Guaranteeing Parsing Termination of Unification Grammars

Efrat Jaeger and **Nissim Francez**
Department of Computer Science
Technion, Israel Institute of Technology
32000 Haifa, Israel

Shuly Wintner
Department of Computer Science
University of Haifa
31905 Haifa, Israel

Abstract

Unification grammars are known to be Turing-equivalent; given a grammar G and a word w, it is undecidable whether $w \in L(G)$. In order to ensure decidability, several constraints on grammars, commonly known as *off-line parsability (OLP)* were suggested. The recognition problem is decidable for grammars which satisfy OLP. An open question is whether it is decidable if a given grammar satisfies OLP. In this paper we investigate various definitions of OLP, discuss their inter-relations and show that some of them are undecidable.

1 Introduction

Context-free grammars are considered to lack the expressive power needed for modelling natural languages. Unification grammars have originated as an extension of context-free grammars, the basic idea being to augment the context-free rules with feature structures (FSs) in order to express additional information. Today, several variants of unification grammars exist, some of which do not necessarily assume an explicit context-free backbone.

The recognition problem (also known as the membership problem), for a grammar G and a string w, is whether $w \in L(G)$. The parsing problem, for a grammar G and a string w, is to deliver all parse trees that G induces on w, determining what structural descriptions are assigned by G to w. The rest of this paper is concerned with recognition.

Unification grammars have the formal power of Turing machines, thus the recognition problem for them is undecidable. In order to ensure decidability of the recognition problem, a constraint called *off-line parsability (OLP)* was suggested. The recognition problem is decidable for OLP grammars. There exist several variants of OLP in the literature (Pereira and Warren, 1983; Johnson, 1988; Haas, 1989; Torenvliet and Trautwein, 1995; Shieber, 1992; Wintner and Francez, 1999; Kuhn, 1999).

Some variants of OLP were suggested without recognizing the existence of all other variants. In this paper we make a comparative analysis of the different OLP variants for the first time. Some researchers (Haas, 1989; Torenvliet and Trautwein, 1995) conjecture that some of the OLP variants are undecidable (it is undecidable whether a grammar satisfies the constraint), although none of them gives any proof of it. There exist some variants of OLP for which decidability holds, but these conditions are too restrictive; there is a large class of non-OLP grammars for which parsing termination is guaranteed. Our main contribution is to show proofs of undecidability for three OLP definitions.

Section 2 defines the basic concepts of our formalism. Section 3 discusses the different OLP definitions. Section 4 gives an analysis of several OLP definitions and the inter-relations among them. Section 5 proves the undecidability of three of the OLP conditions.

2 Preliminaries

The following definitions are based on Francez and Wintner (In preperation) and Carpenter (1992). Grammars are defined over a finite set FEATS of features, a finite set ATOMS of atoms, and a finite set CATS of categories. A *multi-rooted feature structure* (MRS) is a pair $\langle \bar{Q}, G \rangle$ where $G = \langle Q, \delta, \theta \rangle$ is a finite, directed, labelled graph consisting of a set $Q \subseteq$ NODES of nodes, a partial function, $\delta : Q \times \text{FEATS} \rightarrow Q$, specifying the arcs and a partial function, $\theta : Q \rightarrow$ ATOMS, labelling the sinks, and $\bar{Q}$ is an ordered set of distinguished nodes in Q called *roots*. G is not necessarily connected, but the union of all the nodes reachable from all the roots in $\bar{Q}$ is required to yield exactly Q. The *length* of an MRS is the number of its roots, $|\bar{Q}|$.

Meta-variables σ, ρ range over MRSs, and $\delta, \theta, Q, \bar{Q}$ over their constituents. If $\sigma = \langle \bar{Q}, G \rangle$ is an MRS and $\bar{q}_i$ is a root in $\bar{Q}$ then $\bar{q}_i$ naturally

induces an FS $A_i = (Q_i, \bar{q}_i, \delta_i, \theta_i)$, where Q_i is the set of nodes reachable from $\bar{q}_i$, $\theta_i = \theta|_{Q_i}$ and $\delta_i = \delta|_{Q_i}$. Thus σ can be viewed as an ordered sequence $\langle A_1, \ldots, A_n \rangle$ of (not necessarily disjoint) FSs. We use the two views of MRSs interchangeably.

