Highly constrained unification grammars

Daniel Feinstein
Department of Computer Science
University of Haifa
31905 Haifa, Israel
daniel@cs.haifa.ac.il

Shuly Wintner
Department of Computer Science
University of Haifa
31905 Haifa, Israel
shuly@cs.haifa.ac.il

Abstract

Unification grammars are widely accepted as an expressive means for describing the structure of natural languages. In general, the recognition problem is undecidable for unification grammars. Even with restricted variants of the formalism, *off-line parsable* grammars, the problem is computationally hard. We present two natural constraints on unification grammars which limit their expressivity. We first show that *non-reentrant* unification grammars generate exactly the class of context-free languages. We then relax the constraint and show that *one-reentrant* unification grammars generate exactly the class of tree-adjoining languages. We thus relate the commonly used and linguistically motivated formalism of unification grammars to more restricted, computationally tractable classes of languages.

1 Introduction

Unification grammars (UG) (Shieber, 1986; Shieber, 1992; Carpenter, 1992) have originated as an extension of context-free grammars, the basic idea being to augment the context-free rules with non context-free annotations (feature structures) in order to express additional information. They can describe phonological, morphological, syntactic and semantic properties of languages simultaneously and are thus linguistically suitable for modeling natural languages. Several formulations of unification grammars have been proposed, and they are used extensively by computational linguists to describe the structure of a variety of natural languages.

Unification grammars are Turing equivalent: determining whether a given string is generated by a given grammar is as hard as deciding whether a Turing machine halts on the empty input (Johnson, 1988). Therefore, the recognition problem for unification grammars is undecidable in the general case. To ensure its decidability, several constraints on unification grammars, commonly known as the *off-line parsability (OLP) constraints*, were suggested, such that the recognition problem is decidable for off-line parsable grammars (Jaeger et al., 2005). The idea behind all the OLP definitions is to rule out grammars which license trees in which unbounded amount of material is generated without expanding the frontier word. This can happen due to two kinds of rules: ϵ-rules (whose bodies are empty) and unit rules (whose bodies consist of a single element). However, even for unification grammars with no such rules the recognition problem is NP-hard (Barton et al., 1987).

In order for a grammar formalism to make predictions about the structure of natural language its generative capacity must be constrained. It is now generally accepted that Context-free Grammars (CFGs) lack the generative power needed for this purpose (Savitch et al., 1987), due to natural language constructions such as reduplication, multiple agreement and crossed agreement. Several linguistic formalisms have been proposed as capable of modeling these phenomena, including Linear Indexed Grammars (LIG) (Gazdar, 1988), Head Grammars (Pollard, 1984), Tree Adjoining Grammars (TAG) (Joshi, 2003) and Combinatory Categorial Grammars (Steedman, 2000). In a seminal work, Vijay-Shanker and Weir (1994) prove that all four formalisms are weakly equivalent. They all generate the class of *mildly context-sensitive languages* (MCSL), all members

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 1089–1096,
Sydney, July 2006.

of which have recognition algorithms with time complexity $O(n^6)$ (Vijay-Shanker and Weir, 1993; Satta, 1994).[1] As a result of the weak equivalence of four independently developed (and linguistically motivated) extensions of CFG, the class MCSL is considered to be linguistically meaningful, a natural class of languages for characterizing natural languages.

Several authors tried to approximate unification grammars by means of context-free grammars (Rayner et al., 2001; Kiefer and Krieger, 2004) and even finite-state grammars (Pereira and Wright, 1997; Johnson, 1998), but we are not aware of any work which relates unification grammars with the class MCSL. The main objective of this work is to define constraints on UGs which naturally limit their generative capacity. We define two natural and easily testable syntactic constraints on UGs which ensure that grammars satisfying them generate the context-free and the mildly context-sensitive languages, respectively. The contribution of this result is twofold:

- From a theoretical point of view, constraining unification grammars to generate exactly the class MCSL results in a grammatical formalism which is, on one hand, powerful enough for linguists to express linguistic generalizations in, and on the other hand cognitively adequate, in the sense that its generative capacity is constrained;
- Practically, such a constraint can provide efficient recognition algorithms for the limited class of unification grammars.

We define some preliminary notions in section 2 and then show a constrained version of UG which generates the class CFL of context-free languages in section 3. Section 4 presents the main result, namely a restricted version of UG and a mapping of its grammars to LIG, establishing the proposition that such grammars generate exactly the class MCSL. For lack of space, we favor intuitive explanation over rigorous proofs; the full details can be found in Feinstein (2004).

2 Preliminary notions

A CFG is a four-tuple $G^{cf} = \langle V_N, V_t, \mathcal{R}^{cf}, S\rangle$ where V_t is a set of *terminals*, V_N is a set of *non-terminals*, including the *start symbol* S, and $\mathcal{R}^{cf}$ is a set of productions, assumed to be in a normal form where each rule has either (zero or more) non-terminals or a single terminal in its body, and where the start symbol never occurs in the right hand side of rules. The set of all such context-free grammars is denoted CFGS.

In a linear indexed grammar (LIG),[2] strings are derived from nonterminals with an associated stack denoted $A[l_1 \dots l_n]$, where A is a nonterminal, each l_i is a stack symbol, and l_1 is the top of the stack. Since stacks can grow to be of unbounded size during a derivation, some way of partially specifying unbounded stacks in LIG productions is needed. We use $A[l_1 \dots l_n\ \infty]$ to denote the nonterminal A associated with any stack η whose top n symbols are $l_1, l_2 \dots, l_n$. The set of all nonterminals in V_N, associated with stacks whose symbols come from V_s, is denoted $V_N[V_s^*]$.

