EXTENDED UNIFORMLY LIMITED T0L LANGUAGES AND MILD CONTEXT-SENSITIVITY

Suna Bensch[A] Martin Kutrib[B]
Andreas Malcher[B]

[A]Department of Computing Science, Umeå University,
90187 Umeå, Sweden
suna@cs.umu.se

[B]Institut für Informatik, Universität Giessen,
Arndtstr. 2, 35392 Giessen, Germany
{kutrib,malcher}@informatik.uni-giessen.de

Abstract
We study the fixed membership problem for k-uniformly-limited and propagating ET0L *systems (k*ulEPT0L *systems). To this end, the algorithm given in [7] is applied. It follows that k*ulEPT0L *languages are parsable in polynomial time. Since k*ulEPT0L *languages are semi-linear [1] and k*ulEPT0L *systems generate certain non-context-free languages, which capture the non-context-free phenomena occurring in natural languages, this is the last building block to show that k*ulEPT0L *languages, for* $k \geq 2$*, belong to the family of mildly context-sensitive languages.*

1. Introduction

Context-free languages are parsable in polynomial time and there are several membership algorithms based on the concept of dynamic programming for context-free languages [6, 15]. For context-sensitive languages, on the other hand, there are no known polynomial time algorithms for the membership problem. Extended Lindenmayer systems without interaction (E0L systems) can be considered as parallel counterparts of context-free grammars and E0L languages are known to be parsable in polynomial time as well [10]. However, for ET0L languages, that is, languages generated by tabled E0L systems, the membership complexity is NP-complete [12].

Research has studied many language families that lie between "context-free" (or E0L) and "context-sensitive" (or ET0L), (see, for instance [4]). In [7], for example, the authors study the membership complexity for context-free languages generated by context-free grammars which are extended so that context-free productions are applied to a fixed number k of symbols at each derivation step. The authors use results from scheduling theory and dynamic programming to

show that the membership for these extended context-free languages is decidable in polynomial time. The authors in [14], for instance, introduced a restricted version of ET0L systems, namely k-uniformly-limited ET0L systems (abbreviated $k$ulET0L systems). In these systems the parallel rewriting mechanism of Lindenmayer systems is limited such that not all symbols in a word w have to be rewritten, but $\min\{k, |w|\}$ symbols, where k is a positive integer. Note that, if $k = 1$, we have a context-free grammar (see [14]). The crucial differences between these two grammar formalisms are, that (i) the extended context-free grammars in [7] are extended with respect to their sequential rewriting mechanisms (that is, from rewriting one symbol to rewriting k symbols) and $k$ulET0L systems are limited with respect that their parallel substitution mechanisms (that is, from substituting all symbols to substituting k symbols), and (ii) in [7] the lengths of all sentential forms (after rewriting the start symbol) are at least k.

In mathematical linguistics, researchers also have been investigating language families that lie between the context-free language family and the context-sensitive language family in the Chomsky hierarchy. A concept that captures such language families is *mild context-sensitivity*. The notion of mild context-sensitivity was first mentioned in [8], where the author proposed that a class of grammars (and their associated languages) modeling the syntax of natural languages should have the following three characteristics. First, it should describe certain non-context-free structures in order to capture the non-context-free phenomena occurring in natural languages. Second, it should have the constant growth property (the constant growth property is obeyed by every semilinear language) and, third, it should be parsable in polynomial time. Note, that the concept mildly context-sensitive captures a *family of families of languages*, not a single language family. For a formal definition of mildly context-sensitive grammar formalisms see [2]. In the literature there have been many investigations of mildly context-sensitive *sequential* grammar formalisms and their languages (see, for instance, [9, 13]). There have been less investigations of mildly context-sensitive *parallel* grammar formalisms. In [1] the author investigates some restricted versions of limited parallel Lindenmayer systems with respect to their mild context-sensitivity.

In this paper, we apply the fixed membership algorithm given in [7] to propagating $k$ulET0L languages (abbreviated $k$ulEPT0L languages). It follows that $k$ulEPT0L languages are parsable in polynomial time. Moreover, it is known that $k$ulEPT0L languages are semilinear [1]. Additionally, $k$ulEPT0L systems generate non-context-free languages, such as $\{ a^n b^n c^n \mid n \geq 1 \}$, $\{ a^n b^m c^n d^m \mid n, m \geq 1 \}$, and $\{ ww \mid w \in \{a, b\}^+ \}$, which capture the non-context-free phenomena occurring in natural languages. From all this, we conclude that $k$ulEPT0L languages, for $k \geq 2$, belong to the family of mildly context-sensitive languages.

