Polarized Non-projective Dependency Grammars

Alexander Dikovsky

Université de Nantes, IRIN, 2, rue de la Houssinière BP 92208 F 44322 Nantes cedex 3 France
Email: Alexandre.Dikovsky@irin.univ-nantes.fr
http://www.sciences.univ-nantes.fr/info/perso/permanents/dikovsky/

Abstract. *Dependency tree grammars* are proposed in which unbounded discontinuity is resolved through the first available valency saturation. In general, they are expressive enough to generate non-semilinear context sensitive languages, but in the practical situation where the number of non saturated valencies is bounded by a constant, they are weakly equivalent to cf-grammars, are parsable in cubic time, and are stronger than non-projective dependency grammars without long dependencies.

1 Introduction

Dependency based theories of surface syntax are well suited for treating discontinuity, which corresponds in dependency terms to non-projectivity. In contrast to phrase structure, the word order is separated from dependency structure. So one should describe syntactic relations between words explicitly in terms of *precedence* and *dependency* or its transitive closure : *dominance*. eing more flexible, the dependency syntax allows one to express some properties of syntactic structure which are hardly ever expressible in terms of phrase structure. The *projectivity* is one of such properties. It requires that any word occurring between a word g and a word d dependent on g were dominated by g. In first dependency grammars [4] and in some more recent grammars : link grammars [19], projective dependency grammars [10] the projectivity is implied by definition. In some other theories, e.g. in word grammar [6], it is included into the axioms defining acceptable surface structures. In presence of this property, D-trees are in a sense equivalent to phrase structures with head selection. For this reason, D-trees determined by grammars of Robinson [18], categorial grammars [1], and some other formalisms are projective. Projectivity affects the complexity of parsing: as a rule, it allows dynamic programming techniques which lead to polynomial time algorithms (cf. Earley-type algorithm for link grammars in [19]). Meanwhile, till the early 80ies, the search is continued of formalisms treating various non-projective constructions such as pronoun or WH-word extraction in Romance or Germanic languages, interrogative sentences, relative clauses, topicalization, paired conjunctions or prepositions, discontinuous negation, French pronominal clitics, etc. In a sense, non-projectivity and parsing efficiency are mutually exclusive. It is not surprising then, that to find a formalism which would be acceptable from both expressivity and effectiveness points of view is a difficult problem.

There are various formal dependency based approaches to this problem. Most liberal are the dependency grammars in the framework of Meaning-Text Theory [12], where dependencies are defined by independent rules, whose applicability to a pair of word-forms is determined by local constraints. The cost of this liberty is theoretically non-tractable parsing (the NP-hardness argument of Neuhaus and röker [17] applies to them). More recent versions of dependency grammars (see e.g.[8, 11, 2]) impose on non-projective D-trees some constraints weaker than projectivity (cf. meta-projectivity [16] or pseudo-projectivity [8]), sufficient for existence of a polynomial time parsing algorithm. Still another approach is developed in the context of intuitionistic resource-dependent logics, where D-trees are constructed from derivations (cf. e.g. a method in [9] for Lambek calculus). In this context, non-projective D-trees are determined with the use of hypothetical reasoning, structural rules such as commutativity and associativity, and multi-modality (see e.g. [14, 15, 5]).

An approach to non-projective dependencies proposed in this paper is combining the polarity idea of resource-dependent logics and a tree composition mechanism of dependency tree grammars [13, 3], which brings the benefit of effective parsing. The *dependency tree (DT-) grammars* we define here are analyzing in the same sense as categorial grammars: they reduce phrases to types and simultaneously compose the corresponding dependency structures. These structures (we call them *DV-structures*) are discontinuous and polarized in the sense that some their nodes may have valencies (positive for potential governors and negative for potential dependents). Saturating a pair of corresponding valencies introduces a *long dependency*. We impose on dependencies no projectivity-like constraints. Instead, we use a simple discipline of saturating valencies, so called first-available (FA) saturation and we keep track of valencies non-saturated in a DV-structure in its *integral valency*. "First available" means "closest not yet saturated". This natural discipline was already used in the literature (cf. [20]). The polarities are controlled in such a way that a cycle-free DV-structure becomes a D-tree if its integral valency is empty. So a DT-grammar determines D-trees FA-saturating DV-structures of reducible strings.

The first result of this paper is that when the integral valency is bounded by a constant and the FA-saturation causes no cycles, DT-grammars are weakly equivalent to cf-grammars and have a $\boldsymbol{O}(n^3)$-time parsing algorithm. The second result is that they are stronger than non-projective dependency grammars without long dependencies. The weak equivalence to cf-grammars doesn't prevent to express structural discontinuity in terms of dependencies. In presence of FA-saturation, long dependencies represent such phenomena as unbounded raising and extraction. ounded integral valency doesn't mean that the nesting of discontinuous structures is bounded. It means that before nesting a new discontinuous construction the valencies of the current one should be saturated. In Fig. 1(a) we show a sentence with integral valency 2 (extraction from extracted phrase), which is maybe most complex in French. It contrasts with lifting to a host word in the case of French pronominal clitics (see D-tree (b) in Fig. 1), which is described by a local non-projective dependency.

The paper is organized as follows. Dependency structures and their properties are described in Section 2, dependency grammars and main results are in Section 3. All proofs are moved to the Appendix.

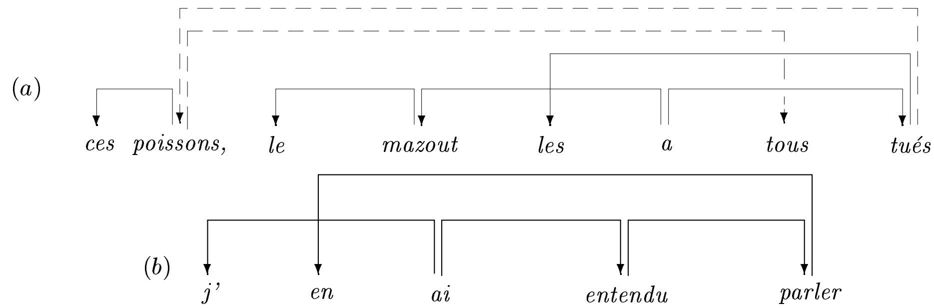


Fig. 1.

2 Dependency structures

We fix alphabets W of *terminals* (words), C of *nonterminals* (syntactic types or classes), and N of *dependency names*. For a string $x \in (W \cup C)^+$, we denote by $\mu(x)$ the list of occurrences of symbols in x.

Definition 1 *Let $x \in (W \cup C)^+$ be a string. A set $\pi = \{d_1, ..., d_n\}$ of trees covering $\mu(x)$ (called* components *of π), which have no nodes in common and whose arcs are labeled by names in N, is a* dependency (D-) structure *on x if one component d_t of π is selected as its* head [1]*. We use the notation $x = w(\pi)$. If x belongs to W^+, π is said to be* terminal*. When π has only one component, it is a* dependency (D-) tree *on x.*

Composition of D-structures is defined as follows.

