RECOGNITION OF LINEAR CONTEXT-FREE REWRITING SYSTEMS*

Giorgio Satta
Institute for Research in Cognitive Science
University of Pennsylvania
Philadelphia, PA 19104-6228, USA
gsatta@linc.cis.upenn.edu

ABSTRACT

The class of linear context-free rewriting systems has been introduced as a generalization of a class of grammar formalisms known as mildly context-sensitive. The recognition problem for linear context-free rewriting languages is studied at length here, presenting evidence that, even in some restricted cases, it cannot be solved efficiently. This entails the existence of a gap between, for example, tree adjoining languages and the subclass of linear context-free rewriting languages that generalizes the former class; such a gap is attributed to "crossing configurations". A few other interesting consequences of the main result are discussed, that concern the recognition problem for linear context-free rewriting languages.

1 INTRODUCTION

Beginning with the late 70's, there has been a considerable interest within the computational linguistics field for rewriting systems that enlarge the generative power of context-free grammars (CFG) both from the weak and the strong perspective, still remaining far below the power of the class of context-sensitive grammars (CSG). The denomination of mildly context-sensitive (MCS) has been proposed for the class of the studied systems (see [Joshi *et al.*, 1991] for discussion). The rather surprising fact that many of these systems have been shown to be weakly equivalent has led researchers to generalize the elementary operations involved in only apparently different formalisms, with the aim of capturing the underlying similarities. The most remarkable attempts in such a direction are found in [Vijay-Shanker *et al.*, 1987] and [Weir, 1988] with the introduction of linear context-free rewriting systems (LCFRS) and in [Kasami *et al.*, 1987] and [Seki *et al.*, 1989] with the definition of multiple context-free grammars (MCFG); both these classes have been inspired by the much more powerful class of generalized context-free grammars (GCFG; see [Pollard, 1984]). In the definition of these classes, the generalization goal has been combined with few theoretically motivated constraints, among which the requirement of efficient parsability; this paper is concerned with such a requirement. We show that from the perpective of efficient parsability, a gap is still found between MCS and some subclasses of LCFRS.

More precisely, the class of LCFRS is carefully studied along two interesting dimensions, to be precisely defined in the following: a) the fan-out of the grammar and b) the production length. From previous work (see [Vijay-Shanker *et al.*, 1987]) we know that the recognition problem for LCFRS is in P when both dimensions are bounded.[1] We complete the picture by observing NP-hardness for all the three remaining cases. If P$\neq$NP, our result reveals an undesired dissimilarity between well known formalisms like TAG, HG, LIG and others for which the recognition problem is known to be in P (see [Vijay-Shanker, 1987] and [Vijay-Shanker and Weir, 1992]) and the subclass of LCFRS that is intended to generalize these formalisms. We investigate the source of the suspected additional complexity and derive some other practical consequences from the obtained results.

*I am indebted to Anuj Dawar, Shyam Kapur and Owen Rambow for technical discussion on this work. I am also grateful to Aravind Joshi for his support in this research. None of these people is responsible for any error in this work. This research was partially funded by the following grants: ARO grant DAAL 03-89-C-0031, DARPA grant N00014-90-J-1863, NSF grant IRI 90-16592 and Ben Franklin grant 91S.3078C-1.

[1] P is the class of all languages decidable in deterministic polynomial time; NP is the class of all languages decidable in nondeterministic polynomial time.

2 TECHNICAL RESULTS

This section presents two technical results that are the most important in this paper. A full discussion of some interesting implications for recognition and parsing is deferred to Section 3. Due to the scope of the paper, proofs of Theorems 1 and 2 below are not carried out in all their details: we only present formal specifications for the studied reductions and discuss the intuitive ideas behind them.

2.1 PRELIMINARIES

Different formalisms in which rewriting is applied independently of the context have been proposed in computational linguistics for the treatment of Natural Language, where the definition of elementary rewriting operation varies from system to system. The class of linear context-free rewriting systems (LCFRS) has been defined in [Vijay-Shanker *et al.*, 1987] with the intention of capturing through a generalization common properties that are shared by all these formalisms.

