ELSEVIER

Theoretical Computer Science 215 (1999) 169–189

Theoretical Computer Science

Decidability problems in grammar systems

Valeria Mihalache [1,2]

Turku Centre for Computer Science, Lemminkaisenkatu 14 A, 4th floor, SF-20520 Turku, Finland

Received November 1996; revised May 1997
Communicated by A. Salomaa

Abstract

Most of the basic decision problems concerning derivations in cooperating distributed grammar systems have so far been open, possibly because of the lack of unifying methods and techniques. In this paper such a unifying device is proposed. It is called a *coverability tree* because it bears some resemblance to the coverability graph of place/transition Petri nets and vector addition systems. The coverability tree is always finite, which leads to rather strong decidability properties concerning both arbitrary and terminal derivations. Our method is largely independent of the mode of the derivations and answers most of the direct decidability questions about the components of the system. © 1999—Elsevier Science B.V. All rights reserved

Keywords: Formal languages; Grammar systems; Coverability tree; Decidability

1. Introduction

Grammar systems were introduced as a formal model for the phenomenon of solving a given problem by dividing it into subproblems to be solved by several "processors", in turn or in parallel. More precisely, *cooperating distributed* (CD, for short) *grammar systems* have been introduced in [2], as a grammatical representation of the *blackboard model* in problem solving (a former similar structure has been considered in [7], while a particular variant of them was approached in [1]). They are a sequential generative device, consisting of several grammars working together, according to a specified protocol (a derivation mode, which can be terminal, $*$, $=k$, $\leqslant k$, or $\geqslant k$, for a $k \geqslant 1$), towards a common goal (a language to be generated). There exists also a parallel counterpart of them, namely parallel communicating (PC, for short) grammar systems, introduced in [8] (these are a formal representation of the classroom model in artificial intelligence).

[1] On leave of absence from the University of Bucharest, Faculty of Mathematics, Romania.
[2] Research supported by the Academy of Finland, Project 137358.

PII: S0304-3975(97)00165-5

Most of the results known in grammar systems area until 1992 can be found in [3], while the most recent results are surveyed in [5]. Yet there are many challenging open questions. Among them, decidability problems constitute the subject matter of this paper.

Due to the effective equality of the family of languages generated by CD grammar systems working in the *terminal* (t) mode with the family of languages generated by $ET0L$ systems, all the classical decidability results (membership, equivalence, inclusion, emptiness, finiteness) for CD grammar systems are the same as for $ET0L$ systems. For the $*$ and $\leqslant k$ derivation modes, the equality in power of CD grammar systems with context-free grammars immediately implies the above decidability results. As for the $=k$ and $\geqslant k$ modes (inducing a strictly increasing generative power of CD grammar systems in comparison to context-free grammars, yet inferior to matrix grammars), the relation with matrix grammars yields a positive answer for those of the above problems that have a positive answer for matrix grammars (membership, emptiness, finiteness).

Yet there remain other decidability aspects, *specific* to grammar systems. Such problems are intrinsic to the notion of *component* of a system. For example, whether a component is ever used in the derivations of a CD grammar system, or whether a component is activated a bounded or an unbounded number of times, or whether it can be activated several times in a row. We approach such problems in the present paper, and prove that they are all decidable. Also, we prove that it is decidable whether a production of the system can be ever used in a derivation process. Our concern is not only on the *terminal* derivations in a system, but also on *all* the operational possibilities of the system, i.e., *all* the derivations. Some related special cases dealing with the t mode were considered earlier in [1]. Our approach is general and entirely different.

In order to have a structural representation for the derivations in a CD grammar system, we first introduce a *coverability* structure for them, and then conclude decidability properties by analyzing the latter. The coverability structure we consider is similar to a very much used construction in the theory of Petri nets, namely to the *coverability graph* for place/transition nets (see [10]), which was initially introduced for vector addition systems, in [6]. For PC grammar systems a similar construction was also considered, namely in [12]. It is worth mentioning that also for Petri nets, vector addition systems and PC grammar systems, the coverability graph/tree was used to deduce the decidability of some problems; in case of PC grammar systems, these problems are related mainly to the query symbols in the definition of the systems.