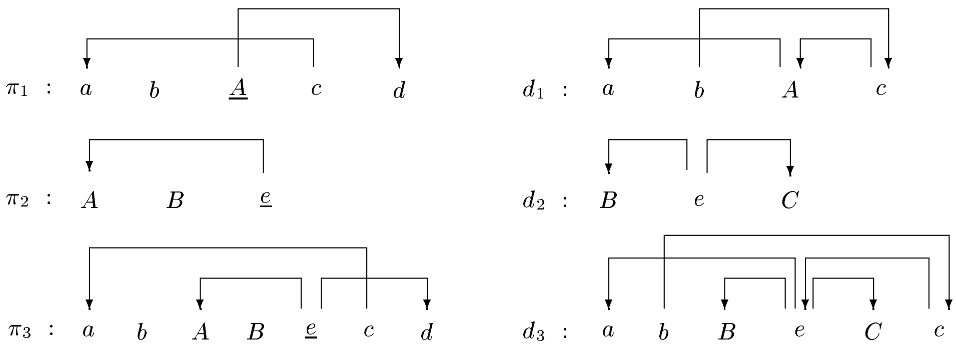
Definition 2 *Let $\pi_1 = \{d_1, ..., d_k\}$ be a D-structure, n be its node, $w(\pi_1) = w_1 n w_2$, and $\pi_2 = \{d'_1, ..., \underline{d'_t}, ..., d'_l\}$ be a D-structure with the head component d'_t. Then the result of the composition of π_2 into π_1 in n is the D-structure $\pi_1[n\backslash\pi_2]$, in which π_2 is substituted for n (i.e. $w(\pi_1[n\backslash\pi_2]) = w_1 w(\pi_2) w_2$), the root of d'_t inherits all dependencies of n in π_1, and the head component is that of π_1 (changed respectively if touched on by composition)* [2].

Composition of D-trees is a special case of this definition. For example, in Fig. 2, $\pi_3 = \pi_1[A\backslash\pi_2]$ and $d_3 = d_1[A\backslash d_2]$.

We will consider *long* the dependencies implicitly specified by *valencies* of nodes in D-structures. A valency indicates that a dependency r is expected at this node with some sign and in some direction. For example, the intuitive sense of a positive valency $+R : r$ of a node n is that a long dependency r would go from n somewhere on its right. Respectively, a negative valency $-L : r$ of m means that a long dependency r would enter m from somewhere on its left. Assigning to terminal nodes of D-structures their valencies, we obtain a new kind of D-structures: *DV-structures*. The explicit dependencies of DV-structures are considered *local*.

[1] We visualize d_t underlining it or its root, when there are some other components.

[2] This composition generalizes the substitution used in TAGs [7] and is not like the adjunction.



π_1 and π_2 are respectively 3- and 2-component D-structures (e.g., in π_1, the head component is the D-tree with the root A and dependent d), whereas π_3 is a four-component D-structure; d_1, d_2 and d_3 are D-trees (d_2 is projective, d_1, d_3 are non-projective).
Fig. 2.

Definition 3 *A* valency *is an expression of one of the forms* $+L:r, +R:r$ *(a* positive *valency), or* $-L:r, -R:r$ *(a* negative *valency),* r *being a dependency name*[3]. *A terminal* n *is* polarized *if a finite list of valencies* $V(n)$ *(its* valency list*) is assigned to it.* n *is* positive, *if* $V(n)$ *does not contain negative valencies, otherwise it is* negative. *As it concerns nonterminals, we presume that* C *is decomposed into two classes : of* positive *(*$C^{(+)}$*) and* negative *(*$C^{(-)}$*) nonterminals respectively. A D-tree with polarized nodes is* positive *if its root is* positive, *otherwise it is* negative.

A D-structure π *on a string* x *of polarized symbols is a* D-structure with valencies *(a* DV-structure*) on* x, *if the following conditions are satisfied :*

(v1) if a terminal node n *of* π *is negative, then* $V_\pi(n)$ *(or* $V(n)$ *when clear) contains exactly one negative valency,*

(v2) if a dependency of π *enters a node* n, *then* n *is positive,*

(v3) the non-head components of π *(if any) are all negative.*

The polarity of a DV structure *is that of its head.*

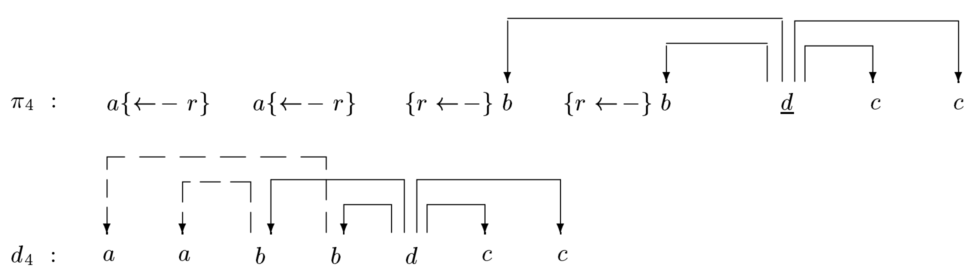


Fig. 3.

[3] We denote the valencies by dashed arcs in braces entering from or going to the indicated direction.

For example, π_4 in Fig. 3 is a DV-structure, in which two occurrences of a with $V(a) = \{-R : r\}$ are two negative non-head components, two occurrences of b with $V(b) = \{+L : r\}$ belong to the head component, and V is empty for the other three occurrences.

Valencies are *saturated* by *long dependencies*, eventually completing terminal DV-structures so that they become D-trees.

Definition 4 *Let π be a terminal DV-structure. A triplet $l =< n_1, n_2, r >$, where n_1, n_2 are nodes of π and $r \in N$, is a* long dependency *with the name r, directed from n_1 to n_2 (notation: $n_1 \overset{r}{--\!>} n_2$), if there are valencies $v_1 \in$* V1p9kPE *such that* kz8ffffff(ff =k1x9kP(((kWff (E =k1e9kP)E ff =k1 (kWff (E =)9kP2 (k)E ff =

(v4) either $n_1 < n_2$ (n_1 precedes *n_2), v_1 Rr, and v_2 $-Lr$, or*

(v5) $n_2 < n_1$, v_1 Lr, and v_2 $-Rr$.

will say that v 1LLdLLsaturates *v_2 by long dependency l.* WMWaWa2The3aMW"Wa2set3aMPW"Wa"k"""MWa *$v_1, v_2 \in V_\pi n_2$ and $n_1 < n_2$,*

$v_1, v_2 \in V_\pi n$ and v_1

Let π_1 be the structure resulting

the long dependency l

and replacing $V_\pi n_1$ by $V_\pi\ n_1$ $V_\pi n_1 \setminus \{v_1$ and $V_\pi n_2$ by $V_\pi\ n_2$

$V_\pi n_2 \setminus \{v_2$. We will say that π_1 is a saturation *of π by l and denote it by $\pi \prec \pi_1$. Among all possible saturations of π wewillsele ct thefollowing particular one :*

Let $v_1 \in V_\pi n_1$

saturated we use the following notion of *integral valency*

Definition 5 *Let π be a terminal DV-structure. If its maximal FA-saturation $MS^1(\pi)$ is a d-tree, we say that this D-tree* saturates π *and call* π saturable*. The* integral valency $\sum\limits_{FA} \pi$ *of π is the list* $\bigcup\limits_{n \in \mu(w(\pi))} V_{MS^1(\pi)}(n)$ *ordered by the order of valencies in π.*

y this definition, $\sum\limits_{FA} MS^1(\pi) = \sum\limits_{FA} \pi$.

Saturability is easily expressed in terms of integral valency.

Lemma 2 *Let π be a terminal DV-structure. Then :*
(1) $MS^1(\pi)$ is a D-tree iff it is cycle-free and $\sum\limits_{FA} \pi = \emptyset$,
(2) π has at most one saturating D-tree.

Nonterminals in DV-structures represent types of substructures. The types are realized through composition of DV-structures which extends that of D-structures.

Definition 6 *Let π_0 be a DV-structure, A be some its nonterminal node, and π_1 be some DV-structure* of the same polarity as A*. Then the* composition *of π_1 into π_0 in the place of A is the structure $\pi = \pi_0[A \backslash \pi_1]$, in which valency sets of terminal nodes and polarities of nonterminal nodes $B \neq A$ are the same as in π_0 and π_1.*

This composition has the following natural properties.