The basic idea underlying the definition of LCFRS is to impose two major restrictions on rewriting. First of all, rewriting operations are applied in the derivation of a string in a way that is independent of the context. As a second restriction, rewriting operations are generalized by means of abstract composition operations that are linear and nonerasing. In a LCFR system, both restrictions are realized by defining an underlying context-free grammar where each production is associated with a function that encodes a composition operation having the above properties. The following definition is essentially the same as the one proposed in [Vijay-Shanker *et al.*, 1987].

Definition 1 *A rewriting system* $G = (V_N, V_T, P, S)$ *is a* linear context-free rewriting system *if:*

(i) V_N *is a finite set of* nonterminal *symbols,* V_T *is a finite set of* terminal *symbols,* $S \in V_N$ *is the* start *symbol; every symbol* $A \in V_N$ *is associated with an integer* $\varphi(A) > 0$, *called the* fan-out *of* A;

(ii) P *is a finite set of* productions *of the form* $A \rightarrow f(B_1, B_2, \ldots, B_r)$, $r \geq 0$, $A, B_i \in V_N$, $1 \leq i \leq r$, *with the following restrictions:*

(a) f *is a function in* C^D, *where* $D = (V_T^*)^\psi$, ψ *is the sum of the fan-out of all* B_i*'s and* $C = (V_T^*)^{\varphi(A)}$;

(b) $f(x_{1,1}, \ldots, x_{1,\varphi(B_1)}, \ldots, x_{r,\varphi(B_r)}) = \langle y_1, \ldots, y_{\varphi(A)} \rangle$ *is defined by some grouping into* $\varphi(A)$ *sequences of all and only the elements in the sequence* $x_{1,1}, \ldots, x_{r,\varphi(B_r)}, a_1, \ldots, a_\sigma$, $\sigma \geq 0$, *where* $a_i \in V_T$, $1 \leq i \leq \sigma$.

The languages generated by LCFR systems are called LCFR languages. We assume that the starting symbol has unitary fan-out. Every LCFR system G is naturally associated with an underlying context-free grammar G_u. The usual context-free derivation relation, written $\Rightarrow_{G_u}$, will be used in the following to denote underlying derivations in G. We will also use the reflexive and transitive closure of such a relation, written $\stackrel{*}{\Rightarrow}_{G_u}$. As a convention, whenever the evaluation of all functions involved in an underlying derivation starting with A results in a $\varphi(A)$-tuple w of terminal strings, we will say that A *derives* w and write $A \stackrel{*}{\Rightarrow}_G w$. Given a nonterminal $A \in V_N$, the language $L(A)$ is the set of all $\varphi(A)$-tuples w such that $A \stackrel{*}{\Rightarrow}_G w$. The language generated by G, $L(G)$, is the set $L(S)$. Finally, we will call LCFRS(k) the class of all LCFRS's with fan-out bounded by k, $k > 0$ and r-LCFRS the class of all LCFRS's whose productions have right-hand side length bounded by r, $r \geq 0$.

2.2 HARDNESS FOR NP

The *membership problem* for the class of linear context-free rewriting systems is represented by means of a formal language LRM as follows. Let G be a grammar in LCFRS and w be a string in V_T^*, for some alphabet V_T^*; the pair $\langle G, w \rangle$ belongs to LRM if and only if $w \in L(G)$. Set LRM naturally represents the problem of the recognition of a linear context-free rewriting language when we take into account both the grammar and the string as input variables. In the following we will also study the decision problems LRM(k) and r-LRM, defined in the obvious way. The next statement is a characterization of r-LRM.

Theorem 1 3SAT $\leq_p$ 1-LRM.

Outline of the proof. Let $\langle U, C \rangle$ be an arbitrary instance of the 3SAT problem, where $U = \{u_1, \ldots, u_p\}$ is a set of variables and $C = \{c_1, \ldots c_n\}$ is a set of clauses; each clause in C is represented by a string of length three over the alphabet of all literals, $L_U = \{u_1, \overline{u}_1, \ldots, u_p, \overline{u}_p\}$. The main idea in the following reduction is to use the derivations of the grammar to guess truth assignments for U and to

use the fan-out of the nonterminal symbols to work out the dependencies among different clauses in C.