Definition 1. *A **Linear Indexed Grammar** is a five tuple $G^{li} = \langle V_N, V_t, V_s, \mathcal{R}^{li}, S\rangle$ where V_t, V_N and S are as above, V_s is a finite set of indices (stack symbols) and $\mathcal{R}^{li}$ is a finite set of productions in one of the following two forms:*

- ***fixed** stack: $N_i[p_1 \dots p_n] \rightarrow \alpha$*
- ***unbounded** stack: $N_i[p_1 \dots p_n\ \infty] \rightarrow \alpha$ or $N_i[p_1 \dots p_n\ \infty] \rightarrow \alpha N_j[q_1 \dots q_m\ \infty]\beta$*

where $N_i, N_j \in V_N$, $p_1 \dots p_n, q_1 \dots q_m \in V_s$, $n, m \geq 0$ and $\alpha, \beta \in (V_t \cup V_N[V_s^*])^*$.

A crucial characteristic of LIG is that only *one* copy of the stack can be copied to a *single* element in the body of a rule. If more than one copy were allowed, the expressive power would grow beyond MCSL.

Definition 2. *Given a LIG $\langle V_N, V_t, V_s, \mathcal{R}^{li}, S\rangle$, the **derivation relation** '$\Rightarrow_{li}$' is defined as follows: for all $\Psi_1, \Psi_2 \in (V_N[V_s^*] \cup V_t)^*$ and $\eta \in V_s^*$,*

- *If $N_i[p_1 \dots p_n] \rightarrow \alpha \in \mathcal{R}^{li}$ then*

$$\Psi_1 N_i[p_1 \dots p_n]\Psi_2 \Rightarrow_{li} \Psi_1 \alpha \Psi_2$$

- *If $N_i[p_1 \dots p_n\ \infty] \rightarrow \alpha \in \mathcal{R}^{li}$ then*

$$\Psi_1 N_i[p_1 \dots p_n \eta]\Psi_2 \Rightarrow_{li} \Psi_1 \alpha \Psi_2$$

- *If $N_i[p_1 \dots p_n\ \infty] \rightarrow \alpha N_j[q_1 \dots q_m\ \infty]\beta \in \mathcal{R}^{li}$ then $\Psi_1 N_i[p_1 \dots p_n \eta]\Psi_2 \Rightarrow_{li} \Psi_1 \alpha N_j[q_1 \dots q_m \eta]\beta\Psi_2$*

[1] The term *mildly context-sensitive* was coined by Joshi (1985), in reference to a less formally defined class of languages. Strictly speaking, what we call MCSL here is also known as the class of *tree-adjoining languages*.

[2] The definition is based on Vijay-Shanker and Weir (1994).

*The **language** generated by G^{li} is $L(G^{li}) = \{w \in V_t^* \mid S[\,] \stackrel{*}{\Rightarrow}_{li} w\}$, where '$\stackrel{*}{\Rightarrow}_{li}$' is the reflexive, transitive closure of '$\Rightarrow_{li}$'.*

Unification grammars are defined over *feature structures* (FSs) which are directed, connected, rooted, labeled graphs, usually depicted as *attribute-value matrices* (AVM). A feature structure A can be characterized by its set of paths, Π_A, an assignment of atomic values to the ends of some paths, $\Theta_A(\cdot)$, and a reentrancy relation '$\leftrightsquigarrow$' relating paths which lead to the same node. A sequence of feature structures, where some nodes may be shared by more than one element, is a *multi-rooted structure* (MRS).

Definition 3. ***Unification grammars** are defined over a signature consisting of a finite set* ATOMS *of atoms; a finite set* FEATS *of features and a finite set* WORDS *of words. A **unification grammar** is a tuple $G^u = \langle \mathcal{R}^u, \mathsf{A}^s, \mathcal{L} \rangle$ where $\mathcal{R}^u$ 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 word $w \in$ WORDS a finite set of feature structures, $\mathcal{L}(w)$, and A^s is a feature structure, the **start symbol**.*

Definition 4. *A unification grammar $\langle \mathcal{R}^u, \mathsf{A}^s, \mathcal{L} \rangle$ over the signature $\langle$ATOMS, FEATS, WORDS$\rangle$ is **non-reentrant** iff for any rule $r^u \in \mathcal{R}^u$, r^u is non-reentrant. It is **one-reentrant** iff for every rule $r^u \in \mathcal{R}^u$, r^u includes at most one reentrancy, between the head of the rule and some element of the body. Let UG_{nr}, UG_{1r} be the sets of all non-reentrant and one-reentrant unification grammars, respectively.*

Informally, a rule is non-reentrant if (on an AVM view) no reentrancy tags occur in it. When the rule is viewed as a (multi-rooted) graph, it is non-reentrant if the in-degree of all nodes is at most 1. A rule is one-reentrant if (on an AVM view) at most one reentrancy tag occurs in it, exactly twice: once in the head of the rule and once in an element of its body. When the rule is viewed as a (multi-rooted) graph, it is one-reentrant if the in-degree of all nodes is at most 1, with the exception of one node whose in-degree can be 2, provided that the only two distinct paths that lead to this node leave from the roots of the head of the rule and an element of the body.