Lemma 3 *(1) If $\pi = \pi_0[A \backslash \pi_1]$ is defined for DV-structures π_0 and π_1, then it is also a DV-structure.*
(2) $\pi_0[A \backslash \pi_1]$ has the same polarity as π_0.

Lemma 4 *Let π_1, π_2 be terminal DV-structures and π_0 a DV-structure such that the compositions $\pi_0[A \backslash \pi_1]$, $\pi_0[A \backslash \pi_2]$ are both defined, and $\sum\limits_{FA} \pi_1 = \sum\limits_{FA} \pi_2$. Then $\sum\limits_{FA} \pi_0[A \backslash MS^1(\pi_1)] = \sum\limits_{FA} \pi_0[A \backslash MS^1(\pi_2)]$.*

Composition may be used to project some DV-structures onto shallow DV-structures representing the same non saturated valencies.

Definition 7 *Let us fix a set $\bar{A} = \{\bar{A} \mid A \in C\}$ of* terminal doubles *of nonterminals in C. Let π_0 be a DV-structure with k occurrences of nonterminals $A_1, ..., A_k$, and $\pi_1, ..., \pi_k$ be some terminal DV-structures with the corresponding polarities. Then the DV-structure $\pi = \pi_0[A_1 \backslash \bar{A}_1, ..., A_k \backslash \bar{A}_k]$ with the valencies defined by assignments: $V_\pi(\bar{A}_i) = \sum\limits_{FA} \pi_i$, $1 \leq i \leq k$, is a* projection *of the structure $\pi_0[A_1 \backslash \pi_1, ..., A_k \backslash \pi_k]$ on π_0. We denote it by $\pi_0\{A_1[\sum\limits_{FA} \pi_1], ..., A_k[\sum\limits_{FA} \pi_k]\}$.*

Clearly, the so defined projection is unique. It also captures the integral valency of the projected structure when this structure results by composition from maximal FA-saturated structures.

Lemma 5 *If the DV-structure* $\pi_0[A_1\backslash\pi_1, ..., A_k\backslash\pi_k]$ *is defined for some* π_0, π_1, ..., π_k, *then*

$$\sum_{FA}\pi_0\{A_1[\sum_{FA}\pi_1], ..., A_k[\sum_{FA}\pi_k]\} = \sum_{FA}\pi_0[A_1\backslash MS^1(\pi_1), ..., A_k\backslash MS^1(\pi_k)].$$

This Lemma follows directly from Lemma 4 because the unit terminal structures $\bar{A}_i$ with valency sets $V_\pi(\bar{A}_i)$, $1 \le i \le k$, are maximally FA-saturated.

Adding long dependencies to DV-structures may in general cause cycles. For example, the structure π_5' in Fig. 4 is the result of adding long dependencies $b \overset{1}{--\!>} e$ and $g \overset{2}{--\!>} b$ into the four-component DV-structure π_5.

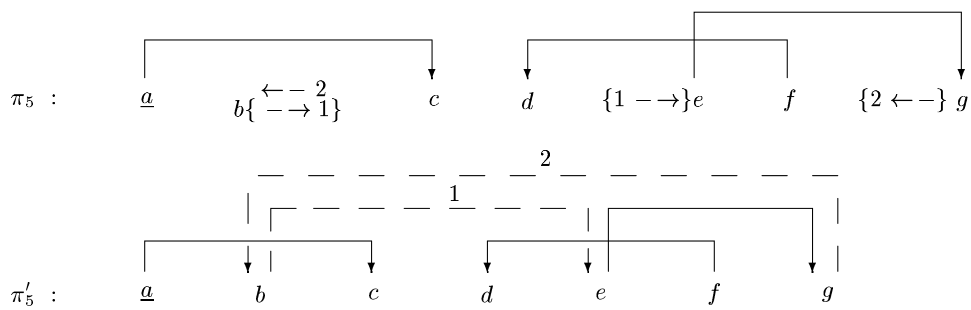


Fig. 4.

Projections capture the existence of cycles in maximal saturations of terminal DV-structures.

Lemma 6 *Let* $\pi_0[A_1\backslash\pi_1, ..., A_k\backslash\pi_k]$ *be defined for some terminal DV-structures* π_0, π_1, ..., π_k, *and* $\hat{\pi}$ *be the projection of the structure* $\pi = \pi_0[A_1\backslash MS^1(\pi_1), ..., A_k\backslash MS^1(\pi_k)]$ *on* π_0. *If* $MS^1(\pi)$ *has a cycle, whereas all* $MS^1(\pi_1), ..., MS^1(\pi_k)$ *are cycle-free, then* $MS^1(\hat{\pi})$ *also has a cycle.*

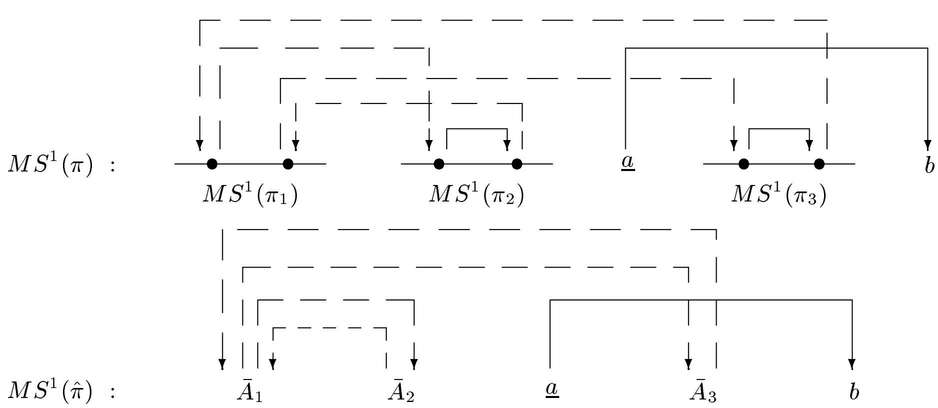


Fig. 5.

Fig. 5 presents an example of a correspondence between the maximal FA-saturation $MS^1(\pi)$ of a composition $\pi = \pi_0[A_1\backslash MS^1(\pi_1), A_2\backslash MS^1(\pi_2), A_3\backslash MS^1(\pi_3)]$ and the maximal FA-saturation of its projection $\hat{\pi}$.

3 Dependency tree grammars

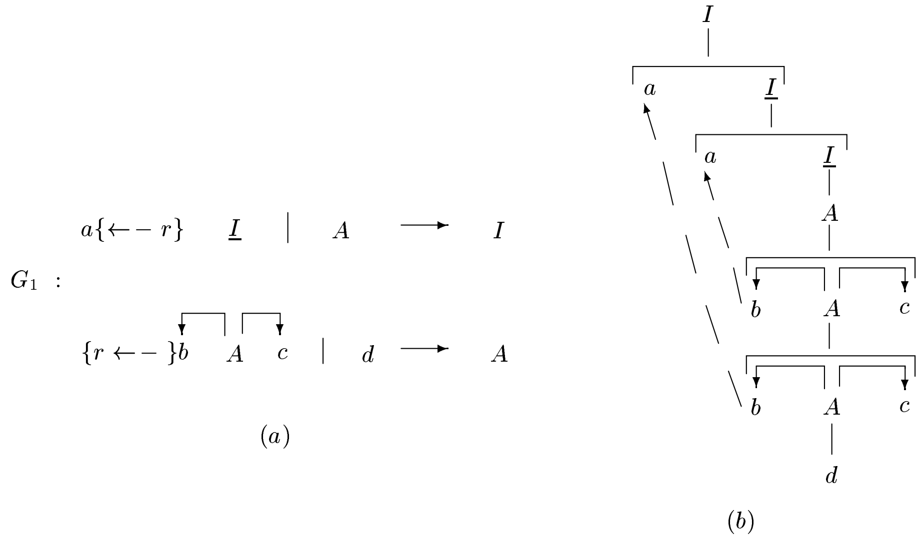
Dependency tree grammars determine DV-structures in the bottom-up manner: they reduce DV-structures to their types. The reduction step is a composition of DV-structures followed by FA-saturation. The yield of a successful reduction is a D-tree.