The *sub-structure* of $\sigma = \langle A_1, \ldots, A_n \rangle$, induced by the pair $\langle i, j \rangle$ and denoted $\sigma^{i \ldots j}$, is $\langle A_i, \ldots, A_j \rangle$. If $i > j$, $\sigma^{i \ldots j} = \lambda$. If $i = j$, σ^i is used for $\sigma^{i \ldots i}$.

An MRS $\sigma = \langle \bar{Q}, G \rangle$ *subsumes* an MRS $\sigma' = \langle \bar{Q}', G' \rangle$ (denoted by $\sigma \sqsubseteq \sigma'$) iff $|\bar{Q}| = |\bar{Q}'|$ and there exists a total function $h : Q \to Q'$ such that for every root $\bar{q}_i \in \bar{Q}, h(\bar{q}_i) = \bar{q}'_i$; for every $q \in Q$ and $f \in$ FEATS, if $\delta(q, f) \downarrow$ then $h(\delta(q, f)) = \delta'(h(q), f)$; and for every $q \in Q$ if $\theta(q) \downarrow$ then $\theta(q) = \theta'(h(q))$.

Skeletal grammars are a variant of unification grammars which have an explicit context-free backbone/skeleton. These grammars can be viewed as an extension of context-free grammars, where every category is associated with an informative FS. An *extended category* is a pair $\langle A, c \rangle$ where A is an FS and $c \in$ CATS.

Definition 2.1. *A **skeletal grammar** (over* FEATS, ATOMS *and* CATS*) is a tuple $G = \langle \mathcal{R}, \mathcal{L}, A^s \rangle$ where $\mathcal{R}$ is a finite set of rules, each of which is an MRS of length $n \geq 1$ (with a designated first element, the head of the rule), and a sequence of length n of categories; $\mathcal{L}$ is a lexicon, which associates with every terminal a (over a fixed finite set Σ of terminals) a finite set of extended categories $\mathcal{L}(a)$; A^s is the start symbol (an extended category).*

A *skeletal form* is a pair $\langle \sigma, \vec{c} \rangle$, where σ is an MRS of length n and $\vec{c}$ is a sequence of n categories ($c_i \in$ CATS for $1 \leq i \leq n$).

Definition 2.2 (Derivation). *Let $\langle \sigma_A, \vec{c_A} \rangle$ and $\langle \sigma_B, \vec{c_B} \rangle$ be forms such that $\sigma_A = \langle A_1, \ldots, A_k \rangle$ and $\sigma_B = \langle B_1, \ldots, B_m \rangle$. $\langle \sigma_A, \vec{c_A} \rangle$ immediately derives $\langle \sigma_B, \vec{c_B} \rangle$ iff there exist a skeletal rule $\langle \rho', \vec{c_R} \rangle \in \mathcal{R}$ of length n and an MRS ρ, $\rho' \sqsubseteq \rho$, such that:*

- *$m = k + n - 2$;*
- *ρ's head is some element i of σ_A: $\rho^1 = \sigma_A^i$;*
- *ρ's body is a sub-structure of σ_B: $\rho^{2 \ldots n} = \sigma_B^{i \ldots i+n-2}$;*
- *The first $i-1$ elements of σ_A and σ_B are identical: $\sigma_A^{1 \ldots i-1} = \sigma_B^{1 \ldots i-1}$;*
- *The last $k-i$ elements of σ_A and σ_B are identical: $\sigma_A^{i+1 \ldots k} = \sigma_B^{m-(k-i+1) \ldots m}$;*
- *$\vec{c_B}$ is obtained by replacing the i-th element of $\vec{c_A}$ by the body of $\vec{c_R}$.*

The reflexive transitive closure of '$\to$' is denoted '$\overset{*}{\to}$'. A form $\langle \sigma_A, \vec{c_A} \rangle$ derives $\langle \sigma_B, \vec{c_B} \rangle$ (denoted $\langle \sigma_A, \vec{c_A} \rangle \overset{*}{\Rightarrow} \langle \sigma_B, \vec{c_B} \rangle$) iff there exist MRSs σ'_A, σ'_B such that $\sigma_A \sqsubseteq \sigma'_A$, $\sigma_B \sqsubseteq \sigma'_B$ and $\langle \sigma'_A, \vec{c_A} \rangle \overset{*}{\to} \langle \sigma'_B, \vec{c_B} \rangle$.