FSs and MRSs are partially ordered by *subsumption*, denoted '$\sqsubseteq$'. The least upper bound with respect to subsumption is *unification*, denoted '$\sqcup$'. Unification is partial; when $\mathsf{A} \sqcup \mathsf{B}$ is undefined we say that the unification *fails* and denote it as $\mathsf{A} \sqcup \mathsf{B} = \top$. Unification is lifted to MRSs: given two MRSs σ and ρ, it is possible to unify the i-th element of σ with the j-th element of ρ. This operation, called *unification in context* and denoted $(\sigma, i) \sqcup (\rho, j)$, yields two modified variants of σ and ρ: (σ', ρ').

In unification grammars, *forms* are MRSs. A form $\sigma_A = \langle \mathsf{A}_1, \ldots, \mathsf{A}_k \rangle$ *immediately derives* another form $\sigma_B = \langle \mathsf{B}_1, \ldots, \mathsf{B}_m \rangle$ (denoted by $\sigma_A \stackrel{1}{\Rightarrow}_u \sigma_B$) iff there exists a rule $r^u \in \mathcal{R}^u$ of length n that licenses the derivation. The head of r^u is matched against some element A_i in σ_A using unification in context: $(\sigma_A, i) \sqcup (r^u, 0) = (\sigma'_A, r')$. If the unification does not fail, σ_B is obtained by replacing the i-th element of σ'_A with the body of r'. The reflexive transitive closure of '$\stackrel{1}{\Rightarrow}_u$' is denoted by '$\stackrel{*}{\Rightarrow}_u$'.

Definition 5. *The **language** of a unification grammar G^u is $L(G^u) = \{w_1 \cdots w_n \in \text{WORDS}^* \mid \mathsf{A}^s \stackrel{*}{\Rightarrow}_u \langle \mathsf{A}_1, \ldots, \mathsf{A}_n \rangle\}$, where $\mathsf{A}_i \in \mathcal{L}(w_i)$ for $1 \leq i \leq n$.*

3 Context-free unification grammars

We define a constraint on unification grammars which ensures that grammars satisfying it generate the class CFL. The constraint disallows *any* reentrancies in the rules of the grammar. When rules are non-reentrant, applying a rule implies that an exact copy of the body of the rule is inserted into the generated (sentential) form, not affecting neighboring elements of the form the rule is applied to. The only difference between rule application in UG_{nr} and the analog operation in CFGS is that the former requires unification whereas the latter only calls for identity check. This small difference does not affect the generative power of the formalisms, since unification can be pre-compiled in this simple case.

The trivial direction is to map a CFG to a non-reentrant unification grammar, since every CFG is, trivially, such a grammar (where terminal and non-terminal symbols are viewed as atomic feature structures). For the inverse direction, we define a mapping from UG_{nr} to CFGS. The non-terminals of the CFG in the image of the mapping are the set of all feature structures defined in the source UG.

Definition 6. *Let ug2cfg : $\mathrm{UG}_{nr} \mapsto$ CFGS be a **mapping of** UG_{nr} **to** CFGS, such that*

if $G^u = \langle \mathcal{R}^u, \mathsf{A}^s, \mathcal{L} \rangle$ *is over the signature* $\langle$ATOMS, FEATS, WORDS$\rangle$ *then ug2cfg*$(G^u) = \langle V_N, V_t, \mathcal{R}^{cf}, S^{cf} \rangle$*, where:*

- $V_N = \{\mathsf{A}_i \mid \mathsf{A}_0 \rightarrow \mathsf{A}_1 \ldots \mathsf{A}_n \in \mathcal{R}^u, i \geq 0\} \cup \{\mathsf{A} \mid \mathsf{A} \in \mathcal{L}(a), a \in$ ATOMS$\} \cup \{\mathsf{A}^s\}$. *V_N is the set of all the feature structures occurring in any of the rules or the lexicon of* G^u.
- $S^{cf} = \mathsf{A}^s$ • $V_t =$ WORDS
- $\mathcal{R}^{cf}$ *consists of the following rules:*
  1. *Let* $\mathsf{A}_0 \rightarrow \mathsf{A}_1 \ldots \mathsf{A}_n \in \mathcal{R}^u$ *and* $\mathsf{B} \in \mathcal{L}(b)$*. If for some* i*,* $1 \leq i \leq n$*,* $\mathsf{A}_i \sqcup \mathsf{B} \neq \top$*, then* $\mathsf{A}_i \rightarrow b \in \mathcal{R}^{cf}$
  2. *If* $\mathsf{A}_0 \rightarrow \mathsf{A}_1 \ldots \mathsf{A}_n \in \mathcal{R}^u$ *and* $\mathsf{A}^s \sqcup \mathsf{A}_0 \neq \top$ *then* $S^{cf} \rightarrow \mathsf{A}_1 \ldots \mathsf{A}_n \in \mathcal{R}^{cf}$.
  3. *Let* $r_1^u = \mathsf{A}_0 \rightarrow \mathsf{A}_1 \ldots \mathsf{A}_n$ *and* $r_2^u = \mathsf{B}_0 \rightarrow \mathsf{B}_1 \ldots \mathsf{B}_m$*, where* $r_1^u, r_2^u \in \mathcal{R}^u$*. If for some* i*,* $1 \leq i \leq n$*,* $\mathsf{A}_i \sqcup \mathsf{B}_0 \neq \top$*, then the rule* $\mathsf{A}_i \rightarrow \mathsf{B}_1 \ldots \mathsf{B}_m \in \mathcal{R}^{cf}$

The size of *ug2cfg*(G^u) is polynomial in the size of G^u. By inductions on the lengths of the derivation sequences, we prove the following theorem:

Theorem 1. *If* $G^u = \langle \mathcal{R}^u, \mathsf{A}^s, \mathcal{L} \rangle$ *is a non-reentrant unification grammar and* $G^{cf} =$ *ug2cfg*(G^u)*, then* $L(G^{cf}) = L(G^u)$.