Definition 8 Syntax. *A* dependency tree (DT-) grammar *is a system* $G = (W, C, N, I, R)$*, where* $I \in C^{(+)}$ *is the axiom (which is a positive nonterminal), and* R *is a set of reduction rules of the form* $\pi \rightarrow A$*, where* $A \in C$ *and* π *is a DV-structure of the same polarity as* A*. In the special case, where the DV-structures in the rules are D-trees, the DT-grammar is* local (LDT-grammar)[5]*.*
Semantics. *1. A terminal rule* $r = (\pi \rightarrow A)$ *is a* reduction *of the structure* $MS^1(\pi)$ *to its type* A *(notation* $MS^1(\pi) \vdash^r A$*).*
[*The* integral valency *of* $MS^1(\pi)$ via r *is* $\sum\limits_r = \sum\limits_{FA} \pi$, π *is the* projection *of this reduction, and* r *has the* reduction rank $rr(r) = 1$.] [6]
2. Let $r = (\pi \rightarrow A)$ *be a reduction rule with* k *nonterminals occurrences* $A_1, ..., A_k$ *in* π, $k > 0$, *and* $\pi_1 \vdash^{\rho_1} A_1$, ..., $\pi_k \vdash^{\rho_k} A_k$ *be some reductions. Then* $\rho = (\rho_1 ... \rho_k; r)$ *is a reduction of the structure* $\pi_0 = MS^1(\pi[A_1\backslash\pi_1, ..., A_k\backslash\pi_k])$ *to its type* A *(notation* $\pi_0 \vdash^\rho A$*).* $\rho_1, ..., \rho_k$ *as well as* ρ *itself are* subreductions *of* ρ*.*
[*The* integral valency *of* π_0 via ρ *is* $\sum\limits_\rho \pi_0 = \sum\limits_{FA} \pi[A_1\backslash\pi_1, ..., A_k\backslash\pi_k] = \sum\limits_{FA} \pi_0$, $\pi\{A_1[\sum\limits_{\rho_1} \pi_1], ..., A_k[\sum\limits_{\rho_k} \pi_k]\}$ *is the* projection *of this reduction, and* ρ *has the* reduction rank $rr(\rho) = 1 + max\{rr(\rho_1), ..., rr(\rho_k)\}$.]
A D-tree d *is* determined *by* G *if there is a reduction* $d \vdash^\rho I$*. The* DT-language determined by G *is the set* $D(G)$ *of all D-trees it determines.* $L(G) = \{w(d) \mid d \in D(G)\}$ *is the* language determined by G. $\mathcal{L}(DTG)$ *and* $\mathcal{L}(LDTG)$ *denote the classes of languages determined by DT-grammars and LDT-grammars respectively.*

Remark. In this definition, the DT-grammars are devoid of various features indispensable for their application in practice, but unnecessary for purely theoretical analysis. In fact, as the categorial grammars, they should have a set of elementary types $E \subset C$. The valencies should be assigned to elementary types (and not to words). The DV-structures should be defined on E^+. There should be a *lexical interpretation* $\lambda \subseteq W \times E$ propagated to phrases by: $a_1...a_n \lambda e_1...e_n$, where $a_i \lambda e_i$, $1 \leq i \leq n$. Respectively, $L(G) = \{x \in W^+ \mid \exists \pi \in D(G)\ (x \lambda w(\pi))\}$. In Fig. 6, we show several rules of the kind.

[5] LDT-grammars are strongly equivalent to dependency tree grammars of [13, 3] which are generating and not analysing as here.
[6] We put into square brackets the notions used for technical needs.

Proposition 2 $\mathcal{D}(P\!-\!DG) \subset \mathcal{D}(LDTG) \subset \mathcal{D}(DTG)$ [7].



In the first rule, $V(a) = \{-R : r\}$ and $I \in C^{(+)}$ is the unit head component. In the third rule, $V(b) = \{+L : r\}$ and the DV-structure in the left part is the single head component with the root A.

Fig. 7.

We are not aiming at a precise c

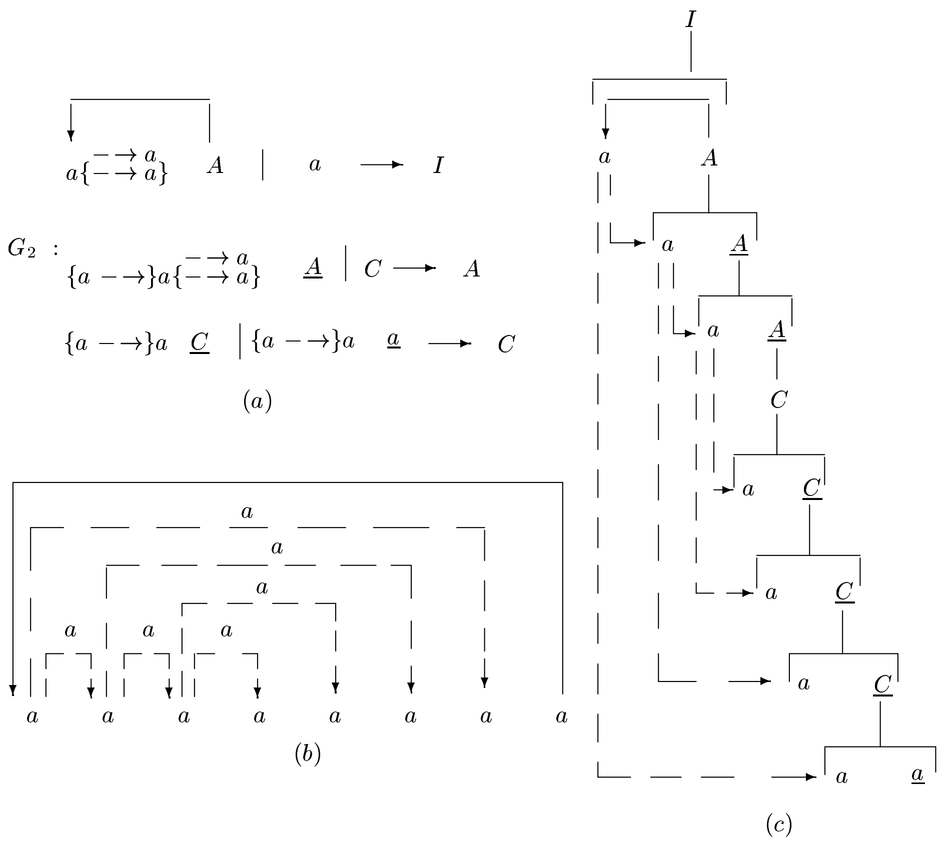


Fig. 8.

It is easy to see that in grammars G_1 in Fig. 7 and G_2 in Fig. 8 the size of integral valency of subreductions grows unlimited (cf. the integral valency of subreductions to type C in DT-grammar G_2). Meanwhile, in grammar G_3 in Fig. 9, the integral valency of all subreductions is either one-element, or empty. So $\sigma(G_3) = 1$. The bounded defect property is of course not local, but as we will see, the grammars can be compiled into equivalent grammars with local control of defect bounded by a constant.

