based account of linear precedence. In *Proc. of the 39th Intern. Conf. ACL’2001*, pages 180–187. ACL & Morgan Kaufman.
- Haïm Gaifman. 1961. Dependency systems and phrase structure systems. Report p-2315, RAND Corp. Santa Monica (CA). Published in: *Information and Control*, 1965, v. 8, n 3, pp. 304–337.
- Alexej V. Gladkij and Igor A. Mel’čuk. 1971. Grammatiki derev’ev. I. Opyt formalizacii preobrazovanij sintaksičeskix struktur

- estestvennogo jazyka [Tree grammars. I. A formalism for syntactic transformations in natural languages]. In *Informacionnye voprosy semiotiki, lingvistiki i avtomatičeskogo perevoda [Informational Problems of Semiotics, Linguistics and Automatic Translation]*, volume 1, pages 16–41. Published In "Linguistics. An International Review", n. 150. April 15, 1975, The Hague : Mouton, pp. 47-82.
- Alexej V. Gladkij. 1966. *Lekcii po Matematičeskoj Lingvistike dlja Studentov NGU [Course of Mathematical Linguistics. Novosibirsk State University (Russ.)]. (French transl. Leçons de linguistique mathématique. fasc. 1, 1970, Dunod).* Novosibirsk State University.
- David Hays. 1960. Grouping and dependency theories. Research memorandum RM-2646, The RAND Corporation. Published in: Proc. of the National Symp. on Machine Translation, Englewood Cliffs (N.Y.), 1961, pp. 258-266.
- Richard Hudson. 1984. *Word Grammar*. Basil Blackwell, Oxford-New York.
- Lydija Iordanskaja and Igor Mel'čuk. 2000. The notion of surface-syntactic relation revisited. valence-controlled surface-syntactic relations in french. In L. Iomdin and L. Krysin, editors, *Slovo v tekste i v slovare [Word in Text and in Dictionary]*, pages 391–433. Languages of Russian Culture, Moscow.
- Ray Jackendoff. 1977. *X-bar Syntax. A Study of Phrase Structure*. MIT Press, Cambridge.
- A.K. Joshi, L.S. Levy, and M. Takahashi. 1975. Tree adjunct grammars. *Journ. of Comput. and Syst. Sci.*, 10(1):136–163.
- Sylvain Kahane, Alexis Nasr, and Owen Rambow. 1998. Pseudo-projectivity : A polynomially parsable non-projective dependency grammar. In *Proc. COLING-ACL*, pages 646–652, Montreal.
- Ronald Kaplan and Joan Bresnan. 1982. Lexical functional grammar : A formal system for grammatical representation. In Bresnan J.W., editor, *The Mental Representation of Grammatical Relations*, pages 173–281. MIT Press, Cambridge, Massachusetts.
- Edward Keenan and Bernard Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8(1):63–99.
- Geert-Jan M. Kruijff. 2001. *A Categorical-Modal Logical Architecture of Informativity. Dependency Grammar Logic & Information Structure*. Ph.D. thesis, Charles University, Prague.
- Joachim Lambek. 1958. The mathematics of sentence structure. *American Mathematical Monthly*, pages 154–170.
- Vincenzo Lombardo and Leonardo Lesmo. 1998. Formal aspects and parsing issues of dependency theory. In *Proc. COLING-ACL*, pages 787–793, Montreal.
- Hiroshi Maruyama. 1990. Structural disambiguation with constraint propagation. In *Proc. of 28th ACL Annual Meeting*, pages 31–38.
- I. Mel'čuk and N.V. Pertsov. 1987. *Surface Syntax of English. A Formal Model Within the Meaning-Text Framework*. John Benjamins Publishing Company, Amsterdam/Philadelphia.
- Igor Mel'čuk. 1997. Vers une linguistique «Sens-Texte» . Leçon inaugurale au Collège de France.
- Michael Moortgat and Glin V. Morrill. 1991. Heads and phrases. Type calculus for dependency and constituent structure. Ms OTS, Utrecht.
- Glin V. Morrill. 1994. *Type Logical Grammar. Categorical Logic of Signs*. Kluwer Academic Publishers, Dordrecht.
- Peter Neuhaus and Norbert Bröker. 1997. The Complexity of Recognition of Linguistically Adequate Dependency Grammars. In *Proc. of 35th ACL Annual Meeting and 8th Conf. of the ECACL*, pages 337–343.
- Carl Pollard and Ivan Sag. 1994. *Head-driven Phrase Structure Grammar*. CSLI, Stanford, California.
- Jane J. Robinson. 1970. Dependency structures and transformational rules. 46(2):259–285.
- Daniel Sleator and Davy Temperly. 1993. Parsing English with a Link Grammar. In *Proc. IWPT'93*, pages 277–291.
- Lucien Tesnière. 1959. *Éléments de syntaxe structurale*. Librairie C. Klincksiek, Paris.