2. Definition and Preliminaries

We assume the reader to be familiar with the basic notions of Lindenmayer systems without interaction such as in [11]. In general we have the following conventions: The set of positive integers is denoted by $\mathbb{N}$ and if we want to include 0, we write $\mathbb{N}_0$. The cardinality of a set A is denoted by $\#A$. Let $V = \{a_1, a_2, \ldots, a_n\}$ be some alphabet, where the order of the symbols

is fix. By V^+ we denote the set of nonempty words; if the empty word λ is included, then we use the notation V^*. The length of a word w in V^* is the number of letters in w and written as $|w|$. The length of the empty word λ is 0.

An ET0L system is a quadruple $G = (\Sigma, H, \psi, \Delta)$, where Σ is an alphabet, $\psi \in \Sigma^*$ is the axiom, $\Delta \subseteq \Sigma$ is the terminal alphabet, and H is a finite set of finite substitutions from Σ into Σ^*. A substitution h in H is called a *table*. For x in Σ we write $x \to y$ if $y \in h(x)$. By $\mathrm{Maxr}(G)$ we denote the length of the longest right-hand side of a production in G. In general, we write $u \underset{G}{\Longrightarrow} w$ if and only if $w \in h(u)$, for u and w in Σ^* and some h in H. If the table should be noted explicitly, we write $u \underset{h}{\Longrightarrow} w$. The reflexive transitive closure of the derivation relation $\underset{G}{\Longrightarrow}$ is denoted by $\overset{*}{\underset{G}{\Longrightarrow}}$.

The language generated by G is $L(G) = \{\, w \in \Delta^* \mid \psi \overset{*}{\underset{G}{\Longrightarrow}} w \,\}$.

An ET0L system G is called *propagating* (EPT0L system, for short) if for all substitutions h in H and all $x \in \Sigma$, we have $\lambda \notin h(x)$.

In a derivation of a $k$ulET0L system at each step of the rewriting process exactly $\min\{k, |w|\}$ symbols of the word w considered have to be rewritten. That is, if $|w| < k$ then all symbols have to be rewritten, but if $|w| \geq k$ then there are $\binom{|w|}{k}$ possibilities to rewrite the word w.

Formally, a *k-uniformly-limited* ET0L *system* G ($k$ulET0L system, for short) ([14]) is a quintuple $G = (\Sigma, H, \psi, \Delta, k)$, where $k \in \mathbb{N}$ and $(\Sigma, H, \psi, \Delta)$ is an ET0L system.

The derivation relation $\underset{G}{\Longrightarrow}$ of a $k$ulET0L system is defined as follows. Let $u, w \in \Sigma^*$ and $h \in H$.

1. If $|u| \geq k$ then $u \underset{G}{\Longrightarrow} w$ if we can write

$$u = v_1 x_1 v_2 x_2 \cdots x_k v_{k+1} \quad \text{and} \quad w = v_1 z_1 v_2 z_2 \cdots z_k v_{k+1}$$

   with $x_i \in \Sigma, v_j \in \Sigma^*, z_i \in h(x_i), i = 1, \ldots, k, j = 1, \ldots, k+1$.
2. If $|u| < k$ then $u \underset{G}{\Longrightarrow} w$ if $w \in h(u)$.

A *sentential form* of a $k$ulET0L system $G = (\Sigma, H, \psi, \Delta, k)$ is a word $w \in \Sigma^*$ with $\psi \overset{*}{\underset{G}{\Longrightarrow}} w$. A propagating $k$ulET0L ($k$ulEPT0L, for short) is defined the same way as for ET0L systems.

The authors in [14] introduced the notion of *pseudo-synchronization* for $k$ulET0L grammars. A $k$ulET0L system G is called pseudo-synchronized if for every $a \in \Delta$ and for every $w \in \Sigma^*$ the following holds: if $a \Longrightarrow w$ then $w \notin \Delta^*$; that is, there are no productions of the form $a \to w$, for $a \in \Delta$ and $w \in \Delta^*$.

Theorem 2.1 ([14], Theorem 3.1) *To every kulE(P)T0L system $G = (\Sigma, H, \psi, \Delta, k)$, there exists an equivalent pseudo-synchronized kulE(P)T0L system $G' = (\Sigma', H', S, \Delta, k)$ such that $L(G) = L(G')$.*

In the following we assume a $k$ulEPT0L system to be pseudo-synchronized without explicitly mentioning.