As it concerns local cycles check, we can point out another property of DT-grammars which turns out to be helpful.

Definition 10 *A reduction* $\pi \vdash^{\rho} A$ *is* locally cycle-free (lc-free)*, if for any its subreduction* ρ'*, and its projection* π'*,* $MS^1(\pi')$ *is cycle-free. A DT-grammar* G *is* lc-free *if any reduction in* G *is lc-free.*

This property is sufficient for a reducible structure to be cycle-free.

Lemma 7 *For any lc-free reduction* $\pi \vdash^{\rho} A$*,* $MS^1(\pi)$ *is cycle-free.*

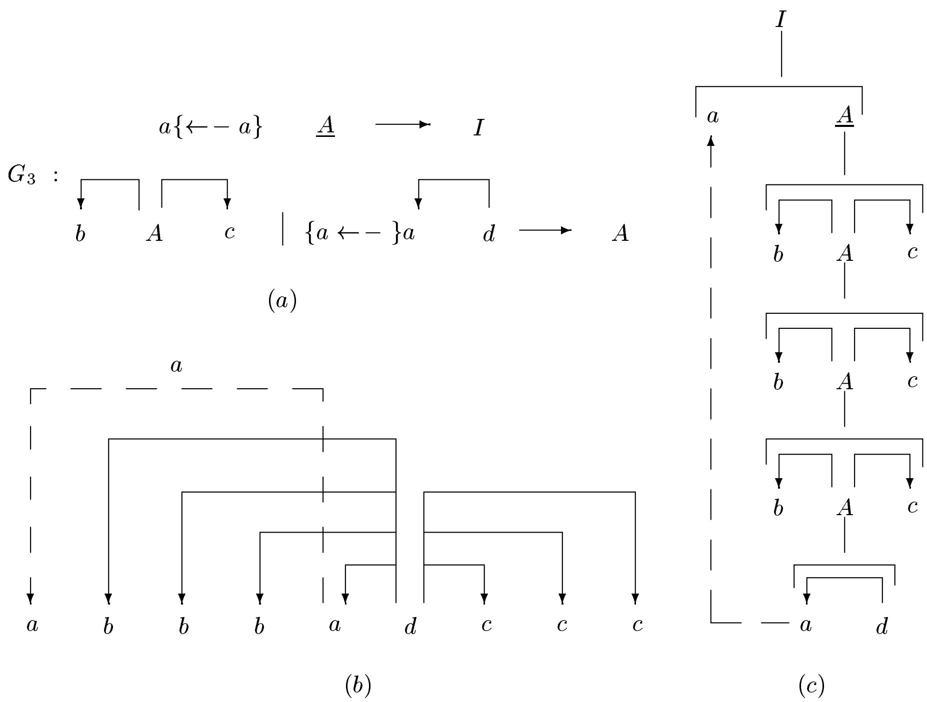


Fig. 9.

All DT-grammars, which appear in our examples, are lc-free. The next theorem shows that lc-free DT-grammars of bounded defect are compiled into cf-grammars which simulate reducibility of D-trees by their rules.

Theorem 1 *For any lc-free DT-grammar G of bounded defect there is a weakly equivalent cf-grammar G_{cf}.*

Proof. In order to simplify notation in this proof, we presume that in nonterminal rules $\pi \to A$, $w(\pi)$ has no terminals. Let $\sigma(G) = q$. The cf-grammar G_{cf} is constructed as follows. It has the same terminals as G, the set of nonterminals of the form $A[V]$, where V is a list of valencies of bounded size $|V| \leq q$, the axiom $I[\emptyset]$, and the rules of the form :

$A[\sum_{FA} \pi] \ \to \ w(\pi)$, where $r = (\pi \to A)$ is a terminal rule of G, and

$A[V_0] \ \to \ A_1[V_1]...A_k[V_k]$,

where for some rule $r = (\pi \to A)$ of G, $w(\pi) = A_1...A_k$, and $V_0, V_1, ..., V_k$ are valency lists of bounded size: $|V_i| \leq q, 0 \leq i \leq k$, such that:

(1) $V_0 = \sum_{FA} \bar{\pi}$, where $\bar{\pi} = \pi[A_1 \backslash \bar{A}_1, ..., A_k \backslash \bar{A}_k]$, with valency set assignments $V(\bar{A}_i) = V_i, 1 \leq i \leq k$, and

(2) $MS^1(\bar{\pi})$ is cycle-free.

The weak equivalence $L(G_{cf}) = L(G)$ follows by Lemmas 2, 7 from the assertion :

Claim. *For $x \in W^+$, there is a derivation $A[V_0] \Rightarrow^*_{G_{cf}} x$ iff there is a reduction $\pi_0 \vdash^\rho A$ such that $x = w(\pi_0)$ and $V_0 = \sum_\rho \pi_0$.* □

An important consequence of Theorem 1 is that lc-free bounded defect DT-grammars have a $\boldsymbol{O}(n^3)$ parsing algorithm.

Corollary 1 *For each lc-free DT-grammar G of bounded defect there is a parsing algorithm for $L(G)$ in time $\boldsymbol{O}(n^3)$.*

Proof. We follow the construction of Theorem 1, and transform G into a strongly equivalent lc-free DT-grammar G^q with defect $q = \sigma(G)$ and with nonterminals keeping integral valencies. In the place of the cf-rule $A[V_0] \to A_1[V_1]...A_t[V_t]$ constructed from a rule $\pi \to A$ of G, we will have in G^q the rule $\pi[A_1\backslash A_1[V_1], ..., A_k\backslash A_k[V_k]] \to A[V_0]$. We apply to G^q the Earley algorithm in charter form (a charter being a DV-structure). □

Remark. The size of G^q is $v^{k(q+1)}$ times greater than that of G, where v is the number of valencies and k is the maximal length of left parts of rules. So theoretically, the constant factor in the $\boldsymbol{O}(n^3)$ time bound is great. In practice, it shouldn't be as awful, because the majority of valencies combinations in the rules can be filtered out while the compilation, and those which are relevant are few (cf. http://bobo.link.cs.cmu.edu/index.html/ where this kind of precompilation is made for link grammars [19]).

Lc-free bounded defect DT-grammars are stronger than LDT-grammars. The DT-language of the grammar G_3 in Fig. 9 models constructions of the type : *"Who do you think Mary thinks John thinks ... Ann thinks Peter loves?"* This DT-language cannot be determined by LDT-grammars, whereas, it can be determined by a DT-grammar with a single long dependency.

Theorem 2 *No LDT-grammar is strongly equivalent to G_3.*

4 Conclusion

The lc-free DT-grammars of bounded defect are essentially simpler to parse as compared with non-projective D-grammars with lifting control [2, 8, 11]. Their weak equivalence to cf-grammars shouldn't be interpreted as a disappointing factor. Really essential is their *strong generative capacity.* We conjecture that in many languages the defect through nesting is bounded by a small constant (at most 2 or 3). If this is true, the bounded defect DT-grammars are strong enough to treat the majority of discontinuous constructions.

Acknowledgments

We wish to thank N. Pertsov and an anonymous reviewer for constructive criticism and comments which helped us to fix inconsistencies in the preliminary version of this paper.

References

1. Y. Bar-Hillel, H. Gaifman, and E. Shamir. On categorial and phrase structure grammars. *Bull. Res. Council Israel*, 9F:1–16, 1960.