Definition 2.3 (Pre-terminals). *Let $w = a_1 \cdots a_n \in \Sigma^*$. $PT_w(j, k)$ is defined if $1 \leq j \leq k \leq n$, in which case it is the skeletal form, $\langle \langle A_j, A_{j+1}, \ldots, A_k \rangle, \langle C_j, C_{j+1}, \ldots, C_k \rangle \rangle$ where $\langle A_i, C_i \rangle \in \mathcal{L}(a_i)$ for $j \leq i \leq k$.*

Definition 2.4 (Language). *The **language** of a skeletal grammar G is $L(G) = \{ w \in \Sigma^* \mid w = a_1 \cdots a_n$ and $\langle A^s \rangle \overset{*}{\Rightarrow} \langle \langle A_1, \ldots, A_n \rangle, \langle C_1, \ldots, C_n \rangle \rangle \}$, where $\langle A_i, C_i \rangle \in \mathcal{L}(a_i)$ for $j \leq i \leq k$.*

Definition 2.5 (Derivation trees). *(also known as constituent structures, c-structure) Let $G = \langle \mathcal{R}, \mathcal{L}, A^s \rangle$ be a skeletal grammar. A tree is a derivation tree admitted by G iff:*

- *The root of the tree is the start symbol A^s;*
- *The internal vertices are extended categories (over the same features, atoms and categories as the grammar G);*
- *The leaves are pre-terminals of length 1;*
- *If a vertex $\langle A, c \rangle$ has k descendants, $\langle B_1, c_1 \rangle, \langle B_2, c_2 \rangle, \ldots, \langle B_k, c_k \rangle$, then $\langle \langle A \rangle, \langle c \rangle \rangle$ immediately derives $\langle \langle B_1, \ldots, B_k \rangle, \langle c_1, \ldots, c_k \rangle \rangle$ with respect to some rule $\langle \rho, \vec{c_R} \rangle \in \mathcal{R}$.*

Definition 2.6. *A **general unification grammar** (over* FEATS *and* ATOMS*) is a tuple $G = \langle \mathcal{R}, \mathcal{L}, A^s \rangle$ where $\mathcal{R}$ is a finite set of rules, each of which is an MRS of length $n \geq 1$; $\mathcal{L}$ is a lexicon, which associates with every terminal a a finite set of FSs $\mathcal{L}(a)$; A^s is the start symbol (an FS).*

General unification grammar formalism do not assume the existence of a context-free backbone. *Derivations*, *pre-terminals*, *languages* and *derivation trees* for general unification grammars are defined similarly to skeletal grammars, ignoring all categories.

3 Off-line-parsability constraints

It is well known that unification based grammar formalisms are Turing-equivalent in their generative capacity (Pereira and Warren, 1983; Johnson, 1988, 87-93); determining whether a given string w is generated by a given grammar G is equivalent to deciding whether a Turing machine M halts on an empty input, which is known to be undecidable. Therefore, the recognition problem is undecidable in the general case. However, for grammars that satisfy a certain restriction, called off-line parsability constraint (OLP), decidability of the recognition problem is guaranteed. In this section we present some different variants of the OLP constraint suggested in the literature. Some of the constraints (Pereira and Warren, 1983; Kaplan and Bresnan, 1982; Johnson, 1988; Kuhn, 1999) apply only to skeletal grammars since they use the term category which is not well defined for general unification grammars. Others (Haas, 1989; Shieber, 1992; Torenvliet and Trautwein, 1995; Wintner and Francez, 1999) are applicable to both skeletal and general unification grammars.

Some of the constraints impose a restriction on allowable derivation trees, but provide no explicit definition of an OLP grammar. Such a definition can be understood in (at least) two manners:

Definition 3.1 (OLP grammar).

1. *A grammar G is OLP iff for every $w \in L(G)$* every *derivation tree for w satisfies the OLP constraint.*
2. *A grammar G is OLP iff for every $w \in L(G)$ there* exists *a derivation tree which satisfies the OLP constraint.*

We begin the discussion with OLP constraints for skeletal grammars. One of the first definitions was suggested by Pereira and Warren (1983). Their constraint was designed for DCGs (a skeletal unification grammar formalism which assumes an explicit context-free backbone) for guaranteeing termination of general proof procedures of definite clause sets. Rephrased in terms of skeletal grammars, the definition is as follows:

Definition 3.2 (Pereira and Warren's OLP for skeletal grammars (OLP_{PW})). *A grammar is off-line parsable iff its context-free skeleton is not infinitely ambiguous.*

The *context-free skeleton* is obtained by ignoring all FSs of the grammar rules and considering only the categories. In Jaeger et al. (2002) we prove that the depth of every derivation tree generated by a grammar whose context-free skeleton is finitely ambiguous is bounded by the number of syntactic categories times the size of its yield, therefore the recognition problem is decidable.