The reader is assumed to be familiar with the basic notions of *trees*, *forests*, *root*, *parent*, *child*, *ancestor*, *descendant*, *internal node* and *leaf*. Let v be a node. The depth of v is its distance from the root of its tree, plus 1. The height of v is the distance to its furthest descendant. Leaves have height zero and roots have depth one. Let F be a forest. The height of F is the maximal height of its roots. By $|F|$ we denote the number of nodes in F and by $\#F$ we denote the number of trees in F. The *bare forest* of F is obtained by deleting all leaves in F; the bare forest of F is denoted by $\mathrm{Bare}(F)$. The *child forest* of F is obtained by deleting all roots from F.

To each derivation of a $k$ulEPT0L system one can associate a *derivation tree* t in a similar fashion as it is done for context-free grammars. Moreover, if a node is labeled with a symbol $a \in \Sigma$, and the label of its children (from left to right) form the word w, then $a \to w$ is a production in a table of the $k$ulEPT0L system. A *derivation forest* is a forest containing a tree for each symbol of the axiom. The roots of the trees are labeled by these symbols. We illustrate some of these notions by the following example.

Example 2.2 (derivation forest) *Let $G = (\Sigma, H, \psi, \Delta, k)$ be the k*ulEPT0L *system with $\Sigma = \{A, B, a, b\}$, $\Delta = \{a, b\}$, $\psi = AB$, $k = 3$, $H = \{h_1, h_2\}$, where the set of tables is given by*

$$h_1 = \{A \to AA, B \to BB, a \to a, b \to b\},$$
$$h_2 = \{A \to a, B \to b, a \to a, b \to b\}.$$

Consider the following derivation, in which we mark the symbols which are rewritten with a dot. Table h_1 is used in the first two derivation steps and then Table h_2 is applied to terminate the derivation:

$$\dot{A}\dot{B} \Longrightarrow \dot{A}\dot{A}\dot{B}B \Longrightarrow AAA\dot{A}\dot{B}B\dot{B} \Longrightarrow \dot{A}\dot{A}\dot{A}abBb \Longrightarrow aa\dot{a}a\dot{b}\dot{B}b \Longrightarrow aaaabbb.$$

The derivation forest for aaaabbb is illustrated in Figure 1. ■

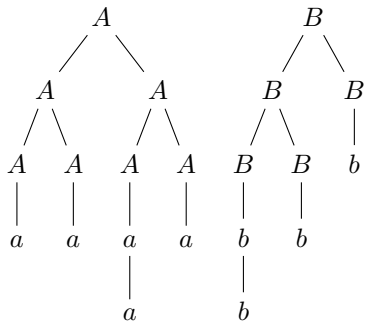

Figure 1: Derivation forest of *aaaabbb* derived by the $k$ulEPT0L system given in Example 2.2.

In the following we will divide the derivations of a $k$ulEPT0L system into two phases. The sentential forms w in a derivation of a $k$ulEPT0L can be divided into a phase in which each $|w| < k$ and into the phase in which each $|w| \geq k$. In the first phase all symbols have to be rewritten and in the second phase exactly k symbols have to be rewritten. The derivation steps in the first phase correspond to derivation steps in an EPT0L derivation and the second phase is called a *k-derivation* of a $k$ulEPT0L system. Note that in a propagating $k$ulET0L the lengths of the sentential forms do not decrease. In the following we define a k-derivation of a $k$ulEPT0L system and its k-derivation forest.

Definition 2.3 (k-derivation forest) *Let $G = (\Sigma, H, \psi, \Delta, k)$ be a $k$ulEPT0L system. A k-derivation in G is a sequence of words $w_1, \ldots, w_{l+1}$, such that, for all $1 \leq i \leq l$, $w_i \underset{G}{\Longrightarrow} w_{i+1}$ and $|w_i| \geq k$, $1 \leq i \leq l+1$.*

The length of the derivation $w_1, \ldots, w_{l+1}$ is l and the ith step is $w_i \Longrightarrow w_{i+1}$.