2. N. Bröker. Separating surface order and syntactic relations in a dependency grammar. In *Proc. COLING-ACL*, pages 174–180, Montreal, 1998.
3. A.Ja. Dikovsky and L.S. Modina. Dependencies on the other side of the curtain. *Traitement Automatique des Langues (TAL)*, 41(1):79–111, 2000.
4. H. Gaifman. Dependency systems and phrase structure systems. Report p-2315, RAND Corp. Santa Monica (CA), 1961. Published in: Information and Control, 1965, v. 8, n. 3, pp. 304-337.
5. M. Hepple. A dependency-based approach to bounded & unbounded movement. In T. Becker and H.-U. Krieger, editors, *Proc. of the 5th Meeting on Math. and Language*, 1997.
6. R.A. Hudson. *Word Grammar*. Basil Blackwell, Oxford-New York, 1984.
7. A.K. Joshi, L.S. Levy, and M. Takahashi. Tree adjunct grammars. *Journ. of Comput. and Syst. Sci.*, 10(1):136–163, 1975.
8. S. Kahane, A. Nasr, and O. Rambow. Pseudo-projectivity : A polynomially parsable non-projective dependency grammar. In *Proc. COLING-ACL*, pages 646–652, Montreal, 1998.
9. A. Lecomte. Proof nets and dependencies. In *Proc. of COLING-92*, pages 394–401, Nantes, 1992.
10. V. Lombardo and L. Lesmo. An earley-type recognizer for dependency grammar. In *Proc. 16th COLING*, pages 723–728, 1996.
11. V. Lombardo and L. Lesmo. Formal aspects and parsing issues of dependency theory. In *Proc. COLING-ACL*, pages 787–793, Montreal, 1998.
12. I. Mel'čuk. *Dependency Syntax*. SUNY Press, Albany, NY, 1988.
13. L.S. Modina. On Some Formal Grammars Generating Dependency Trees. In *Proc. of the MFCS'75, Lecture Notes in Computer Science*, number 32, pages 326–329, 1975.
14. M. Moortgat. La grammaire catégorielle généralisée : le calcul de lambek-gentzen. In Ph. Miller and Th. Torris, editors, *Structure of languages and its mathematical aspects*, pages 127–182. Hermes, Paris, 1990.
15. M. Moortgat and R. Oehrle. Adjacency, dependency and order. In *Proc. of Ninth Amsterdam Colloquium*, 1994.
16. A. Nasr. A formalism and a parser for lexicalized dependency grammars. In *Proc. Int. Workshop on Parsing Technology*, pages 186–195, Prague, 1995.
17. P. Neuhaus and N. Bröker. 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, 1997.
18. Jane J. Robinson. Dependency structures and transformational rules. *Language*, 46(2):259–285, 1970.
19. D. D. Sleator and D. Temperly. Parsing English with a Link Grammar. In *Proc. IWPT'93*, pages 277–291, 1993.
20. E. Stabler. Derivational minimalism. In Ch. Retoré, editor, *Logical Aspects of Computational Linguistics*, number 1328 in LNAI, pages 68–95, Nancy, 1996. Springer Verlag.

APPENDIX

Proof of Lemma 1. (1) In order to add a long dependency $l = \left(n_1 \ \overset{r}{- - >} \ n_2\right)$ to π, the node n_2 must be negative. By point $(v2)$ of Definition 3, this excludes the existence of another dependency (local or long) entering n_2.

(2) As a consequence of (1), n_2 can only be the root of a negative component β of π. So if n_1 also belongs to β, then a cycle appears in this component. Suppose that n_1 does not belong to β and α is a component of π which contains it. Let us verify that in this case π_1 is a DV-structure. First of all, let us remark that introduction of l into π cannot augment valency sets, which means that point (v1) of Definition 3 is satisfied in π_1. By definition of saturation, after that l was added, the negative valency it saturates is subtracted from $V(n_2)$ so by point (v1), n_2 becomes positive in π_1. This means that point (v2) of Definition 3 is also true in π_1. Suppose that α is positive in π. Then it is the head component of π, and it is changed. Indeed, adding l makes $\alpha \cup \beta$ a tree, hence a new component of π_1. Negative valencies are never introduced by saturation, therefore this component is positive. So it is a new (and single) head component of π_1. Negative components different from β rest intact. So points (v3) of Definition 3 is also true, and π_1 is a DV-structure in this case. Now suppose that α is also negative. The subgraph $\alpha \cup \beta$ becomes a tree in π_1. Its root is that of α. So it rests negative, and $\alpha \cup \beta$ becomes a new negative component of π_1. Other negative components rest intact. If there was a positive head component in π, then it also rest intact, and single positive component in π_1. So in this case too point (v3) of Definition 3 is satisfied, and π_1 is a DV-structure. □.

Proof of Lemma 2. (1) The only-if part being trivial, let us prove the if-part. By assumption, π is a terminal DV-structure. So it is a union of components $\alpha_0, \alpha_1, ..., \alpha_t$, each component being a tree. One of these components, e.g., α_0 is the head. By point (v3) of Definition 3, all the other components are negative. As we have already remarked in the proof of the Lemma 1, a unique negative valency in each component α_i, $1 \leq i \leq t$, is that of its root. Suppose that $MS^1(\pi)$ is cycle-free. Then by Lemma 1, it is a DV-structure. The condition $\sum\limits_{FA} \pi = \emptyset$ means that all valencies are saturated in $MS^1(\pi)$. So $MS^1(\pi)$ is the union of non-intersecting subtrees $\alpha_0, \alpha_1, ..., \alpha_t$, expanded by arrows in a way that the root of each subtree (maybe besides α_0) is entered by a new arrow, and no two arrows (old or new) enter the same node. Let us consider the graph $c(MS^1(\pi))$, with the nodes $\alpha_0, \alpha_1, ..., \alpha_t$, in which an arrow goes from α_i to α_j, if there is an arrow in $MS^1(\pi)$ from a node of α_i to the root of α_j. Clearly, $MS^1(\pi)$ is a D-tree iff $c(MS^1(\pi))$ is a tree. We have already seen that $c(MS^1(\pi))$ is cycle-free and that no two its arrows enter the same node. It rests for us to prove that it is connected. Suppose the contrary. Let us consider a maximal connected subgraph of $c(MS^1(\pi))$, which doesn't contain α_0, if it is positive. In this subgraph, for each node there is an arrow entering it. So this subgraph contains a cycle, which contradicts the premise.

Point (2) is an immediate consequence of the uniqueness of the maximal FA-saturation of a terminal DV-structure. □

Proof of Lemma 3. The composition is never made in a terminal node. This means that point (v1) of Definition 3 is satisfied in π, if it is defined. In order to verify (v2), we should check only one node : the new node n, which replaces A in π_0. Suppose that a dependency entered A in π_0. Then A is a positive nonterminal. If composition is defined, then π_1 is a positive DV-structure. There are two cases. First is that the root of π_1 is a positive nonterminal B. Then $n = B$ is a positive node of π. Second is that the root of π_1 is a terminal b with positive $V_{\pi_1}(b)$. Then $n = b$ and $V_{\pi_1}(b) = V_{\pi(n)}$. So in both cases, point (v2) of Definition 3 is satisfied in π. In order to verify point (v3), let us consider a non-head component α of π. If α is a non-head component of π_1, or it is a non-head component of π_0, which is not affected by composition, then it is either a non-head component of π. Suppose that α is the result of composing the head component π_{11} of π_1 into a component β of π_0 in A. The fact that α is a non-head component of π means that β was a non-head component of π_0. Therefore, β was negative. If A is not the root of β, then the root is not affected by composition and rests negative. So is α in this case. If A is the root of β, then it is a negative nonterminal. So if π is defined, then both π_1 and π_{11} are negative, and therefore, the root of π_{11}. Another two cases arise. First is that the root of π_{11} is a negative nonterminal B. Then B becomes the root of α. Second is that the root of π_{11} is a terminal b with negative $V_{\pi_1}(b)$. Then b becomes the root of α, and $V_{\pi_1}(b) = V_{\pi(b)}$. In both cases, α is negative. So point (v3) of Definition 3 is also satisfied in π, and it is a DV-structure.