Corollary 2. *Non-reentrant unification grammars are weakly equivalent to* CFGS.

4 Mildly context-sensitive UG

In this section we show that *one-reentrant unification grammars* generate exactly the class MCSL. In such grammars each rule can have at most one reentrancy, reflecting the LIG situation where stacks can be copied to exactly one daughter in each rule.

4.1 Mapping LIG to UG_{1r}

In order to simulate a given LIG with a unification grammar, a dedicated signature is defined based on the parameters of the LIG.

Definition 7. *Given a LIG* $\langle V_N, V_t, V_s, \mathcal{R}^{li}, S \rangle$*, let* τ *be* $\langle$ATOMS, FEATS, WORDS$\rangle$*, where* ATOMS $= V_N \cup V_s \cup \{elist\}$, FEATS $= \{$HEAD, TAIL$\}$*, and* WORDS $= V_t$.

We use τ throughout this section as the signature over which UGs are defined. We use FSs over the signature τ to represent and simulate LIG symbols. In particular, FSs will encode lists in the natural way, hence the features HEAD and TAIL. For the sake of brevity, we use standard list notation when FSs encode lists. LIG symbols are mapped to FSs thus:

Definition 8. *Let toFs be a mapping of LIG symbols to feature structures, such that:*

1. *If* $t \in V_t$ *then toFs*$(t) = \langle t \rangle$
2. *If* $N \in V_N$ *and* $p_i \in V_s, 1 \leq i \leq n$*, then toFs*$(N[p_1, \ldots, p_n]) = \langle N, p_1, \ldots, p_n \rangle$

The mapping *toFs* is extended to sequences of symbols by setting *toFs*$(\alpha\beta) =$ *toFs*(α)*toFs*(β). Note that *toFs* is one to one.

When FSs that are images of LIG symbols are concerned, unification is reduced to identity:

Lemma 3. *Let* $X_1, X_2 \in V_N[V_s^*] \cup V_t$*. If* *toFs*$(X_1) \sqcup$ *toFs*$(X_2) \neq \top$ *then* *toFs*$(X_1) =$ *toFs*(X_2).

When a feature structure which is represented as an unbounded list (a list that is not terminated by $elist$) is unifiable with an image of a LIG symbol, the former is a prefix of the latter.

Lemma 4. *Let* $\mathsf{C} = \langle p_1, \ldots, p_n, \boxed{i} \rangle$ *be a non-reentrant feature structure, where* $p_1, \ldots, p_n \in V_s$*, and let*$X \in V_N[V_s^*] \cup V_t$*. Then* $\mathsf{C} \sqcup$*toFs*$(X) \neq \top$ *iff* *toFs*$(X) = \langle p_1, \ldots, p_n, \alpha \rangle$*, for some* $\alpha \in V_s^*$.

To simulate LIGs with UGs we represent each symbol in the LIG as a feature structure, encoding the stack of LIG non-terminals as lists. Rules that propagate stacks (from mother to daughter) are simulated by means of reentrancy in the UG.

Definition 9. *Let lig2ug be a mapping of* LIGS *to* UG_{1r}*, such that if* $G^{li} = \langle V_N, V_t, V_s, \mathcal{R}^{li}, S \rangle$ *and* $G^u = \langle \mathcal{R}^u, \mathsf{A}^s, \mathcal{L} \rangle =$ *lig2ug*(G^{li}) *then* G^u *is over the signature* τ *(definition 7),* $\mathsf{A}^s =$ *toFs*$(S[\,])$*, for all* $t \in V_t$*,* $\mathcal{L}(t) = \{$*toFs*$(t)\}$ *and* $\mathcal{R}^u$ *is defined by:*

- *A LIG rule of the form* $X_0 \rightarrow \alpha$ *is mapped to the unification rule* *toFs*$(X_0) \rightarrow$ *toFs*(α)
- *A LIG rule of the form* $N_i[p_1, \ldots, p_n\ \infty] \rightarrow \alpha\ N_j[q_1, \ldots, q_m\ \infty]\ \beta$ *is mapped to the unification rule* $\langle N_i, p_1, \ldots, p_n, \boxed{1} \rangle \rightarrow$ *toFs*$(\alpha)\ \langle N_j, q_1, \ldots, q_m, \boxed{1} \rangle$ *toFs*(β)

Evidently, *lig2ug*$(G^{li}) \in \mathrm{UG}_{1r}$ for any LIG G^{li}.

Theorem 5. *If $G^{li} = \langle V_N, V_t, V_s, \mathcal{R}^{li}, S^{li}\rangle$ is a LIG and $G^u = lig2ug(G^{li})$ then $L(G^u) = L(G^{li})$.*

4.2 Mapping UG_{1r} to LIG

We are now interested in the reverse direction, namely mapping UGs to LIG. Of course, since UGs are more expressive than LIGs, only a subset of the former can be correctly simulated by the latter. The differences between the two formalisms can be summarized along three dimensions:

The basic elements UG manipulates feature structures, and rules (and forms) are MRSs; whereas LIG manipulates terminals and non-terminals with stacks of elements, and rules (and forms) are sequences of such symbols.

Rule application In UG a rule is applied by *unification in context* of the rule and a sentential form, both of which are MRSs, whereas in LIG, the head of a rule and the selected element of a sentential form must have the same non-terminal symbol and consistent stacks.