*A k-*derivation forest *is a derivation forest that corresponds to a k-derivation.*

Example 2.4 (k-derivation forest) *Note that the derivation forest given in Example 2.2 is not a k-derivation forest, since $|\psi| < k$. Figure 2 illustrates a k-derivation forest for the k-derivation $\dot{A}\dot{A}\dot{B}B \Longrightarrow AAA\dot{A}\dot{B}B\dot{B} \Longrightarrow \dot{A}\dot{A}\dot{A}abBb \Longrightarrow aa\dot{a}ab\dot{B}b \Longrightarrow aaaabbb$ as derived by the $k$ulEPT0L system given in Example 2.2.* ■

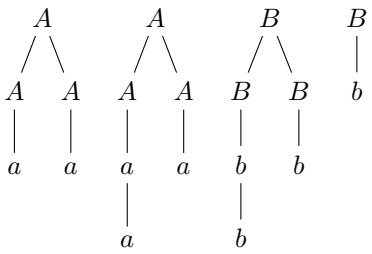

Figure 2: k-derivation forest of *aaaabbb* derived by the $k$ulEPT0L system given in Example 2.2.

Next we turn to define *schedules* on forests following [7]. Let F be a forest. We assume that there are k processors that correspond to rewriting k symbols in each derivation step. Every node in F is interpreted as a *task*. We assume that all tasks are unit-length (that is, they take the same time to be executed). The parent-child relation in F specifies the precedence constraints (that is, the parent nodes are scheduled before its children nodes). A *k-schedule* is then a sequence of *slots*, where each slot contains up to k tasks. Each slot has a corresponding time unit. Each slot indicates which of the at most k tasks are to be scheduled in the corresponding time unit. Each slot is filled with symbols that occur on the left hand side of a production in a table $h \in H$ of a given $k$ulEPT0L system $G = (\Sigma, H, \psi, \Delta, k)$. Figure 3 illustrates these notions.

Definition 2.5 (k-schedule [7]) *Let F be a forest and let $k \geq 1$. A k-schedule of F is a function σ mapping the nodes of F onto the set $\{1, \ldots, l\}$, for some $l \leq |F|$, such that*

time →	1	2	...	l
processor 1 →				
processor 2 →				
⋮				
processor k →				
	slot 1	slot 2		slot l

Figure 3: An illustration of a k-schedule of length l with k processors. Each cell will be filled with one symbol from an alphabet.

1. $1 \leq \#\sigma^{-1}(i) \leq k$ *for all* $1 \leq k \leq l$,
2. *for each pair of nodes* v_1, v_2 *in* F, *if* v_2 *is a successor of* v_1, *then* $\sigma(v_2) > \sigma(v_1)$.

The length of σ is l and $\sigma^{-1}(i)$ is called the ith slot of σ. The tasks of slot i are scheduled at time i (that is, $\#\sigma^{-1}(i)$ out of the k processors are assigned a task at that time). There are $k - \#\sigma^{-1}(i)$ *idle periods* in slot i (that is, periods in which not all k processors are used). A schedule σ has $p(\sigma)$ idle periods, where

$$p(\sigma) = \sum_{i=1}^{l} (k - \#\sigma^{-1}(i)) = l \cdot k - |F|.$$

A schedule σ is *optimal* for F if there is no schedule σ' of F with $p(\sigma') \leq p(\sigma)$. Note that optimal schedules have minimal length. The number of idle periods of F, denoted $p(F)$, is the number of idle periods in an *optimal schedule* for F.

If $p(\sigma) = 0$ in a schedule σ, then σ is called *perfect*.

1	2	3	4	5
A	A	A	A	a
B	A	B	A	b
	B	B	A	B

1	2	3	4
A	A	A	a
A	B	A	b
B	B	A	B

Figure 4: The table on the left side is a k-schedule, $k = 3$, for the derivation forest given in Figure 1; the schedule is optimal and has one idle period. The table on the right side is a k-schedule, $k = 3$, for the k-derivation forest given in Figure 2; it is a perfect schedule.

Observe that the bare forests of k-derivation forests have perfect k-schedules. If $v_1, \ldots, v_k$ are symbols that are rewritten in step i during a k-derivation, then $v_1, \ldots, v_k$ occur in the ith slot of the corresponding perfect schedule.

Lemma 2.6 ([7]) *A derivation forest is a* k*-derivation forest if and only if its bare forest has a perfect schedule.*

Lemma 2.7 *(first and second phase) Let* $G = (\Sigma, H, \psi, \Delta, k)$ *be a* $k$ulEPT0L *system and let* $\psi \underset{G}{\Longrightarrow} w_1 \underset{G}{\overset{*}{\Longrightarrow}} \cdots \underset{G}{\overset{*}{\Longrightarrow}} w_l \underset{G}{\overset{*}{\Longrightarrow}} \cdots \underset{G}{\overset{*}{\Longrightarrow}} w_n = w$ *be a derivation of* w *in* G, *where* $|\psi| < k$, $|w_i| < k$, *for* $1 \leq i \leq l-1$ *and* $|w_l| \geq k$.