For every $1 \leq k \leq p$, let $\mathcal{A}_k = \{c_j \mid u_k \text{ is a substring of } c_j\}$ and let $\overline{\mathcal{A}}_k = \{c_j \mid \overline{u}_k \text{ is a substring of } c_j\}$; let also $w = c_1 c_2 \dots c_n$. We define a linear context-free rewriting system $G = (V_N, C, P, S)$ such that $V_N = \{T_i, F_i \mid 1 \leq i \leq p+1\} \cup \{S\}$, every nonterminal (but S) has fan-out n and P contains the following productions (f_I denotes the identity function on $(C^*)^n$):

(i) $S \to f_0(T_1)$,
$S \to f_0(F_1)$,
where $f_0(x_1, \dots, x_n) = x_1 \dots x_n$;

(ii) for every $1 \leq k \leq p$ and for every $c_j \in \mathcal{A}_k$:
$T_k \to f_k^{(j)}(T_k)$,
$T_k \to f_I(T_{k+1})$,
$T_k \to f_I(F_{k+1})$,
where $f_k^{(j)}(x_1, \dots, x_n) = \langle x_1, \dots, x_j c_j, \dots, x_n \rangle$;

(iii) for every $1 \leq k \leq p$ and for every $c_j \in \overline{\mathcal{A}}_k$:
$F_k \to \overline{f}_k^{(j)}(F_k)$,
$F_k \to f_I(T_{k+1})$,
$F_k \to f_I(F_{k+1})$,
where $\overline{f}_k^{(j)}(x_1, \dots, x_n) = \langle x_1, \dots, x_j c_j, \dots, x_n \rangle$;

(iv) $T_{p+1} \to f_{p+1}()$,
$F_{p+1} \to f_{p+1}()$,
where $f_{p+1}() = \langle \varepsilon, \dots, \varepsilon \rangle$.

From the definition of G it directly follows that $w \in L(G)$ implies the existence of a truth-assignment that satisfies C. The converse fact can be shown starting from a truth assignment that satisfies C and constructing a derivation for w using (finite) induction on the size of U. The fact that $\langle G, w \rangle$ can be constructed in polynomial deterministic time is also straightforward (note that each function $f_k^{(j)}$ or $\overline{f}_k^{(j)}$ in G can be specified by an integer j, $1 \leq j \leq n$). □

The next result is a characterization of LRM(k) for every $k \geq 2$.

Theorem 2 3SAT $\leq_p$ LRM(2).

Outline of the proof. Let $\langle U, C \rangle$ be a generic instance of the 3SAT problem, $U = \{u_1, \dots, u_p\}$ and $C = \{c_1, \dots, c_n\}$ being defined as in the proof of Theorem 1. The idea in the studied reduction is the following. We define a rather complex string $w^{(1)} w^{(2)} \dots w^{(p)} w_c$, where w_c is a representation of the set C and $w^{(i)}$ controls the truth assignment for the variable u_i, $1 \leq i \leq p$. Then we construct a grammar G such that $w^{(i)}$ can be derived by G only in two possible ways and only by using the first string components of a set of nonterminals $N^{(i)}$ of fan-out two. In this way the derivation of the substring $w^{(1)} w^{(2)} \dots w^{(p)}$ by nonterminals $N^{(1)}, \dots, N^{(p)}$ corresponds to a guess of a truth assignment for U. Most important, the right string components of nonterminals in $N^{(i)}$ derive the symbols within w_c that are compatible with the truth-assignment chosen for u_i. In the following we specify the instance $\langle G, w \rangle$ of LRM(2) that is associated to $\langle U, C \rangle$ by our reduction.