Propagation of information in rules In UG information is shared through reentrancies, whereas In LIG, information is propagated by copying the stack from the head of the rule to one element of its body.

We show that one-reentrant UGs can all be correctly mapped to LIG. For the rest of this section we fix a signature $\langle \textsc{Atoms}, \textsc{Feats}, \textsc{Words}\rangle$ over which UGs are defined. Let NRFSS be the set of all non-reentrant FSs over this signature.

One-reentrant UGs induce highly constrained (sentential) forms: in such forms, there are no reentrancies whatsoever, neither between distinct elements nor within a single element. Hence all the FSs in forms induced by a one-reentrant UG are non-reentrant.

Definition 10. *Let A be a feature structure with no reentrancies. The **height** of A, denoted $|\mathsf{A}|$, is the length of the longest path in A. This is well-defined since non-reentrant feature structures are acyclic. Let $G^u = \langle \mathcal{R}^u, \mathsf{A}^s, \mathcal{L}\rangle \in \mathrm{UG}_{1r}$ be a one-reentrant unification grammar. The **maximum height** of the grammar, $maxHt(G^u)$, is the height of the highest feature structure in the grammar. This is well defined since all the feature structures of one-reentrant grammars are non-reentrant.*

The following lemma indicates an important property of one-reentrant UGs. Informally, in any FS that is an element of a sentential form induced by such grammars, if two paths are long (specifically, longer than the maximum height of the grammar), they must have a long common prefix.

Lemma 6. *Let $G^u = \langle \mathcal{R}^u, \mathsf{A}^s, \mathcal{L}\rangle \in \mathrm{UG}_{1r}$ be a one-reentrant unification grammar. Let A be an element of a sentential form induced by G^u. If $\pi \cdot \langle \mathrm{F}_j\rangle \cdot \pi_1, \pi \cdot \langle \mathrm{F}_k\rangle \cdot \pi_2 \in \Pi_A$, where $\mathrm{F}_j, \mathrm{F}_k \in \textsc{Feats}$, $j \neq k$ and $|\pi_1| \leq |\pi_2|$, then $|\pi_1| \leq maxHt(G^u)$.*

Lemma 6 facilitates a view of all the FSs induced by such a grammar as (unboundedly long) lists of elements drawn from a finite, predefined set. The set consists of all features in FEATS and all the non-reentrant feature structures whose height is limited by the maximal height of the unification grammar. Note that even with one-reentrant UGs, feature structures can be unboundedly deep. What lemma 6 establishes is that if a feature structure induced by a one-reentrant unification grammar is deep, then it can be represented as a *single* "core" path which is long, and all the sub-structures which "hang" from this core are depth-bounded. We use this property to encode such feature structures as *cords*.

Definition 11. *Let $\Psi : \textsc{NrFss} \times \textsc{Paths} \mapsto (\textsc{Feats} \cup \textsc{NrFss})^*$ be a mapping such that if A is a non-reentrant FS and $\pi = \langle \mathrm{F}_1, \ldots, \mathrm{F}_n\rangle \in \Pi_A$, then the **cord** $\Psi(\mathsf{A}, \pi)$ is $\langle \mathsf{A}_1, \mathrm{F}_1, \ldots, \mathsf{A}_n, \mathrm{F}_n, \mathsf{A}_{n+1}\rangle$, where for $1 \leq i \leq n+1$, A_i are non-reentrant FSs such that:*

- $\Pi_{A_i} = \{\langle \mathrm{G}\rangle \cdot \pi \mid \langle \mathrm{F}_1, \ldots, \mathrm{F}_{i-1}, \mathrm{G}\rangle \cdot \pi \in \Pi_A, i \leq n, \mathrm{G} \neq \mathrm{F}_i\} \cup \{\varepsilon\}$
- $\Theta_{A_i}(\pi) = \Theta_A(\langle \mathrm{F}_1, \ldots, \mathrm{F}_{i-1}\rangle \cdot \pi)$ *(if it is defined).*

*We also define $last(\Psi(\mathsf{A}, \pi)) = \mathsf{A}_{n+1}$. The **height** of a cord is defined as $|\Psi(\mathsf{A}, \pi)| = \max_{1\leq i\leq n+1}(|\mathsf{A}_i|)$. For each cord $\Psi(\mathsf{A}, \pi)$ we refer to A as the **base feature structure** and to π as the **base path**. The **length** of a cord is the length of the base path.*

The function Ψ is one to one: given $\Psi(\mathsf{A}, \pi)$, both A and π are uniquely determined.

Lemma 7. *Let G^u be a one-reentrant unification grammar and let A be an element of a sentential form induced by G^u. Then there is a path $\pi \in \Pi_A$ such that $|\Psi(\mathsf{A}, \pi)| < maxHt(G^u)$.*

Lemma 7 implies that every non-reentrant FS (i.e., FSs induced by one-reentrant grammars) can be represented as a height-limited cord. This mapping resolves the first difference between LIG and UG, by providing a representation of the *basic elements*. We use cords as the stack contents of LIG non-terminals: cords can be unboundedly long, but so can LIG stacks; the crucial point is that cords are height limited, implying that they can be represented using a *finite* number of elements.