Kaplan and Bresnan (1982) suggested a linguisticly motivated OLP constraint which refers to valid derivations for the lexical functional grammar formalism (LFG), a skeletal grammar formalism. They impose constraints on two kinds of ϵ's, optionality and controlled ϵ's, but as these terms are not formally defined, we use a variant of their constraint, suggested by Johnson (1988, 95-97), eliminating all ϵ's of any kind.

Definition 3.3 (Johnson's OLP (OLP_J)). *A constituent structure satisfies the off-line parsability constraint iff it does not include a non-branching dominance chain in which the same category appears twice and the empty string ϵ does not appear as a lexical form annotation of any (terminal) node.*

This constraint bounds the depth of any OLP derivation tree by a linear function of the size of its yield, thus ensuring decidability of the recognition problem.

Johnson's definition is a restriction on allowable c-structures rather than on the grammar itself. We use definition 3.1 for OLP_J grammars and refer only to its second part since it is less restrictive.

The next definition is also based on Kaplan and Bresnan's constraint and is also dealing only with OLP derivations. OLP grammar definitions are according to definition 3.1.

X-bar theory grammars (Chomsky, 1975) have a strong linguistic justification in describing natural languages. Unfortunately neither Kaplan and Bresnan's nor Johnson's constraints allow such grammars, since they do not allow derivation trees in which the same category appears twice in a non-branching dominance chain. Kuhn (1999) refers to the problem from a linguist's point of view. The purpose of his constraint was to expand the class of grammars which satisfy Kaplan and Bresnan's constraint in order to allow X-bar derivations. Again, since there exists no formal definition of the different kinds of ϵ's we assume that ϵ does not represent a lexical item (no ϵ-rules).

Definition 3.4 (Kuhn's OLP (OLP_K)). *A c-structure derivation is valid iff no category appears twice in a non-branching dominance chain with the same f-annotation.*

Kuhn (1999) gives some examples of X-bar theory derivation trees of German and Italian sentences which contain the same category twice in a non-branching dominance chain with a different f-annotation. Therefore they are invalid OLP derivation trees (by both Kaplan and Bresnan's and Johnson's constraints), but they satisfy Kuhn's OLP constraint.

According to Kuhn (1999), "The Off-line parsability condition is a restriction on allowable c-structures excluding that for a given string, infinitely many c-structure analyses are possible". In other words, Kuhn assumes that OLP is, in fact, a condition that is intended to guarantee finite ambiguity. Kuhn's definition may allow X-bar derivations, but it does not ensure finite ambiguity. The following grammar is an LFG grammar generating c-structures in which the same category appears twice in a non-branching dominance chain only with a different f-annotation, therefore it satisfies Kuhn's definition of OLP. But the grammar is infinitely ambiguous:

$$P \rightarrow \underset{(\uparrow subj = \downarrow)}{P} \qquad \mathcal{L}(b) = \{\langle P, (\uparrow pred) = \text{`}b'\rangle\}$$

$$\begin{array}{l} \mathrm{P}\ \left[subj : \left[subj : \ldots \left[pred : \text{`}b'\right] \ldots\right]\right] \\ \downarrow \\ \mathrm{P}\ \left[subj : \ldots \left[pred : \text{`}b'\right] \ldots\right] \\ \vdots \\ \downarrow \\ \mathrm{P}\ \left[pred : \text{`}b'\right] \end{array}$$

Therefore, it is not clear whether the condition guarantees parsing termination nor decidability of the recognition problem and we exclude Kuhn's definition from further analysis.

The following definitions are applicable to both skeletal and general unification grammars. The first constraint was suggested by Haas (1989). Based on the fact that not every natural unification grammar has an obvious context-free backbone, Haas suggested a constraint for guaranteeing solvability of the parsing problem which is applicable to all unification grammar formalisms.