For every $1 \leq i \leq p$, let $\mathcal{A}_i = \{c_j \mid u_i \text{ is included in } c_j\}$ and $\overline{\mathcal{A}}_i = \{c_j \mid \overline{u}_i \text{ is included in } c_j\}$; let also $m_i = |\mathcal{A}_i| + |\overline{\mathcal{A}}_i|$. Let $Q = \{a_i, b_i \mid 1 \leq i \leq p\}$ be an alphabet of not already used symbols; for every $1 \leq i \leq p$, let $w^{(i)}$ denote a sequence of $m_i + 1$ alternating symbols a_i and b_i, i.e. $w^{(i)} \in (a_i b_i)^+ \cup (a_i b_i)^* a_i$. Let $G = (V_N, Q \cup C, P, S)$; we define $V_N = \{S\} \cup \{A_j^{(i)} \mid 1 \leq i \leq p,\ 1 \leq j \leq m_i\}$ and $w = w^{(1)} w^{(2)} \dots w^{(p)} c_1 c_2 \dots c_n$. In order to specify the productions in P, we need to introduce further notation. We define a function α such that, for every $1 \leq i \leq p$, the clauses $c_{\alpha(i,1)}, c_{\alpha(i,2)}, \dots c_{\alpha(i,|\mathcal{A}_i|)}$ are all the clauses in $\mathcal{A}_i$ and the clauses $c_{\alpha(i,|\mathcal{A}_i|+1)}, \dots c_{\alpha(i,m_i)}$ are all the clauses in $\overline{\mathcal{A}}_i$. For every $1 \leq i \leq p$, let $\gamma(i,1) = a_i b_i$ and let $\gamma(i,h) = a_i$ (resp. b_i) if h is even (resp. odd), $2 \leq h \leq m_i$; let also $\overline{\gamma}(i,h) = a_i$ (resp. b_i) if h is odd (resp. even), $1 \leq h \leq m_i - 1$, and let $\overline{\gamma}(i, m_i) = a_i b_i$ (resp. $b_i a_i$) if m_i is odd (resp. even). Finally, let $z = \sum_{i=1}^{p} m_i$. The following productions define set P (the example in Figure 1 shows the two possible ways of deriving by means of P the substring $w^{(i)}$ and the corresponding part of $c_1 \dots c_n$).

(i) for every $1 \leq i \leq p$:
(a) for $1 \leq h \leq |\mathcal{A}_i|$:
$A_h^{(i)} \to \langle \gamma(i,h), c_{\alpha(i,h)} \rangle$,
$A_h^{(i)} \to \langle \gamma(i,h), \varepsilon \rangle$,
$A_h^{(i)} \to \langle \overline{\gamma}(i,h), \varepsilon \rangle$;
(b) for $|\mathcal{A}_i| + 1 \leq h \leq m_i$:
$A_h^{(i)} \to \langle \gamma(i,h), \varepsilon \rangle$,
$A_h^{(i)} \to \langle \overline{\gamma}(i,h), c_{\alpha(i,h)} \rangle$,
$A_h^{(i)} \to \langle \overline{\gamma}(i,h), \varepsilon \rangle$;

(ii) $S \to f(A_1^{(1)}, \dots, A_{m_1}^{(1)}, \dots, A_{m_p}^{(p)})$,

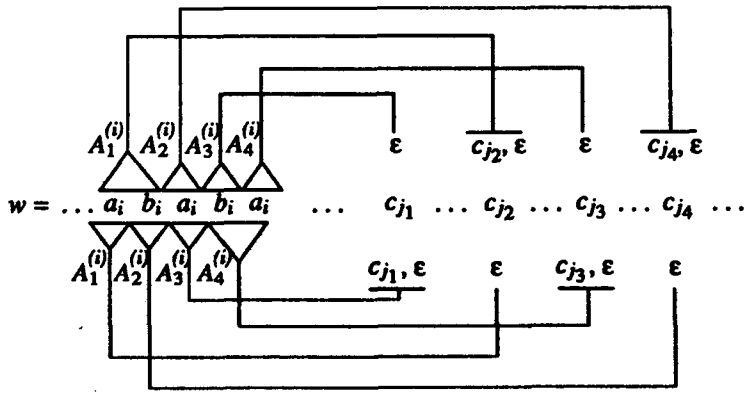

Figure 1: Let $\mathcal{A}_i = \{c_{j_2}, c_{j_4}\}$ and $\overline{\mathcal{A}}_i = \{c_{j_1}, c_{j_3}\}$. String $w^{(i)}$ can be derived in only two possible ways in G, corresponding to the choice $u_i = true/false$. This forces the grammar to guess a subset of the clauses contained in $\mathcal{A}_i/\overline{\mathcal{A}}_i$, in such a way that all of the clauses in C are derived only once if and only if there exists a truth-assignment that satisfies C.

where f is a function of $2z$ string variables defined as

$$f(x_1^{(1)}, y_1^{(1)}, \ldots, x_{m_1}^{(1)}, y_{m_1}^{(1)}, \ldots, x_{m_p}^{(p)}, y_{m_p}^{(p)}) = x_1^{(1)} x_2^{(1)} \ldots x_{m_1}^{(1)} \ldots x_{m_p}^{(p)} y_1 y_2 \ldots y_n$$

and for every $1 \leq j \leq n$, y_j is any sequence of all variables $y_h^{(i)}$ such that $\alpha(i,h) = j$.