A word w is in $L(G)$ if and only if there exists a derivation $\psi \overset{*}{\underset{G}{\Longrightarrow}} w$, such that

1. *there is a derivation tree t of w_l from ψ, and*
2. *there is a k-derivation tree s of w from w_l, such that $p(\mathrm{B}are(s)) = 0$.*

We refer to the derivation from ψ to w_l as the first phase and to the derivation from w_l to w as the second phase. If $|\psi| \geq k$, then the derivation has no first phase.

Proof. All slots in a k-schedule for a k-derivation are filled with tasks and there are no idle periods. □

The following algorithm is for obtaining a *Highest Level First* (HLF) k-schedule for a forest. Roughly speaking, the algorithm builds an HLF k-schedule for a forest F by scheduling the nodes on the longest paths in a tree in F.

Algorithm 2.8 ([7])

1. *If F consists of at least k trees, then $\sigma^{-1}(1)$ contains the roots of the k highest trees (for trees of equal height the choice is arbitrary).*
2. *Otherwise, $\sigma^{-1}(1)$ is the set of all the roots.*
3. *The tail of the schedule is constructed similarly, with the nodes in $\sigma^{-1}(1)$ deleted from F.*

The k-schedules in Figure 4 are not HLF k-schedules. An HLF k-schedule is given in Figure 5.

1	2	3	4
A	A	A	A
A	B	A	b
B	B	a	B

Figure 5: An HLF k-schedule for the k-derivation forest given in Figure 2.

Theorem 2.9 ([3, 5]) *Any HLF schedule for a forest is optimal.*

3. Polynomial Membership Algorithm

We divide the membership testing into the two phases of a derivation of a $k$ulEPT0L system (see Lemma 2.7). The first phase ends when the length of the sentential form is at least k. Since there are at most $(|\Sigma| + 1)^{k-1}$ different sentential forms whose length is less than k and the systems are propagating, testing membership in the first phase can be done in constant time.

For the second phase we use the algorithm in [7] for extended context-free grammars, where k nonterminal symbols are replaced in each derivation step. The algorithm in [7] is bottom-up and keeps track of all the derivation trees deriving subwords of the input word, similar to the CYK algorithm. These derivation trees are parameterized and we obtain a polynomial

size characterization. There may be a family of derivation forests for a given word. The parameterized trees are called *frames* and the parameters are the root symbol, the start and end position of the subword in w, the height of the subtree and the number of its nodes.

The algorithm computes then the number of idle periods for a frame collection by using the *median*. In [5] the median was used to present a polynomial time scheduling algorithm for forests and other graphs assuming a constant number of processors. Note that the number of idle periods for various frame collections have to be computed. Intuitively, all those k-derivation trees which are higher than the median are "hard" to schedule, whereas all other k-derivation trees are "easy" to schedule. There are only a polynomial number of frame collections that are "hard" to schedule and this achieves a polynomial time algorithm for solving membership for a constant k and a constant $k$ulEPT0L system.

Definition 3.1 (k-median [7]) *The k-median of a forest F is one plus the height of the kth highest tree of F. If F contains less than k trees, then the median is zero. The* k-high forest *of F is the set of all those trees in F which are strictly higher than the median. The* k-low forest *is the set of the remaining trees.*

The k-high forest and k-low forest of a forest F are denoted by $\mathrm{High}_k(F)$ and $\mathrm{Low}_k(F)$, respectively.

Theorem 3.2 ([7]) *Let F be a forest and σ be a k-schedule for $\mathrm{High}_k(F)$ with q idle periods. Then there is a schedule σ' for the whole forest F, such that:*

1. *if $q \geq |\mathrm{Low}_k(F)|$, then the length of σ' is as most as long as the length of σ;*
2. *if $q < |\mathrm{Low}_k(F)|$, then σ' has idle periods only in its last slot.*

Lemma 3.3 ([7]) *Assume that the HLF schedules for $\mathrm{High}_k(F)$ have q idle periods. Then the HLF schedules for F have*

1. *$q - |\mathrm{Low}_k(F)|$ idle periods if $q \geq |\mathrm{Low}_k(F)|$,*
2. *$-|F| \mod k$ idle periods, otherwise.*

The following lemma is a restatement for $k$ulEPT0L derivations and restricts the length of a derivation for a word w in a $k$ulEPT0L language polynomially in n.