Haas' definition of a derivation tree is slightly different from the definition given above (definition 2.5). He allows derivation trees with non-terminals at their leaves, therefore a tree may represent a partial derivation.

Definition 3.5 (Haas' Depth-boundedness (DB)). *A unification grammar is depth-bounded iff for every $L > 0$ there is a $D > 0$ such that every parse tree for a sentential form of L symbols has depth less than D.*

According to Haas (1989), "a depth-bounded grammar cannot build an unbounded amount of tree structure from a bounded number of symbols". Therefore, for each sentential form of length n there exist a finite number of partial derivation trees, guaranteeing parsing termination.

The OLP_{PW} definition applies only to skeletal grammars, general unification grammars do not necessarily yield an explicit context-free skeleton. But the definition can be extended for all unification grammar formalisms:

Definition 3.6 (Finite ambiguity for unification grammars (FA)). *A unification grammar G is OLP iff for every string w there exist a finite number of derivation trees.*

Shieber's OLP definition (Shieber, 1992, 79–82) is defined in terms of logical constraint based grammar formalisms. His constraint is defined in logical terms, such as models and operations on models. We reformulate the definition in terms of FSs.

Definition 3.7 (Shieber's OLP (OLP_S)). *A grammar G is off-line parsable iff there exists a finite-ranged function F on FSs such that $F(A) \sqsubseteq A$ for all A and there are no derivation trees admitted by G in which a node $\langle A \rangle$ dominates a node $\langle B \rangle$, both are roots of sub-trees with an identical yield and $F(A) = F(B)$.*

The constraint is intended to bound the depth of every derivation tree by the range of F times the size of its yield. Thus the recognition problem is decidable.

Johnson's OLP constraint is too restrictive, since it excludes all repetitive unary branching chains and ϵ- rules, furthermore, it is applicable only to skeletal grammars. Therefore, Torenvliet and Trautwein (1995) have suggested a more liberal constraint, which is applicable to all unification grammar formalisms.

Definition 3.8 (Honest parsability constraint (HP)). *A grammar G satisfies the Honest Parsability Constraint (HPC) iff there exists a polynomial p s.t. for each $w \in L(G)$ there exists a derivation with at most $p(|w|)$ steps.*

The definition guarantees that for every string of the grammar's language there exists at least one polynomial depth (in the size of the derived string) derivation tree. Furthermore, the definition allows X-bar theory derivation trees, since a category may appear twice in a non-branching dominance chain as long as the depth of the tree is bounded by a polynomial function of its yield.

4 OLP Analysis

In this section we first give some grammar examples and mention their OLP properties, then compare the different variants of OLP definitions using these examples. The examples use a straightforward encoding of lists as FSs, where an empty list is denoted by $\langle\,\rangle$, and $\langle \mathit{head} \mid \mathit{tail} \rangle$ represents a list whose first item is $head$, followed by $tail$.

Figure 1 lists an example unification grammar generating the language $\{b\}^+$. A string of N occurrences of b has exactly one parse tree and its depth is $2N$. Therefore, G_1 is FA and HP. G_1 is neither DB nor OLP_S; it may generate arbitrarily deep derivation trees (containing lists of increasing length) whose frontier consists of only one symbol, and thus there exists no finite-ranged function mapping each FS on such a derivation to a finite set of FSs.

$$A^s = \begin{bmatrix} \text{CAT} : s \\ \text{WORD} : \langle s \rangle \end{bmatrix}$$

$$\mathcal{R} = \left\{ \begin{array}{l} \begin{bmatrix} \text{CAT} : s \\ \text{WORD} : \langle s \rangle \end{bmatrix} \longrightarrow \begin{bmatrix} \text{CAT} : p \\ \text{WORD} : \langle t \rangle \end{bmatrix} \\ \begin{bmatrix} \text{CAT} : p \\ \text{WORD} : \boxed{1} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{CAT} : p \\ \text{WORD} : \langle t \mid \boxed{1} \rangle \end{bmatrix} \\ \begin{bmatrix} \text{CAT} : p \\ \text{WORD} : \boxed{1} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{CAT} : q \\ \text{WORD} : \boxed{1} \end{bmatrix} \\ \begin{bmatrix} \text{CAT} : q \\ \text{WORD} : \langle t \mid \boxed{1} \rangle \end{bmatrix} \longrightarrow \begin{bmatrix} \text{CAT} : q \\ \text{WORD} : \boxed{1} \end{bmatrix} \begin{bmatrix} \text{CAT} : q \\ \text{WORD} : \langle t \rangle \end{bmatrix} \end{array} \right\}$$

$$\mathcal{L}(b) = \left\{ \begin{bmatrix} \text{CAT} : q \\ \text{WORD} : \langle t \rangle \end{bmatrix} \right\}$$

Figure 1: A unification grammar, G_1, $L(G_1) = \{b\}^+$.