2. Preliminary notions

Throughout this paper, we use the notation and basic results of formal language theory from [4, 11]; for grammar systems notions we refer to [3]. A few notations are specified in the following.

For an alphabet V, V^* denotes the free monoid generated by V; the empty string is denoted by λ, $|x|$ is the length of $x \in V^*$, and $|x|_U$ is the number of occurrences in $x \in V^*$ of symbols of $U \subseteq V$. When $U = \{a\}$, we denote $|x|_a$ instead of $|x|_{\{a\}}$. The left-hand side of a production r (like of a Chomsky grammar) is denoted by $lhs(r)$ and its right-hand side is denoted by $rhs(r)$. We do not make any distinction between a production rule and the (unique) label associated to it.

Definition 1. A CD grammar system (of degree n, $n \geqslant 1$) is a construct

$$\Gamma = (N, T, G_1, G_2, \ldots, G_n, S),$$

where N, T are disjoint alphabets, $S \in N$, and $G_i = (N, T, P_i), 1 \leqslant i \leqslant n$, are usual context-free grammars without start symbols (N is the non-terminal set, T is the terminal set, P_i is the set of context-free rules over $N \cup T$).

For $x, y \in (N \cup T)^*$ and $1 \leqslant i \leqslant n$ we denote by

$$x \underset{G_i}{\Rightarrow} y,\ x \overset{=k}{\underset{G_i}{\Rightarrow}} y,\ x \overset{\leqslant k}{\underset{G_i}{\Rightarrow}} y,\ x \overset{\geqslant k}{\underset{G_i}{\Rightarrow}} y,\ x \overset{*}{\underset{G_i}{\Rightarrow}} y,\ x \overset{t}{\underset{G_i}{\Rightarrow}} y,$$

a derivation with respect to P_i using one rule, exactly k rules, at most k rules, at least k rules, any (strictly positive) number of rules, and as many rules as possible, respectively (if $x \overset{t}{\underset{G_i}{\Rightarrow}} y$, then there is no $z \in (N \cup T)^*$ such that $y \underset{G_i}{\Rightarrow} z$).

For $f \in \{*, t, =k, \leqslant k, \geqslant k \mid k \geqslant 1\}$, a terminal derivation in the f-mode is of the form

$$\delta : S \overset{f}{\underset{G_{i_1}}{\Rightarrow}} w_1 \overset{f}{\underset{G_{i_2}}{\Rightarrow}} w_2 \overset{f}{\underset{G_{i_3}}{\Rightarrow}} \cdots \overset{f}{\underset{G_{i_s}}{\Rightarrow}} w_s = w \in T^*$$

for $s \geqslant 1$, $1 \leqslant i_j \leqslant n$, $1 \leqslant j \leqslant s$. The language generated by Γ in the f-mode, denoted by $L_f(\Gamma)$, consists of all strings $w \in T^*$ for which a derivation δ as above exists.

Let $\Gamma = (N, T, G_1, \ldots, G_n, S)$ be a CD grammar system of degree n. Let the derivation mode be an arbitrary (fixed) $f \in \{*, t, =k, \leqslant k, \geqslant k \mid k \geqslant 1\}$. We consider the set of non-terminal symbols, N, to be ordered, i.e., $N = \{A_1, \ldots, A_m\}$, $m \geqslant 1$. Without loss of generality, assume $A_1 = S$.

A production $r \in P_i$, $1 \leqslant i \leqslant n$, is said to be *enabled at a sentential form* x of Γ if r can be applied to x, during a derivation by the component G_i. If r is enabled at x, then r may occur, *yielding* a new sentential form, say y. This is denoted by $x \overset{r}{\Rightarrow} y$. A production r is said to be *enabled in* Γ if there exists a derivation $S \overset{\sigma}{\underset{\Gamma}{\Rightarrow}} x \overset{r}{\Rightarrow} y$, where σ is an arbitrary applicable (with respect to the derivation mode f) sequence of productions.

Note that the notion of enabledness actually depends on the derivation mode of the system. We do not include this in the definition to avoid complicating notation. It will always be clear from the context to which derivation mode we refer.