Lemma 3.4 *Let $G = (\Sigma, H, \psi, \Delta, k)$ be a $k$ulEPT0L system and let $w \in \Delta^*$ with $|w| = n$. Then $w \in L(G)$ if and only if there exists a derivation tree t of w from ψ, such that the height of t is at most $f(n) = nk^{2k}(\#\Sigma)^{k(k+1)/2}$ and $|t| \leq nf(n)$.*

The bound on the total number of nodes $|t|$ follows from the fact that in each tree level, there can be at most n nodes.

The following definition is a restatement for $k$ulEPT0L systems and its k-derivation trees. The k-derivation trees are parameterized into *frames*.

Definition 3.5 *Let G be a* $k$ulEPT0L *and let $w = a_1 \cdots a_n$, where $a_1, \ldots, a_n \in \Delta$ and $|w| \geq k$. A frame R (of w) is a quintuple (A, l, r, h, c), such that $A \in \Sigma$ is the root of R, $1 \leq l \leq r \leq n$, and there is a k-derivation tree t of $a_l \cdots a_r$ from A in G, such that its bare tree has height h and c nodes. If the derivation tree is of height zero, that is, A is a terminal symbol, then $c = 0$ and $h = -1$.*

A tree t as above is called a *frame tree* for R. The height of a frame R is h and the size of R, denoted $|R|$, is c.

An ordered set $\mathcal{R}$ of frames (A, l, r, h, c) is called a *frame collection*, where all A in $\mathcal{R}$ occur on the left hand side of a production in a table $h \in H$ of a given $k$ulEPT0L system. The height of $\mathcal{R}$ is the maximum of the frame heights in $\mathcal{R}$ and the size of $\mathcal{R}$ is the sum of the sizes of the frames in $\mathcal{R}$. If F is a forest, such that the ith tree in F is a frame tree for the ith frame in $\mathcal{R}$, for $1 \leq i \leq \#F = \#\mathcal{R}$, then F is called a *frame forest of* $\mathcal{R}$ (see Example 3.6 for an example).

Example 3.6 *The forest in Figure 2 is a frame forest for the frame collection $\mathcal{R}$:*

$$\mathcal{R} = \{(A, 1, 4, 3, 8), (B, 5, 7, 3, 6)\}.$$

■

The notions of k-median, k-high collection (k-high forest), and k-low collection (k-low forest) are defined similarly for frame collections. In particular, the number of idle periods for a frame collection is given by

$$p(\mathcal{R}) = \min\{p(\text{Bare}(F)) \mid F \text{ is a frame forest of } \mathcal{R}\}.$$

The following is a restatement of Lemma 2.7 for frames.

Lemma 3.7 *Let $G = (\Sigma, H, \psi, \Delta, k)$ be a* $k$ulEPT0L *system and let $\psi \underset{G}{\Longrightarrow} w_1 \overset{*}{\underset{G}{\Longrightarrow}} \cdots \overset{*}{\underset{G}{\Longrightarrow}} w_l \overset{*}{\underset{G}{\Longrightarrow}} \cdots \overset{*}{\underset{G}{\Longrightarrow}} w_n = w$ be a derivation of w in G, where $|\psi| < k$, $|w_i| < k$, for $1 \leq i \leq l-1$ and $|w_l| \geq k$. Furthermore, let $S \to w_l$ be an additional rule for G, where S is a new symbol in $\Sigma \setminus \Delta$ and $S \to w_l$ a new production in a new table h_{new} in H.*

A word w is in $L(G)$ if and only if there exists a derivation $\psi \overset{*}{\underset{G}{\Longrightarrow}} w$, such that

1. *there is a derivation tree t of w_l from ψ, and*
2. *there exists a frame $R = (S, 1, n, h, c)$, for some h and c, such that $p(R) = 0$.*

If $|\psi| \geq k$, then let $S \to \psi$ be an additional rule in a new table in H of G.