It is easy to see that $|G|$ and $|w|$ are polynomially related to $|U|$ and $|C|$. From a derivation of $w \in L(G)$, we can exhibit a truth assignment that satisfies C simply by reading the derivation of the prefix string $w^{(1)}w^{(2)} \ldots w^{(p)}$. Conversely, starting from a truth assignment that satisfies C we can prove $w \in L(G)$ by means of (finite) induction on $|U|$: this part requires a careful inspection of all items in the definition of G. □

2.3 COMPLETENESS FOR NP

The previous results entail NP-hardness for the decision problem represented by language LRM; here we are concerned with the issue of NP-completeness. Although in the general case membership of LRM in NP remains an open question, we discuss in the following a normal form for the class LCFRS that enforces completeness for NP (i.e. the proposed normal form does not affect the hardness result discussed above). The result entails NP-completeness for problems r-LRM ($r \geq 1$) and LRM(k) ($k \geq 2$).

We start with some definitions. In a linear context-free rewriting system G, a derivation $A \stackrel{*}{\Rightarrow}_G w$ such that w is a tuple of null strings is called a *null* derivation. A *cyclic* derivation has the underlying form $A \stackrel{*}{\Rightarrow}_G \alpha A \beta$, where both α and β derive tuples of empty strings and the overall effect of the evaluation of the functions involved in the derivation is a bare permutation of the string components of tuples in $L(A)$ (no recombination of components is admitted). A cyclic derivation is *minimal* if it is not composed of other cyclic derivations. Because of null derivations in G, a derivation $A \stackrel{*}{\Rightarrow}_G w$ can have length not bounded by any polynomial in $|G|$; this peculiarity is inherited from context-free languages (see for example [Sippu and Soisalon-Soininen, 1988]). The same effect on the length of a derivation can be caused by the use of cyclic subderivations: in fact there exist permutations of k elements whose period is not bounded by any polynomial in k. Let $\mathcal{N}$ and $\mathcal{C}$ be the set of all nonterminals that can start a null or a cyclic derivation respectively; it can be shown that both these sets can be constructed in deterministic polynomial time by using standard algorithms for the computation of graph closure.

For every $A \in \mathcal{C}$, let $\mathcal{C}(A)$ be the set of all permutations associated with minimal cyclic productions starting with A. We define a normal form for the class LCFRS by imposing some bound on the length of minimal cyclic derivations: this does not alter the weak generative power of the formalism, the only consequence being the one of imposing some canonical base for (underlying) cyclic derivations. On the basis of such a restriction, representations for sets $\mathcal{C}(A)$ can be constructed in deterministic polynomial time, again by graph closure computation.

Under the above assumption, we outline here a proof of LRM∈NP. Given an instance $\langle G, w \rangle$ of the LRM problem, a nondeterministic Turing machine

M can decide whether $w \in L(G)$ in time polynomial in $|\langle G, w\rangle|$ as follows. M guesses a "compressed" representation ρ for a derivation $S \overset{\rho'}{\Rightarrow}_G w$ such that:

(i) null subderivations within ρ' are represented by just one step in ρ, and

(ii) cyclic derivations within ρ' are represented in ρ by just one step that is associated with a guessed permutation of the string components of the involved tuple.

We can show that ρ is size bounded by a polynomial in $|\langle G, w\rangle|$. Furthermore, we can verify in deterministic polynomial time whether ρ is a valid derivation of w in G. The not obvious part is verifying the permutation guessed in (ii) above. This requires a test for membership in the group generated by permutations in $\mathcal{C}(A)$: such a problem can be solved in deterministic polynomial time (see [Furst *et al.*, 1980]).