This reasoning also proves point (2). $\square$

Proof of Lemma 4. In fact, it suffice to prove the following assertion :

Claim. *If there is a long dependency* $l = \left(n \overset{r}{--\!>} n'\right)$ *which is first available in* $\pi_0[A\backslash MS^1(\pi_1)]$*, then the first available long dependency in* $\pi_0[A\backslash MS^1(\pi_2)]$ *exists and saturates exactly the same valencies as* l.

Then the Lemma will result by the evident induction on the number of FA-saturation steps needed to obtain $MS^1(\pi_0[A\backslash MS^1(\pi_1)])$ from $\pi_0[A\backslash MS^1(\pi_1)]$. Now, suppose that l is the first available long dependency in $\pi_0[A\backslash MS^1(\pi_1)]$, and that n belongs to π_1, n' belongs to π_0, and l saturates a valency v' by a valency v. This means that v' is a negative valency, $v' \in V_{\pi_0}(n')$, $v \in V_{MS^1(\pi_1)}(n)$, and so $v \in \sum\limits_{FA} \pi_1 = \sum\limits_{FA} \pi_2$. Thus in DV-structure $MS^1(\pi_2)$, there is a node m, which has the same positive valency $v \in V_{MS^1(\pi_2)}(m)$, and which is not saturated inside $MS^1(\pi_2)$. Let us remark that the integral valencies $\sum\limits_{FA} \pi_1$, $\sum\limits_{FA} \pi_2$ coincide as lists (i.e. together with the order of valencies as it is defined in Definition 4). So if there were some other positive valency in $MS^1(\pi_2)$ in some its node m_1 preceding m and saturating a negative valency in π_0, then some corresponding node with this property would also be in $MS^1(\pi_1)$. Therefore, l would not be the first available. This means that the long dependency $l' = \left(n \overset{r}{--\!>} m\right)$ is first available in $\pi_0[A\backslash MS^1(\pi_2)]$ and it also saturates v' by v.

A similar reasoning applies in the other cases: when n belongs to π_0 and n' belongs to π_1 or when both n and n' belong to π_0. $\square$

Proof of Lemma 6. Let us call "old" the dependencies in $MS^1(\pi_1), ..., MS^1(\pi_k)$, and "new" those long dependencies in $MS^1(\pi)$, which are not old. $MS^1(\pi_1), ..., MS^1(\pi_k)$ being cycle-free and $MS^1(\pi)$ having a cycle, there is at least one new dependency $l = \left(n_1 \overset{r}{--\!>} n_2\right))$ in $MS^1(\pi)$, which belongs to the cycle but not to the components $MS^1(\pi_1), ..., MS^1(\pi_k)$. Without loss of generality, we can presume that this dependency is the first available in π. Suppose that l saturates valency $v_2 = -O : r$ by valency $v_1 = +\bar{O} : r$. Let us consider the case, where there are some components $MS^1(\pi_{i_1})$, $MS^1(\pi_{i_2})$ of π, which contain respectively the nodes n_1 and n_2, other cases being similar. By the choice of l, $i_1 \neq i_2$, $v_1 \in \sum\limits_{FA} \pi_{i_1}$ and $v_2 \in \sum\limits_{FA} \pi_{i_2}$. By definition of projection, we have $v_1 \in V_{\hat{\pi}}(\bar{A}_{i_1})$ and $v_2 \in V_{\hat{\pi}}(\bar{A}_{i_2})$. Let us show that there is a long dependency $l' = \left(\bar{A}_{i_1} \overset{r}{--\!>} \bar{A}_{i_2}\right)$ in $MS^1(\hat{\pi})$, which saturates v_2 by v_1. Since l is first available in π, v_1 is the first non saturated positive valency in π. Moreover, by definition of the composition of DV-structures, it is also the first positive valency in $\sum\limits_{FA} \pi_{i_1}$. So it is also first in $\hat{\pi}$ and in $V_{\hat{\pi}}(\bar{A}_{i_1})$. v_2 belonging to $V_{\hat{\pi}}(\bar{A}_{i_2})$, this means that a long dependency $l' = \left(\bar{A}_{i_1} \ -\overset{r}{-}> \ \bar{A}_j\right)$ goes from $\bar{A}_{i_1}$ in the direction of $\bar{A}_{i_2}$, but not further then $\bar{A}_{i_2}$. It also cannot be closer to $\bar{A}_{i_1}$ than $\bar{A}_{i_2}$, because in this case n_2 should belong to some component $MS^1(\pi_j)$, which is closer to $MS^1(\pi_{i_1})$ than $MS^1(\pi_{i_2})$. Therefore, $j = i_2$. In this manner, we show that for each new long dependency in $MS^1(\pi)$, which belongs to the cycle but not to the components $MS^1(\pi_1), ..., MS^1(\pi_k)$, which goes from some $MS^1(\pi_i)$ and enters some $MS^1(\pi_j)$, there is a long dependency in $MS^1(\hat{\pi})$ which goes from $\bar{A}_i$ and enters $\bar{A}_j$. This proves that there is a cycle in $MS^1(\hat{\pi})$. □

Proof of Lemma 7. Suppose that $MS^1(\pi')$ has a cycle for some subreduction $\pi' \vdash^{\rho'} A'$ of ρ with $rr(\rho') = 1$ (clearly, this cycle would be present in $MS^1(\pi)$ too). Then $\rho' = r$ and $r = (\pi' \to A')$ is a rule of G. In this case, π' is the projection of ρ' and the presence of a cycle in $MS^1(\pi')$ contradicts the premise of Lemma. This means that if $MS^1(\pi)$ has a cycle, then there is a subreduction $\pi' \vdash^{\rho'} A'$ of ρ such that $rr(\rho') > 1$, $\rho' = (\rho_1 ... \rho_m; r)$, $r = (\pi_0 \to B)$ is a rule of G with nonterminals $B_1, ..., B_m$ in π_0, $\pi_i \vdash^{\rho_i} B_i$ and $MS^1(\pi_i)$ being cycle-free for all $1 \le i \le m$, whereas $MS^1(\pi')$ has a cycle. However in this case, by Lemma 6, the projection $\hat{\pi}' = \pi_0\{B_1[\sum\limits_{FA} \pi_1], ..., B_m[\sum\limits_{FA} \pi_m]\}$ of ρ' produces a cycle in $MS^1(\hat{\pi}')$, which again contradicts the premise of Lemma. □

Proof of the Claim in Theorem 1.
(**Only-if-part**). We prove it by induction on the height of the derivation tree.

If the derivation $A[V_0] \Rightarrow^*_{G_{cf}} x$ has a tree of height 1, then it is a one-step derivation with the rule $A[V_0] \to x$ applied. This rule is constructed from a terminal rule $r = (\pi \to A)$ of G. So for $\pi_0 = MS^1(\pi)$, $\pi_0 \vdash^r A$, $x = w(\pi_0)$, and $V_0 = \sum\limits_{FA} \pi_0 = \sum\limits_{r} \pi_0$.