We now show how to simulate, in LIG, the unification in context of a rule and a sentential form. The first step is to have exactly one non-terminal symbol (in addition to the start symbol); when all non-terminal symbols are identical, only the content of the stack has to be taken into account. Recall that in order for a LIG rule to be applicable to a sentential form, the stack of the rule's head must be a *prefix* of the stack of the selected element in the form. The only question is whether the two stacks are equal (fixed rule head) or not (unbounded rule head). Since the contents of stacks are cords, we need a property relating two cords, on one hand, with unifiability of their base feature structures, on the other. Lemma 8 establishes such a property. Informally, if the base path of one cord is a prefix of the base path of the other cord and all feature structures along the common path of both cords are unifiable, then the base feature structures of both cords are unifiable. The reverse direction also holds.

Lemma 8. *Let* $\mathsf{A}, \mathsf{B} \in$ NRFSS *be non-reentrant feature structures and* $\pi_1, \pi_2 \in$ PATHS *be paths such that* $\pi_1 \in \Pi_B$, $\pi_1 \cdot \pi_2 \in \Pi_A$, $\Psi(\mathsf{A}, \pi_1 \cdot \pi_2) = \langle \mathsf{t}_1, \mathrm{F}_1, \ldots, \mathrm{F}_{|\pi_1|}, \mathsf{t}_{|\pi_1|+1}, \mathrm{F}_{|\pi_1|+1}, \ldots, \mathsf{t}_{|\pi_1 \cdot \pi_2|+1} \rangle$, $\Psi(\mathsf{B}, \pi_1) = \langle \mathsf{s}_1, \mathrm{F}_1, \ldots, \mathsf{s}_{|\pi_1|+1} \rangle$, *and* $\langle \mathrm{F}_{|\pi_1|+1} \rangle \notin \Pi_{s_{|\pi_1|+1}}$. *Then* $\mathsf{A} \sqcup \mathsf{B} \neq \top$ *iff for all* i, $1 \leq i \leq |\pi_1| + 1$, $\mathsf{s}_i \sqcup \mathsf{t}_i \neq \top$.

The length of a cord of an element of a sentential form induced by the grammar cannot be bounded, but the length of any cord representation of a rule head is limited by the grammar height. By lemma 8, unifiability of two feature structures can be reduced to a comparison of two cords representing them and only the prefix of the longer cord (as long as the shorter cord) affects the result. Since the cord representation of any grammar rule's head is limited by the height of the grammar we always choose it as the shorter cord in the comparison.

We now define, for a feature structure C (which is a head of a rule) and some path π, the set that includes all feature structures that are both unifiable with C and can be represented as a cord whose height is limited by the grammar height and whose base path is π. We call this set the *compatibility set* of C and π and use it to define the set of all possible prefixes of cords whose base FSs are unifiable with C (see definition 13). Crucially, the compatibility set of C is finite for any feature structure C since the heights and the lengths of the cords are limited.

Definition 12. *Given a non-reentrant feature structure* C*, a path* $\pi = \langle \mathrm{F}_1, \ldots, \mathrm{F}_n \rangle \in \Pi_C$ *and a natural number* h*, the* ***compatibility set****,* $\Gamma(\mathsf{C}, \pi, h)$*, is defined as the set of all feature structures* A *such that* $\mathsf{C} \sqcup \mathsf{A} \neq \top$*,* $\pi \in \Pi_A$*, and* $|\Psi(\mathsf{A}, \pi)| \leq h$*.*

The compatibility set is defined for a feature structure and a given path (when h is taken to be the grammar height). We now define two similar sets, FH and UH, for a given FS, independently of a path. When rules of a one-reentrant unification grammar are mapped to LIG rules (definition 14), FH and UH are used to define heads of fixed and unbounded LIG rules, respectively. A single unification rule is mapped to a *set* of LIG rules, each with a different head. The stack of the head is some member of the sets FH and UH. Each such member is a prefix of the stack of potential elements of sentential forms that the LIG rule can be applied to.

Definition 13. *Let* C *be a non-reentrant feature structure and* h *be a natural number. Then:*

$$\mathrm{FH}(\mathsf{C}, h) = \{\Psi(\mathsf{A}, \pi) \mid \pi \in \Pi_C, \mathsf{A} \in \Gamma(\mathsf{C}, \pi, h)\}$$
$$\mathrm{UH}(\mathsf{C}, h) = \{\Psi(\mathsf{A}, \pi) \cdot \langle \mathrm{F} \rangle \mid \Psi(\mathsf{A}, \pi) \in \mathrm{FH}(\mathsf{C}, h), \Theta_C(\pi) \uparrow, \mathrm{F} \in \mathrm{FEATS}, val(last(\Psi(\mathsf{C} \sqcup \mathsf{A}, \pi)), \langle \mathrm{F} \rangle) \uparrow\}$$

This accounts for the second difference between LIG and one-reentrant UG, namely *rule application*. We now briefly illustrate our account of the last difference, *propagation of information in rules*. In UG_{1r} information is shared between the rule's head and a single element in its body. Let $r^u = \langle \mathsf{C}_0, \ldots, \mathsf{C}_n \rangle$ be a reentrant unification rule in which the path μ_e, leaving the e-th element of the body, is reentrant with the path μ_0 leaving the head. This rule is mapped to a *set* of LIG rules, corresponding to the possible rule heads induced by the compatibility set of C_0. Let r be a member of this set, and let X_0 and X_e be the head and the e-th element of r, respectively. Reentrancy in r^u is modeled in the LIG rule by copying the stack from X_0 to X_e. The major complication is the contents

of this stack, which varies according to the cord representations of C_0 and C_e and to the reentrant paths.