3 IMPLICATIONS

In the previous section we have presented general results regarding the membership problem for two subclasses of the class LCFRS. Here we want to discuss the interesting status of "crossing dependencies" within formal languages, on the base of the above results. Furthermore, we will also derive some observations concerning the existence of highly efficient algorithms for the recognition of fan-out and production-length bounded LCFR languages, a problem which is already known to be in the class P.

3.1 CROSSING CONFIGURATIONS

As seen in Section 2, LCFRS(2) is the class of all LCFRS of fan-out bounded by two, and the membership problem for the corresponding class of languages is NP-complete. Since LCFRS(1) = CFG and the membership problem for context-free languages is in P, we want to know what is added to the definition of LCFRS(2) that accounts for the difference (assuming that a difference exists between P and NP). We show in the following how a binary relation on (sub)strings derived by a grammar in LCFRS(2) is defined in a natural way and, by discussing the previous result, we will argue that the additional complexity that is perhaps found within LCFRS(2) is due to the lack of constraints on the way pairs of strings in the defined relation can be composed within these systems.

Let $G \in$ LCFRS(2); in the general case, any nonterminal in G having fan-out two derives a set of pair of strings; these sets define a binary relation that is called here *co-occurrence*. Given two pairs (w_1, w_1') and (w_2, w_2') of strings in the co-occurrence relation, there are basically two ways of composing their string components within a rule of G: either by nesting (wrapping) one pair within the other, e.g. $w_1 w_2 w_2' w_1'$, or by creating a *crossing configuration*, e.g. $w_1 w_2 w_1' w_2'$; note how in a crossing configuration the co-occurrence dependencies between the substrings are "crossed". A close inspection of the construction exhibited by Theorem 2 shows that grammars containing an unbounded number of crossing configurations can be computationally complex if no restriction is provided on the way these configurations are mutually composed. An intuitive idea of why such a lack of restriction can lead to the definition of complex systems is given in the following.

In [Seki *et al.*, 1989] a tabular method has been presented for the recognition of general LCFR languages as a generalization of the well known CYK algorithm for the recognition of CFG's (see for instance [Younger, 1967] and [Aho and Ullman, 1972]). In the following we will apply such a general method to the recognition of LCFRS(2), with the aim of having an intuitive understanding of why it might be difficult to parse unrestricted crossing configurations. Let w be an input string of length n. In Figure 2, the case of a production $p_1 : A \rightarrow f(B_1, B_2, \ldots, B_r)$ is depicted in which a number r of crossing configurations are composed in a way that is easy to recognize; in fact the right-hand side of p_1 can be recognized step by step. For a symbol X, assume

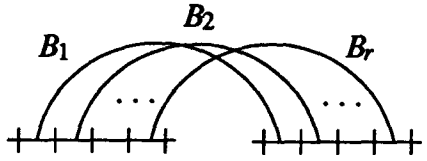

Figure 2: Adjacent crossing configurations defining a production $p_1 : A \rightarrow f(B_1, B_2, \ldots, B_r)$ where each of the right-hand side nonterminals has fan-out two.

that the sequence $X, \langle i_1, i_2\rangle, \ldots, \langle i_{q-1}, i_q\rangle$ means X derives the substrings of w that matches the positions $\langle i_1, i_2\rangle, \ldots, \langle i_{q-1}, i_q\rangle$ within w; assume also that $A[t]$ denotes the result of the t-th step in the recognition of p_1's right-hand side, $1 \leq t < r$. Then each elementary step in the recognition of p_1 can

be schematically represented as an inference rule as follows:

$$\frac{A[t], \langle i_1, i_{t+1}\rangle, \langle j_1, j_{t+1}\rangle \quad B_{t+1}, \langle i_{t+1}, i_{t+2}\rangle, \langle j_{t+1}, j_{t+2}\rangle}{A[t+1], \langle i_1, i_{t+2}\rangle, \langle j_1, j_{t+2}\rangle} \quad (1)$$

The computation in (1) involves six indices ranging over $\{1..n\}$; therefore in the recognition process such a step will be computed no more than $O(n^6)$ times.

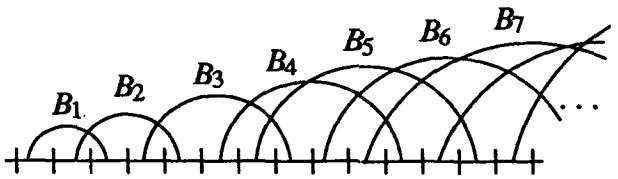

Figure 3: Sparse crossing configurations defining a production $p_2 : A \rightarrow f(B_1, B_2, \ldots, B_r)$; every non-terminal B_i has fan-out two.