Definition 3.8 (child collection [7]) *Let $R = (A, l, r, h, c)$ be a frame of a word w, and let $\mathcal{R} = \{R_1, \ldots, R_j\}$ be a frame collection of w, where $R_i = (A_i, l_i, r_i, h_i, c_i)$ for all $1 < i \leq j$. We*

say that $\mathcal{R}$ is a child collection of R if: $A \to A_1 \cdots A_j \in H$, $l = l_1$, $r_j = r$, and $l_i = r_{i-1} + 1$ for all $2 \leq i \leq j$, $h = 1 + \max\{h_1, \ldots, h_j\}$, and $c = 1 + \sum_{i=1}^{j} c_i$.

A child collection *of a frame collection $\mathcal{R}$ is obtained by choosing a child collection for each of the frames in $\mathcal{R}$ and taking their union.*

By Lemma 3.4, we only consider those frames $R = (A, l, r, h, c)$ with bounded length. Since there are at most $\#\Sigma$ choices for A, and n choices for each l and r and since $f(n)$ in Lemma 3.4 is a linear function, the following bound is obtained.

Corollary 3.9 ([7]) *There are $\mathcal{O}(n^5)$ frames to be computed while testing membership for a word of length n.*

The following corollary is a restatement of Lemma 3.3 for frames.

Corollary 3.10 ([7]) *Let $\mathcal{R}$ be a frame collection. Then*

$$p(\mathcal{R}) = \begin{cases} p(\mathrm{High}_k(\mathcal{R})) - |\mathrm{Low}_k(\mathcal{R})| \text{ if } p(\mathrm{High}_k(\mathcal{R})) \geq |\mathrm{Low}_k(\mathcal{R})|, \\ -|\mathcal{R}| \mod k, \text{ otherwise.} \end{cases}$$

By definition, a k-high collection consists of at most $k-1$ frames and by Corollary 3.9 at most $\mathcal{O}(n^{5(k-1)})$ idle periods of k-high collections have to be computed.

Lemma 3.11 ([7]) *Let j be the number of frames in a frame collection $\mathcal{I}$, where $\mathcal{I} = \mathrm{High}_k(\mathcal{I})$. Then*

$$p(\mathcal{I}) = \begin{cases} 0 \text{ if } \mathcal{I} \textit{ is empty}, \\ k - j + \min\{\, p(\mathcal{R}') \mid \mathcal{R}' \textit{ is a child collection of } \mathcal{I} \,\}, \text{ otherwise.} \end{cases}$$

The algorithm in [7] first constructs all the frames for the input word w, and then, computes the number of idle periods for each possible high collection. The number of idle periods for each frame of a word w is computable in polynomial time. The number of idle periods of various frame collections is computed, by increasing height, using the recurrences stated in Corollary 3.10 and Lemma 3.11. Finally, it is tested whether there exists a frame that covers all of w, that is, a frame of the form $(S, 1, |w|, h, c)$ and that has $k-1$ idle periods.

The algorithm does not only decide membership but also provides the information necessary to construct a k-derivation for an input word w (see [7]).

Theorem 3.12 ([7]) *Algorithm 1 runs in time polynomial in n, $\mathcal{O}(n^{5(k-1)(\mathrm{Maxr}(G)+1)+1})$, if both k and G are constant.*

Algorithm 1: Adjusted membership algorithm from [7] for the second phase in $k$ulEPT0L derivations

Data: A $k$ulEPT0L $G = (\{A_1, \ldots, A_m\}, H, S, \Delta, k)$, where H contains a new table h_{new} with $S \to w'$ as its single production, and a word $w = A_{p_1} \cdots A_{p_n} \in \Delta^*$.

Result: `accept` if $w \in L(G)$, otherwise `reject`.

begin

`/* test if` $w' = w$ `and` S `directly derives` w' `in one derivation step; */`

if $S \to w$ **then** `accept;`

`/* construct all the frames of` w`, of height` h`; */`

for $i := 1$ **to** n **do** $(A_{p_i}, i, i, -1, 0)$ `is a frame;`

for $h := 0$ **to** $f(n)$ **do**

  forall the $g \in H$ **do**

    forall the $A \to B_1 B_2 \cdots B_j \in g$ **do**

      forall the $1 \leq l_0 \leq \ldots \leq l_j \leq n$ **do**

        forall the $h_1, \ldots, h_j$ *with* $\max\{h_1, \ldots, h_j\} = h - 1$ **do**

          forall the $0 \leq c_1, \ldots, c_j \leq nf(n)$ **do**

            if $(B_i, l_{i-1}, l_i, h_i, c_i)$ *is a frame or* $1 \leq i \leq j$ **then**

              $(A, l_0, l_j, h, c_1 + c_2 + \ldots + c_j + 1)$ `is a frame;`

`/* compute the number of idle periods for all collections up to` $k - 1$ `frames; */`