Figure 2 lists an example unification grammar generating the language $\{b\}$. There exist infinitely many derivation trees, of arbitrary depths, for the string b, therefore, G_2 is neither DB nor FA nor OLP_S. G_2 is HP; there exists a derivation tree for b of depth 2.

Figure 3 lists an example unification grammar generating the language $\{b\}^+$. A string of N occurrences of b has exactly one parse tree. The feature DEPTH represents the current depth of the derivation tree; at each derivation step an item is added to the DEPTH list. The feature TEMP represents the number of derivation steps before generating the next b symbol. Every application of the second rule doubles the depth of TEMP list (with respect to its length after the previous application of the rule). Thus the number of derivation steps for generating each b is always twice the number of steps for generating its predecessor, and for every sentential form of length N any partial derivation tree's depth is bounded by an exponential function of N (approximately 2^N). Therefore G_3 is FA and DB but neither OLP_S nor HP.

$$A^s = \begin{bmatrix} \text{CAT} : s \\ \text{WORD} : \langle s \rangle \end{bmatrix}$$

$$\mathcal{R} = \left\{ \begin{array}{l} \begin{bmatrix} \text{CAT} : s \\ \text{WORD} : \langle s \rangle \end{bmatrix} \longrightarrow \begin{bmatrix} \text{CAT} : p \\ \text{WORD} : \langle t \rangle \end{bmatrix} \\ \begin{bmatrix} \text{CAT} : p \\ \text{WORD} : \boxed{1} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{CAT} : p \\ \text{WORD} : \langle t \mid \boxed{1} \rangle \end{bmatrix} \\ \begin{bmatrix} \text{CAT} : p \\ \text{WORD} : \boxed{1} \end{bmatrix} \longrightarrow \begin{bmatrix} \text{CAT} : q \\ \text{WORD} : \boxed{1} \end{bmatrix} \\ \begin{bmatrix} \text{CAT} : q \\ \text{WORD} : \langle t \mid \boxed{1} \rangle \end{bmatrix} \longrightarrow \begin{bmatrix} \text{CAT} : q \\ \text{WORD} : \boxed{1} \end{bmatrix} \end{array} \right\}$$

$$\mathcal{L}(b) = \left\{ \begin{bmatrix} \text{CAT} : q \\ \text{WORD} : \langle t \rangle \end{bmatrix} \right\}$$

Figure 2: A unification grammar, G_2, $L(G_2) = \{b\}$.

$$A^s = \begin{bmatrix} \text{CAT} : s \\ \text{DEPTH} : \langle\,\rangle \\ \text{TEMP} : \langle\,\rangle \end{bmatrix}$$

$$\mathcal{R} = \left\{ \begin{array}{l} \begin{bmatrix} \text{CAT} : s \\ \text{DEPTH} : \langle\,\rangle \\ \text{TEMP} : \langle\,\rangle \end{bmatrix} \rightarrow \begin{bmatrix} \text{CAT} : p \\ \text{DEPTH} : \langle t \rangle \\ \text{TEMP} : \langle\,\rangle \end{bmatrix} \\ \begin{bmatrix} \text{CAT} : p \\ \text{DEPTH} : \boxed{1} \\ \text{TEMP} : \langle\,\rangle \end{bmatrix} \rightarrow \begin{bmatrix} \text{CAT} : p \\ \text{DEPTH} : \langle t \mid \boxed{1} \rangle \\ \text{TEMP} : \boxed{1} \end{bmatrix} \begin{bmatrix} \text{CAT} : l \\ \text{LEX} : t \end{bmatrix} \\ \begin{bmatrix} \text{CAT} : p \\ \text{DEPTH} : \boxed{1} \\ \text{TEMP} : \langle t \mid \boxed{2} \rangle \end{bmatrix} \rightarrow \begin{bmatrix} \text{CAT} : p \\ \text{DEPTH} : \langle t \mid \boxed{1} \rangle \\ \text{TEMP} : \boxed{2} \end{bmatrix} \\ \begin{bmatrix} \text{CAT} : p \\ \text{DEPTH} : [\,] \\ \text{TEMP} : \langle\,\rangle \end{bmatrix} \rightarrow \begin{bmatrix} \text{CAT} : l \\ \text{LEX} : t \end{bmatrix} \end{array} \right\}$$

$$\mathcal{L}(b) = \left\{ \begin{bmatrix} \text{CAT} : l \\ \text{LEX} : t \end{bmatrix} \right\}$$