On the contrary, Figure 3 presents a production p_2 defined in such a way that its recognition is considerably more complex. Note that the co-occurrence of the two strings derived by B_1 is crossed once, the co-occurrence of the two strings derived by B_2 is crossed twice, and so on; in fact crossing dependencies in p_2 are sparse in the sense that the adjacency property found in production p_1 is lost. This forces a tabular method as the one discussed above to keep track of the distribution of the co-occurrences recognized so far, by using an unbounded number of index pairs. Few among the first steps in the recognition of p_2's right-hand side are as follows:

$$\frac{A[2], \langle i_1, i_4\rangle, \langle i_5, i_6\rangle \quad B_3, \langle i_4, i_5\rangle, \langle i_8, i_9\rangle}{A[3], \langle i_1, i_6\rangle, \langle i_8, i_9\rangle}$$

$$\frac{A[3], \langle i_1, i_6\rangle, \langle i_8, i_9\rangle \quad B_4, \langle i_6, i_7\rangle, \langle i_{11}, i_{12}\rangle}{A[4], \langle i_1, i_7\rangle, \langle i_8, i_9\rangle, \langle i_{11}, i_{12}\rangle}$$

$$\frac{A[4], \langle i_1, i_7\rangle, \langle i_8, i_9\rangle, \langle i_{11}, i_{12}\rangle \quad B_5, \langle i_7, i_8\rangle, \langle i_{13}, i_{14}\rangle}{A[5], \langle i_1, i_9\rangle, \langle i_{11}, i_{12}\rangle, \langle i_{13}, i_{14}\rangle} \quad (2)$$

From Figure 3 we can see that a different order in the recognition of A by means of production p_2 will not improve the computation.

Our argument about crossing configurations shows why it might be that recognition/parsing of LCFRS(2) cannot be done efficiently. If this is true, we have a gap between LCFR systems and well known mildly context-sensitive formalisms whose membership problem is known to have polynomial solutions. We conclude that, in the general case, the addition of restrictions on crossing configurations should be seriously considered for the class LCFRS.

As a final remark, we derive from Theorem 2 a weak generative result. An open question about LCFRS(k) is the existence of a canonical bilinear form: up to our knowledge no construction is known that, given a grammar $G \in$ LCFRS(k) returns a weakly equivalent grammar $G' \in$ 2-LCFRS(k). Since we know that the membership problem for 2-LCFRS(k) is in P, Theorem 2 entails that the construction under investigation cannot take polynomial time, unless P=NP. The reader can easily work out the details.

3.2 RECOGNITION OF r-LCFRS(k)

Recall from Section 2 that the class r-LCFRS(k) is defined by the simultaneous imposition to the class LCFRS of bounds k and r on the fan-out and on the length of production's right-hand side respectively. These classes have been discussed in [Vijay-Shanker *et al.*, 1987], where the membership problem for the corresponding languages has been shown to be in P, for every fixed k and p. By introducing the notion of degree of a grammar in LCFRS, actual polynomial upper-bounds have been derived in [Seki *et al.*, 1989]: this work entails the existence of an integer function $u(r,k)$ such that the membership problem for r-LCFRS(k) can be solved in (deterministic) time $O(|G||w|^{u(r,k)})$. Since we know that the membership problems for r-LCFRS and LCFRS(k) are NP-hard, the fact that $u(r,k)$ is a (strictly increasing) non-asymptotic function is quite expected.