Suppose that the derivation $A[V_0] \Rightarrow^*_{G_{cf}} x$ has a tree T of height $h > 1$. Then the rule applied at the first step is not terminal. So it has the form $A[V_0] \to A_1[V_1] ... A_k[V_k]$. This means that T can be decomposed into the minimal tree corresponding to the application of this rule to A and k trees of derivations $A_1[V_1] \Rightarrow^*_{G_{cf}} x_1, ..., A_k[V_k] \Rightarrow^*_{G_{cf}} x_k$, where $x = x_1 ... x_k$. These trees are of heights lesser than h. Hence, by induction hypothesis, there are reductions $\pi_i \vdash^{\rho_i} A_i$ such that $x_i = w(\pi_i)$ and $V_i = \sum\limits_{\rho_i} \pi_i$, $1 \le i \le k$. Now let us consider the rule $r = (\pi \to A)$ of G, from which the rule $A[V_0] \to A_1[V_1] ... A_k[V_k]$ was constructed. By point (1) of construction, $V_0 = \sum\limits_{FA} \bar{\pi}$, where $\bar{\pi} = \pi[A_1 \backslash \bar{A}_1, ..., A_k \backslash \bar{A}_k]$, with valency set assignments $V(\bar{A}_i) = V_i, 1 \le i \le k$. On the other hand, by the induction hypothesis, $V_i = \sum\limits_{\rho_i} \pi_i$, $1 \le i \le k$. By Lemma 5, $V_0 = \sum\limits_{FA} \bar{\pi} = \sum\limits_{FA} \pi_0$, where $\pi_0 = MS^1(\pi[A_1 \backslash \pi_1, ..., A_k \backslash \pi_k])$. We see that $\rho = (\rho_1 ... \rho_k; r)$ realizes a reduction $\pi_0 \vdash^{\rho} A$, $\bar{\pi}$ is the projection of this reduction, and $V_0 = \sum\limits_{\rho} \pi_0$. It only remains to remark that by definition of the composition, $w(\pi_0) = x$.

(**If-part**). We prove it by induction on $rr(\rho)$.
If $rr(\rho) = 1$, then ρ is a reduction by a terminal rule $r = (\pi \to A)$ of G. Since $\sigma(G) = q$, to this rule corresponds in G_{cf} the rule $A[\sum\limits_{FA} \pi] \to w(\pi)$. This rule gives

the needed one-step derivation $A[V_0] \Rightarrow^*_{G_{cf}} x$ for $V_0 = \sum_{FA} \pi$ and $x = w(\pi)$.

Suppose that $\pi_0 \vdash^{\rho} A$, $rr(\rho) = j > 1$, $V_0 = \sum_{\rho} \pi_0$, and the if-part of the Claim is proven for the reductions of ranks lesser than j. This means that there is a rule $r = (\pi \to A)$ in G such that $w(\pi) = A_1...A_k$, $\rho = (\rho_1...\rho_k; r)$, and subreductions $\pi_i \vdash^{\rho_i} A_i$, of ranks $rr(\rho_i) < j$, $1 \leq i \leq k$, such that $\pi_0 = MS^1(\pi[A_1\backslash\pi_1, ..., A_k\backslash\pi_k])$. By Lemma 5, the projection $\bar{\pi} = \pi\{A_1[\sum_{\rho_1} \pi_1], ..., A_k[\sum_{\rho_k} \pi_k]\}$ of ρ satisfies the equation $V_0 = \sum_{FA} \bar{\pi}$. Besides this, $|V_0| \leq \sigma(G) = q$. Next, by induction hypothesis, for each $i, 1 \leq i \leq k$, and for $V_i = \sum_{\rho_i} \pi_i$, there is a derivation $A_i[V_i] \Rightarrow^*_{G_{cf}} x_i$, where $x_i = w(\pi_i)$. Hence, $w(\pi) = w(\pi_1)...w(\pi_k) = x$. Now, G being lc-free, $MS^1(\bar{\pi})$ is cycle-free. So the rule r satisfies the conditions (1),(2) in the construction of G_{cf}. Therefore, G_{cf} has the rule $A[V_0] \to A_1[V_1]...A_k[V_k]$ constructed from r. This rule can be used in the derivation
$A[V_0] \Rightarrow^*_{G_{cf}} A_1[V_1]...A_k[V_k] \Rightarrow^*_{G_{cf}} x_1 A_2[V_2]...A_k[V_k] \Rightarrow^*_{G_{cf}} x_1...x_k = x$. □

Proof of Theorem 2. Suppose that there is an LDT-grammar G such that $D(G) = D(G_3)$. Let T be the tree of some derivation $I \Rightarrow^*_G \pi$ with terminal π. Let N_1 and N_2 be some nonterminal nodes of T such that N_2 is a descendent of N_1. This means that the complete subtree $T(N_2)$ of T with the root N_2 is a complete subtree of $T(N_1)$ and $w(T(N_1)) = \omega_1 w(T(N_2))\omega_2$ for some terminal strings ω_1, ω_2 (called respectively *left* and *right wings* of N_1, N_2). The pair of nodes N_1, N_2 is *iterable*, if the nodes have the same nonterminal label B. Clearly, the length of the longest terminal string which does not contain an occurrence of a (left or right) wing of an iterable pair is bounded by $\beta(G) = c(r-1)2^{(|C|+1)log\ r}$, where r is the maximal length of left hand sides of rules of G. So if we consider a DV-structure π_G such that $w(\pi_G) = ab^{\beta(G)+1}adc^{\beta(G)+1}$, we can find in any its derivation tree T an iterable pair N_1, N_2, whose nonempty left or right terminal wing ω_1 is a substring of $b^{\beta(G)+1}$. If this wing were right, replacing $T(N_1)$ by $T(N_2)$ we would obtain a DV-structure of a string of the form $ab^i adc^{\beta(G)+1}$, for some $i < \beta(G)+1$. So ω_1 is a left wing of the iterable pair. The same reasoning shows that the corresponding right wing ω_2 cannot be empty and is a substring of $c^{\beta(G)+1}$. Therefore, in the path from N_2 to d there exists a nonterminal node N' such that a and d both belong to $w(T(N'))$. Let N_3 be the last such node in the path and B be its (nonterminal) label. Since G is a local DT-grammar, π_G is a D-tree in which all dependencies are local. So in particular, in a rule $\pi_1 \to B$ applied in N_3, $w(\pi_1) = \alpha_1 A \alpha_2 D \alpha_3$, where A is a predecessor of the inner occurrence of a in $w(\pi_G)$ (or a itself) and D is a predecessor of the occurrence of d in $w(\pi_G)$ (or d itself). Since d is the root of π_G, there is a local dependency in π_1 from D to A. Moreover, if $A = a$, then in the same π_1 there is an occurrence of a predecessor A_0 of the outermost left occurrence of a in $w(\pi_G)$ (or this outermost left occurrence of a itself) and the dependency from A to A_0. If $A \neq a$, then in the path from A to the inner a there is the first node N_4 labelled by a nonterminal A_1 such that in a rule $\pi_2 \to A_1$ applied in N_4, $w(\pi_2) = \gamma_1 A_l \gamma_2 A_r \gamma_3$, where A_l is a predecessor of the outermost left occurrence of a in $w(\pi_G)$ (or this occurrence of a itself) and A_r is a predecessor of the inner occurrence of a in $w(\pi_G)$ (or this occurrence of a itself) and the dependency from A_r to A_l. In both cases we see that a predecessor of the outermost left occurrence of a in $w(\pi_G)$ is a descendent of N_2 and the more so of N_1. Meanwhile, the outermost left occurrence of a in $w(\pi_G)$ precedes the left wing ω_1 and hence cannot be a descendent of N_1. A contradiction. □