Figure 3: A unification grammar, G_3, $L(G_3) = \{b\}^+$.

Inter-relations among the OLP definitions

Below we make a comparison of all given OLP definitions; such relationships were not investigated in the past. We begin by considering skeletal grammars.

Johnson's condition is the only one omitting all ϵ's, thus none of the others implies OLP_J.

OLP_J – HP: The depth of any OLP_J derivation tree is bounded by a linear function of its yield, therefore for every string there exists a derivation tree of at most a polynomial depth, and an OLP_J grammar is HP.

OLP_J – OLP_{PW},DB,FA,OLP_S: The grammar of figure 2 is an OLP_J grammar (viewing CAT as the category) but it does not satisfy the other constraints.

OLP_{PW} – DB,FA,OLP_S,HP: By Jaeger et al. (2002), the depth of any derivation tree (partial/non-partial) admitted by an OLP_{PW} grammar is bounded by a linear function of the size of its yield, thus an OLP_{PW} grammar satisfies all the other constraints. A grammar satisfying the constraints may still have an infinitely ambiguous context-free backbone.

We continue the analysis by comparing the definitions which are applicable to general unification grammars.

DB – FA: A DB grammar is also FA; it can only generate derivation trees whose depth is bounded by a function of their yield, and there exist only a finite number of derivation trees up to a certain depth. By figure 1, an FA grammar is not necessarily DB.

DB – OLP_S: None of the conditions implies the other. The grammar of figure 3 is DB but not OLP_S. A grammar whose language consists of only one word, and its derivation is of a constant depth, may still contain a redundant rule generating arbitrarily deep trees whose frontier is of length 1. Thus it is OLP_S but not DB.

DB,FA – HP: DB means that *every* derivation tree is bounded by some function of its yield. HP means that for every string there *exist* at least one derivation tree of a polynomial depth of its yield. The grammar of figure 3 is DB and FA, but since every derivation tree's depth is exponential in the size of its yield, it is not HP. The grammar of figure 2 is HP, but since it is infinitely ambiguous, it is neither FA nor DB.

FA,HP – OLP_S: The depth of any derivation tree admitted by an OLP_S grammar is bounded by a linear function of its yield. Thus an OLP_S grammar is FA and HP. By figure 1, an FA and HP grammar is not necessarily OLP_S.

Figure 4 depicts the inter-relations hierarchy diagram of the OLP definitions, separated for skeletal and general unification grammars. The arrows represent the implications discussed above.

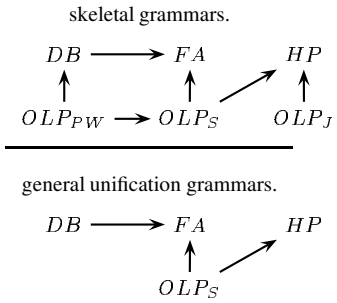

Figure 4: Inter-relations Hierarchy diagram.

5 Undecidability proofs

For the definitions which are applicable only to skeletal grammars it is easy to verify whether a grammar satisfies the constraint. The definitions that apply to arbitrary unification grammars are harder to check. In this section we give sketches of proofs of undecidability of three of the OLP definitions: Finite Ambiguity (FA), Depth-Boundedness (DB) and Shieber's OLP (OLP_S).

Theorem 1. *Finite ambiguity is undecidable.*

Proof sketch. *In order to show that finite ambiguity is undecidable, we use a reduction from the membership problem, which is known to be undecidable (Johnson, 1988). We assume that there exists an algorithm, A_{FA}, for deciding FA and show how it can be used to decide whether $w \in L(G)$.*

Given a string w and a grammar G, construct G', by adding the rule $A^s \rightarrow A^s$ to G's set of rules. Apply A_{FA} to G', G' is FA on w iff $w \notin L(G)$. If $w \in L(G)$ then $w \in L(G')$, therefore by applying the rule $A^s \rightarrow A^s$ infinitely many times, there exist an infinite number of derivation trees for w admitted by G'. If $w \notin L(G)$ then $w \notin L(G')$, no application of the additional rule would generate any derivation tree for w, and G' is finitely ambiguous.