With the aim of finding efficient parsing algorithms, in the following we want to know to which extent the polynomial upper-bounds mentioned above can be improved. Let us consider for the moment the class 2-LCFRS(k); if we restrict ourselves to the normal form discussed in Section 2.3, we know that the recognition problem for this class is NP-complete. Assume that we have found an optimal recognizer for this class that runs in worst case time $l(G,w,k)$; therefore function l determines the best lower-bound for our problem. Two cases then arises. In a first case we have that l is not bounded by any polynomial p in $|G|$ and $|w|$: we can easily derive that P$\neq$NP. In fact if the converse is true, then there exists a Turing machine M that is able to recognize 2-LCFRS in deterministic time $|\langle G,w\rangle|^q$, for some q. For every $k > 0$, construct a Turing machine $M^{(k)}$ in the following way. Given $\langle G,w\rangle$ as input, $M^{(k)}$ tests whether $G \in$2-LCFRS(k) (which

is trivial); if the test fails, $M^{(k)}$ rejects, otherwise it simulates M on input $\langle G, w\rangle$. We see that $M^{(k)}$ is a recognizer for the class 2-LCFRS(k) that runs in deterministic time $|\langle G, w\rangle|^q$. Now select $\overline{k}$ such that, for a worst case input $w \in \Sigma^*$ and $G \in$ 2-LCFRS($\overline{k}$), we have $l(G, w, \overline{k}) > |\langle G, w\rangle|^q$: we have a contradiction, because $M^{(k)}$ will be a recognizer for 2-LCFRS($\overline{k}$) that runs in less than the lower-bound claimed for this class. In the second case, on the other hand, we have that l is bounded by some polynomial p in $|G|$ and $|w|$; a similar argument applies, exhibiting a proof that P=NP.

From the previous argument we see that finding the "best" recognizer for 2-LCFRS(k) is as difficult as solving the P vs. NP question, an extremely difficult problem. The argument applies as well to r-LCFRS(k) in general; we have then evidence that considerable improvement of the known recognition techniques for r-LCFRS(k) can be a very difficult task.

4 CONCLUSIONS

We have studied the class LCFRS along two dimensions: the fan-out and the maximum right-hand side length. The recognition (membership) problem for LCFRS has been investigated, showing NP-hardness in all three cases in which at least one of the two dimensions above is unbounded. Some consequences of the main result have been discussed, among which the interesting relation between crossing configurations and parsing efficiency: it has been suggested that the addition of restrictions on these configurations should be seriously considered for the class LCFRS. Finally, the issue of the existence of efficient algorithms for the class r-LCFRS(k) has been addressed.

References

[Aho and Ullman, 1972] A. V. Aho and J. D. Ullman. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ, 1972.

[Furst *et al.*, 1980] M. Furst, J. Hopcroft, and E. Luks. Polynomial-time algorithms for permutation groups. In *Proceedings of the 21 th IEEE Annual Symposium on the Foundations of Computer Science*, 1980.

[Joshi *et al.*, 1991] A. Joshi, K. Vijay-Shanker, and D. Weir. The convergence of mildly context-sensitive grammatical formalisms. In P. Sells, S. Shieber, and T. Wasow, editors, *Foundational Issues in Natual Language Processing*. MIT Press, Cambridge MA, 1991.

[Kasami *et al.*, 1987] T. Kasami, H. Seki, and M. Fujii. Generalized context-free grammars, multiple context-free grammars and head grammars. Technical report, Osaka University, 1987.

[Pollard, 1984] C. Pollard. *Generalized Phrase Structure Grammars, Head Grammars and Natural Language*. PhD thesis, Stanford University, 1984.

[Seki *et al.*, 1989] H. Seki, T. Matsumura, M. Fujii, and T. Kasami. On multiple context-free grammars. Draft, 1989.

[Sippu and Soisalon-Soininen, 1988] S. Sippu and E. Soisalon-Soininen. *Parsing Theory: Languages and Parsing*, volume 1. Springer-Verlag, Berlin, Germany, 1988.

[Vijay-Shanker and Weir, 1992]
K. Vijay-Shanker and D. J. Weir. Parsing constrained grammar formalisms, 1992. To appear in Computational Linguistics.

[Vijay-Shanker *et al.*, 1987] K. Vijay-Shanker, D. J. Weir, and A. K. Joshi. Characterizing structural descriptions produced by various grammatical formalisms. In 25th *Meeting of the Association for Computational Linguistics (ACL'87)*, 1987.

[Vijay-Shanker, 1987] K. Vijay-Shanker. *A Study of Tree Adjoining Grammars*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1987.

[Weir, 1988] D. J. Weir. *Characterizing Mildly Context-Sensitive Grammar Formalisms*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, 1988.

[Younger, 1967] D. H. Younger. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10:189–208, 1967.