$p(\{\}) := 0$;

for $h := -1$ **to** $f(n)$ **do**

  forall the *frame collections* $\mathcal{R}$ *of height* h*, consisting of up to* $k - 1$ *frames, each of positive height* **do**

    $q := \infty$;

    forall the *child collections* $\mathcal{R}'$ *of* $\mathcal{R}$ **do**

      forall the $\mathrm{High}_k(\mathcal{R}')$ *where all* A *in the frames occur on the left hand side of a production in a table* $g \in H$ **do**

        $\mathcal{I} := \mathrm{High}_k(\mathcal{R}')$;

        `/* Since the height of` $\mathcal{I}'$ `is` $h - 1$`, we can recur on` $p(\mathcal{I}')$`; */`

        if $p(\mathcal{I}') \geq |\mathrm{Low}_k(\mathcal{R}')|$ **then**

          $p(\mathcal{R}') := p(\mathcal{I}') - |\mathrm{Low}_k(\mathcal{R}')|$

        else $p(\mathcal{R}') := -|\mathcal{R}'| \mod k$;

        $q := \min\{q, p(\mathcal{R}')\}$;

    $p(\mathcal{R}) := k - \#\mathcal{R} + q$;

`/* This is the membership test; */`

for $h := 1$ **to** $f(n)$ **do**

  for $c := 2$ **to** $n \cdot f(n)$ **do**

    if $(S, 1, n, h, c)$ *is a frame and* $p((S, 1, n, h, c)) = k - 1$ **then**

      `accept`

    else `reject;`

References

[1] S. BENSCH, *Parallel systems as mildly context-sensitive grammar formalisms*. Ph.D. thesis, Universität Potsdam, Potsdam, Germany, 2009.

[2] H. BORDIHN, Mildly context-sensitive grammars. In: C. MARTÍN-VIDE, V. MITRANA, G. PĂUN (eds.), *Formal Languages and Application, Studies in Fuzziness and Soft Computing 148*. Springer, Berlin, 2004, 163–173.

[3] J. BRUNO, Deterministic and stochastic scheduling problems with tree-like precedence constraints. *NATO Conference* (1981).

[4] J. DASSOW, G. PĂUN, *Regulated Rewriting in Formal Language Theory*. Springer, 1989.

[5] D. DOLEV, M. WARMUTH, Scheduling flat graphs. *SIAM Journal on Computing* 14 (1985) 3, 638–657.

[6] J. EARLEY, An efficient context-free parsing algorithm. *Communications of the ACM* 13 (1970) 2, 94–102.

[7] J. GONCZAROWSKI, M. K. WARMUTH, Applications of scheduling theory to formal language theory. *Theoretical Computer Science* 37 (1985), 217–243.

[8] A. K. JOSHI, Tree adjoining grammars: how much context-sensitivity is required to provide reasonable structural descriptions? In: D. R. DOWTY, L. KARTTUNEN, A. M. ZWICKY (eds.), *Natural Language Parsing. Psychological, Computational, and Theoretical Perspectives*. Cambridge University Press, New York, 1985, 206–250.

[9] A. K. JOSHI, K. VIJAY-SHANKER, D. J. WEIR, The convergence of mildly context-sensitive grammar formalisms. In: T. WASOW, P. SELLS (eds.), *The Processing of Linguistic Structure*. MIT Press, 1991, 31–81.

[10] J. OPATRNY, K. CULIK II, Time complexity of recognition and parsing of E0L languages. In: A. LINDENMAYER, G. ROZENBERG (eds.), *Automata, Languages, Development*. North-Holland, Amsterdam, 1976, 243–250.

[11] G. ROZENBERG, A. SALOMAA, *The Mathematical Theory of L-Systems*. Academic Press, New York, 1980.

[12] J. VAN LEEUWEN, The membership question for ET0L languages is polynomially complete. *Information Processing Letters* 3 (1967), 138–143.

[13] K. VIJAY-SHANKER, D. J. WEIR, The equivalence of four extensions of context-free grammars. *Mathematical Systems Theory* 27 (1994) 6, 511–545.

[14] D. WÄTJEN, E. UNRUH, On extended k-uniformly limited T0L systems and languages. *Information Processing and Cybernetics EIK 26* 5/6 (1990), 283–299.

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