Summing up, in a LIG simulating a one-reentrant UG, FSs are represented as stacks of symbols. The set of stack symbols V_s, therefore, is defined as a set of height bounded non-reentrant FSs. Also, all the features of the UG are stack symbols. V_s is finite due to the restriction on FSs (no reentrancies and height-boundedness). The set of terminals, V_t, is the words of the UG. There are exactly two non-terminal symbols, S (the start symbol) and N.

The set of rules is divided to four. The *start rule* only applies once in a derivation, simulating the situation in UGs of a rule whose head is unifiable with the start symbol. *Terminal rules* are a straight-forward implementation of the lexicon in terms of LIG. *Non-reentrant rules* are simulated in a similar way to how rules of a non-reentrant UG are simulated by CFG (section 3). The major difference is the head of the rule, X_0, which is defined as explained above. *One-reentrant rules* are simulated similarly to non-reentrant ones, the only difference being the selected element of the rule body, X_e, which is defined as follows.

Definition 14. *Let ug2lig be a* ***mapping of*** UG_{1r} ***to*** LIGS*, such that if* $G^u = \langle \mathcal{R}^u, \mathsf{A}^s, \mathcal{L} \rangle \in \mathrm{UG}_{1r}$ *then* $\mathit{ug2lig}(G^u) = \langle V_N, V_t, V_s, \mathcal{R}^{li}, S \rangle$*, where* $V_N = \{N, S\}$ *(fresh symbols),* $V_t =$ WORDS*,* $V_s =$ FEATS $\cup \{\mathsf{A} \mid \mathsf{A} \in \text{NRFSS}, |\mathsf{A}| \leq \mathit{maxHt}(G^u)\}$*, and* $\mathcal{R}^{li}$ *is defined as follows:*[3]

1. $S[\,] \rightarrow N[\Psi(\mathsf{A}^s, \varepsilon)]$
2. *For every* $w \in$ WORDS *such that* $\mathcal{L}(w) = \{\mathsf{C}_0\}$ *and for every* $\pi_0 \in \Pi_{\mathsf{C}_0}$*, the rule* $N[\Psi(\mathsf{C}_0, \pi_0)] \rightarrow w$ *is in* $\mathcal{R}^{li}$*.*
3. *If* $\langle \mathsf{C}_0, \ldots, \mathsf{C}_n \rangle \in \mathcal{R}^u$ *is a non-reentrant rule, then for every* $X_0 \in \text{LIGHEAD}(\mathsf{C}_0)$ *the rule* $X_0 \rightarrow N[\Psi(\mathsf{C}_1, \varepsilon)] \ldots N[\Psi(\mathsf{C}_n, \varepsilon)]$ *is in* $\mathcal{R}^{li}$*.*
4. *Let* $r^u = \langle \mathsf{C}_0, \ldots, \mathsf{C}_n \rangle \in \mathcal{R}^u$ *and* $(0, \mu_0) \overset{r^u}{\rightsquigarrow} (e, \mu_e)$*, where* $1 \leq e \leq n$*. Then for every* $X_0 \in \text{LIGHEAD}(\mathsf{C}_0)$ *the rule*

$$X_0 \rightarrow N[\Psi(\mathsf{C}_1, \varepsilon)] \ldots N[\Psi(\mathsf{C}_{e-1}, \varepsilon)] \; X_e \; N[\Psi(\mathsf{C}_{e+1}, \varepsilon)] \ldots N[\Psi(\mathsf{C}_n, \varepsilon)]$$

is in $\mathcal{R}^{li}$*, where* X_e *is defined as follows. Let* π_0 *be the base path of* X_0 *and* A *be the base feature structure of* X_0*. Applying the rule* r^u *to* A*, define* $(\langle \mathsf{A} \rangle, 0) \sqcup (r^u, 0) = (\langle \mathsf{P}_0 \rangle, \langle \mathsf{P}_0, \ldots, \mathsf{P}_e, \ldots, \mathsf{P}_n \rangle)$*.*

   (a) *If* μ_0 *is not a prefix of* π_0 *then* $X_e = N[\Psi(\mathsf{P}_e, \mu_e)]$*.*

   (b) *If* $\pi_0 = \mu_0 \cdot \nu$*,* $\nu \in$ PATHS *then*

      i. *If* $X_0 = N[\Psi(\mathsf{A}, \pi_0)]$ *then* $X_e = N[\Psi(\mathsf{P}_e, \mu_e \cdot \nu)]$*.*

      ii. *If* $X_0 = N[\Psi(\mathsf{A}, \pi_0), \text{F } \infty]$ *then* $X_e = N[\Psi(\mathsf{P}_e, \mu_e \cdot \nu), \text{F } \infty]$*.*

By inductions on the lengths of the derivations we prove that the mapping is correct:

Theorem 9. *If* $G^u \in \mathrm{UG}_{1r}$*, then* $L(G^u) = L(\mathit{ug2lig}(G^u))$*.*

5 Conclusions

The main contribution of this work is the definition of two constraints on unification grammars which dramatically limit their expressivity. We prove that non-reentrant unification grammars generate exactly the class of context-free languages; and that one-reentrant unification grammars generate exactly the class of mildly context-sensitive languages. We thus obtain two linguistically plausible constrained formalisms whose computational processing is tractable.

This main result is primarily a formal grammar result. However, we maintain that it can be easily adapted such that its consequences to (practical) computational linguistics are more evident. The motivation behind this observation is that reentrancy only adds to the expressivity of a grammar formalism when it is potentially *unbounded*, i.e., when infinitely many feature structures can be the possible values at the end of the reentrant paths. It is therefore possible to modestly extend the class of unification grammars which can be shown to generate exactly the class of mildly context-sensitive languages, by allowing also a limited form of multiple reentrancies among the elements in a rule (e.g., to handle agreement phenomena). This can be most useful for grammar writers, and at the same time adds nothing to the expressivity of the formalism. We leave the formal details of such an extension to future work.