Since the membership problem is undecidable, it is undecidable whether there exist only a finite number of derivation trees for a string w admitted by G. Hence finite ambiguity is undecidable.

Theorem 2. *Depth-boundedness is undecidable.*

Proof sketch. *In order to prove undecidability of depth-boundedness, we use a reduction from the Turing machines halting problem, which is known to be undecidable (Hopcroft and Ullman, 1979, 183-185). We assume that there exists an algorithm, A_{DB}, for deciding DB and show how it can be used to decide whether a Turing machine M terminates on the empty input ϵ.*

Johnson (1988) suggested a transformation from the Turing machines halting problem to unification grammars. The transformation generates a grammar, G_M, which consists of unit-rules only, and can generate at most one complete derivation tree. Assume the existence of an algorithm A_{DB}. Apply A_{DB} to G_M. If G_M is DB then the grammar generates a complete derivation tree, therefore its language is non empty and M terminates on the empty input. Otherwise, $L(G) = \emptyset$ and M does not terminate on the empty input. Thus, we have decided the Turing machines halting problem.

Theorem 3. *OLP_S is undecidable.*

Proof sketch. *In order to prove undecidability of OLP_S, we use a combination of the undecidability proofs of DB and FA. Given a Turing machine M, construct G_M using Johnson's reduction, then construct G'_M by adding $A^s \rightarrow A^s$ to G_M. Assume the existence of an algorithm A_S, deciding OLP_S. G'_M is OLP_S iff M does not terminate on the empty input. Thus, by applying A_S on G'_M, we have decided the Turing machines halting problem.*

6 Conclusions

In this paper we compare several variants of the OLP constraint for the first time. We give sketches of proofs of undecidability of three OLP conditions, full proofs along with undecidability proofs of other conditions are given in Jaeger et al. (2002). In Jaeger et al. (2002) we also give a novel OLP constraint as well as an algorithm for deciding whether a grammar satisfies it. The constraint is applicable to all unification grammar formalisms. It is more liberal than the existing constraints that are limited to skeletal grammars only, yet, unlike all definitions that are applicable to general unification grammars, it can be tested efficiently.

Acknowledgements

The work of Nissim Francez was partially funded by the vice-president's fund for the promotion of research at the Technion. The work of Shuly Wintner was supported by the Israeli Science Foundation (grant no. 136/1).

References

Bob Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press.

Noam Chomsky. 1975. Remarks on nominalization. In Donald Davidson and Gilbert H. Harman, editors, *The Logic of Grammar*, pages 262–289. Dickenson Publishing Co., Encino, California.

Nissim Francez and Shuly Wintner. In preperation. Feature structure based linguistic formalisms.

Andrew Haas. 1989. A parsing algorithm for unification grammar. *Computational Linguistics, 15(4):219–232*.

J. Hopcroft and J. Ullman. 1979. Introduction to automata theory languages and computation.

Efrat Jaeger, Nissim Francez, and Shuly Wintner. 2002. Unification grammars and off-line parsability. Technical report, Technion, Israel Institute of Technology.

Mark Johnson. 1988. *Attribute-Value Logic and the Theory of Grammar*. CSLI Lecture Notes. CSLI.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-functional grammar: A formal system for grammatical representation. *The MIT Press*, page 266.

Jonas Kuhn. 1999. Towards a simple architecture for the structure-function mapping. *Proceedings of the LFG99 Conference*.

Fernando C. N. Pereira and David H. D. Warren. 1983. Parsing as deduction. *Proceedings of ACL - 21*.

Stuart M. Shieber. 1992. Constraint-based grammar formalisms. *MIT Press*.

Leen Torenvliet and Marten Trautwein. 1995. A note on the complexity of restricted attribute-value grammars. *ILLC Research Report and Technical Notes Series CT-95-02, University of Amsterdam, Amsterdam*.

Shuly Wintner and Nissim Francez. 1999. Off-line parsability and the well-foundedness of subsumption. *Journal of Logic, Language and Information, 8(1):1-16*, January.