This work can also be extended in other directions. The mapping of one-reentrant UGs to LIG is highly verbose, resulting in LIGs with a huge

[3] For a non-reentrant FS C_0, we define: $\text{LIGHEAD}(\mathsf{C}_0)$ as $\{N[\eta] \mid \eta \in \text{FH}(\mathsf{C}_0, \mathit{maxHt}(G^u))\} \cup \{N[\eta\ \infty] \mid \eta \in \text{UH}(\mathsf{C}_0, \mathit{maxHt}(G^u))\}$

number of rules. We believe that it should be possible to optimize the mapping such that much smaller grammars are generated. In particular, we are looking into mappings of one-reentrant UGs to other MCSL formalisms, notably TAG.

The two constraints on unification grammars (non-reentrant and one-reentrant) are parallel to the first two classes of the Weir (1992) hierarchy of languages. A possible extension of this work could be a definition of constraints on unification grammars that would generate all the classes of the hierarchy. Another direction is an extension of one-reentrant unification grammars, where the reentrancy does not have to be between the head and one element of the body. Also of interest are two-reentrant unification grammars, possibly with limited kinds of reentrancies.

Acknowledgments

This research was supported by The Israel Science Foundation (grant no. 136/01). We are grateful to Yael Cohen-Sygal, Nissim Francez and James Rogers for their comments and help.

References

G. Edward Barton, Jr., Robert C. Berwick, and Eric Sven Ristad. 1987. The complexity of LFG. In G. Edward Barton, Jr., Robert C. Berwick, and Eric Sven Ristad, editors, *Computational Complexity and Natural Language*, Computational Models of Cognition and Perception, chapter 3, pages 89–102. MIT Press, Cambridge, MA.

Bob Carpenter. 1992. *The Logic of Typed Feature Structures*. Cambridge University Press.

Daniel Feinstein. 2004. Computational investigation of unification grammars. Master's thesis, University of Haifa.

Gerald Gazdar. 1988. Applicability of indexed grammars to natural languages. In Uwe Reyle and Christian Rohrer, editors, *Natural Language Parsing and Linguistic Theories*, pages 69–94. Reidel.

Efrat Jaeger, Nissim Francez, and Shuly Wintner. 2005. Unification grammars and off-line parsability. *Journal of Logic, Language and Information*, 14(2):199–234.

Mark Johnson. 1988. *Attribute-Value Logic and the Theory of Grammar*, volume 16 of *CSLI Lecture Notes*. CSLI, Stanford, California.

Mark Johnson. 1998. Finite-state approximation of constraint-based grammars using left-corner grammar transforms. In *Proceedings of the 17th international conference on Computational linguistics*, pages 619–623.

Aravind K. Joshi. 1985. Tree Adjoining Grammars: How much context Sensitivity is required to provide a reasonable structural description. In D. Dowty, I. Karttunen, and A. Zwicky, editors, *Natural Language Parsing*, pages 206–250. Cambridge University Press, Cambridge, U.K.

Aravind K. Joshi. 2003. Tree-adjoining grammars. In Ruslan Mitkov, editor, *The Oxford handbook of computational linguistics*, chapter 26, pages 483–500. Oxford university Press.

Bernd Kiefer and Hans-Ulrich Krieger. 2004. A context-free superset approximation of unification-based grammars. In Harry Bunt, John Carroll, and Giorgio Satta, editors, *New Developments in Parsing Technology*, pages 229–250. Kluwer Academic Publishers.

Fernando C. N. Pereira and Rebecca N. Wright. 1997. Finite-state approximation of phrase-structure grammars. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*, Language, Speech and Communication, chapter 5, pages 149–174. MIT Press, Cambridge, MA.

Carl Pollard. 1984. *Generalized phrase structure grammars, head grammars and natural language*. Ph.D. thesis, Stanford University.

Manny Rayner, John Dowding, and Beth Ann Hockey. 2001. A baseline method for compiling typed unification grammars into context free language models. In *Proceedings of EUROSPEECH 2001*, Aalborg, Denmark.

Giorgio Satta. 1994. Tree-adjoining grammar parsing and boolean matrix multiplication. In *Proceedings of the 20st Annual Meeting of the Association for Computational Linguistics*, volume 20.

Walter J. Savitch, Emmon Bach, William Marsh, and Gila Safran-Naveh, editors. 1987. *The formal complexity of natural language*, volume 33 of *Studies in Linguistics and Philosophy*. D. Reidel, Dordrecht.

Stuart M. Shieber. 1986. *An Introduction to Unification Based Approaches to Grammar*. Number 4 in CSLI Lecture Notes. CSLI.

Stuart M. Shieber. 1992. *Constraint-Based Grammar Formalisms*. MIT Press, Cambridge, Mass.

Mark Steedman. 2000. *The Syntactic Process*. Language, Speech and Communication. The MIT Press, Cambridge, Mass.

K. Vijay-Shanker and David J. Weir. 1993. Parsing some constrained grammar formalisms. *Computational Linguistics*, 19(4):591 – 636.

K. Vijay-Shanker and David J. Weir. 1994. The equivalence of four extensions of context-free grammars. *Mathematical systems theory*, 27:511–545.

David J. Weir. 1992. A geometric hierarchy beyond context-free languages. *Theoretical Computer Science*, 104:235–261.