The set $\mathbb{N}$ of natural numbers is extended by a special symbol ω to the set $\mathbb{N}_\omega = \mathbb{N} \cup \{\omega\}$. The operations "+", "−", "·" and the relation "$\leqslant$" over $\mathbb{N}$ are partially extended to $\mathbb{N}_\omega$ by

$$\omega + \omega = \omega + n = n + \omega = \omega,$$

$$\omega - n = \omega,$$

$$\omega \cdot n = n \cdot \omega = \omega,$$

$$n \leqslant \omega$$

for any $n \in \mathbb{N}$.

For a $k \geqslant 1$, the operations and relations over $\mathbb{N}$ ($\mathbb{N}_\omega$) are extended to the set of vectors $\mathbb{N}^k$ ($\mathbb{N}^k_w$), by applying them componentwise.

Any string $x \in (N \cup T)^*$ has associated in $\mathbb{N}^{m+2}$, vectors of the form

$$M_x = (|x|_{A_1}, \ldots, |x|_{A_m}, j, k').$$

To any sentential form x in a grammar system, such a vector $M_x \in \mathbb{N}^{m+2}$ is associated, where the $(m+1)$th coordinate, j, $0 \leqslant j \leqslant n$, stands for the component grammar which has last been active in generating x (0 is for marking the start symbol), while the $(m+2)$th coordinate, k', $0 \leqslant k' \leqslant k$, is to count the derivation steps performed in a component grammar (its value will be 0 only in case of $x = S$).

Any production rule r in a P_i, $1 \leqslant i \leqslant n$, has an associated vector in $\mathbb{N}^{m+2}$,

$$\Delta r = (|rhs(r)|_{A_1} - |lhs(r)|_{A_1}, \ldots, |rhs(r)|_{A_m} - |lhs(r)|_{A_m}, 0, 1).$$

For a sequence of productions $\alpha = r_1 r_2 \ldots r_s$, denote $\Delta\alpha = \sum_{i=1}^{s} \Delta r_i$.

For a vector $U \in \mathbb{N}^{m+2}_\omega \cup \mathbb{N}^{m+2}$, denote by $U(j)$ the jth component of it, $1 \leqslant j \leqslant m+2$.

Next define a computation over the vectors in $\mathbb{N}^{m+2}_\omega$, in a similar way to the definition of a computation in a vector addition system or in a Petri net (see [6, 10]).

Let $U \in \mathbb{N}^{m+2}_\omega$ and let $r \in P_i$, for an i, $1 \leqslant i \leqslant n$.

Production r is *enabled at the vector* U (in Γ), denoted by $U[r>_\Gamma$, if

$$U(j) \geqslant |lhs(r)|_{A_j} \quad \text{for any } j,\ 1 \leqslant j \leqslant m,$$

and one of the following holds:
(1) $1 \leqslant U(m+2) < k$, $U(m+1) = i$, or
(2) $U(m+2) = 0$, $U(m+1) = 0$, or
(3) $U(m+2) = k$, $U(m+1) \neq 0$.
If r is enabled at U, then r may occur, *yielding* a new vector $W \in \mathbb{N}^{m+2}_\omega$, denoted $U[r>_\Gamma W$, and given by
(1′) if $1 \leqslant U(m+2) < k$, then $W = U + \Delta r$;
(2′) if $U(m+2) = 0$, then $W(j) = U(j) + \Delta r(j)$, for any j, $1 \leqslant j \leqslant m$, while $W(m+1) = i$, $W(m+2) = 1$;
(3′) if $U(m+2) = k$, then $W(j) = U(j) + \Delta r(j)$, for any j, $1 \leqslant j \leqslant m$, while $W(m+1) = i$, $W(m+2) = 1$.

3. Coverability properties of CD grammar systems

Our aim in what follows is to provide a tree structure in which any derivation in a CD grammar system is associated a path emanating from the root. Such a tree will exhibit all the advantages of having a graphical representation. Moreover, since this tree will be thought as finite, an exhaustive analysis of it will be possible. We call this *coverability tree* of CD grammar systems, due to the similarity to the coverability graph in the theory of Petri nets (vector addition systems).

Roughly speaking, a coverability tree is a tree whose edges are labelled by production rules, and whose nodes are meant to correspond to the sentential forms in the system, such that the transformation of sentential forms by production rules is represented. The nodes are labelled either by vectors associated to sentential forms, or by vectors that "cover" vectors associated to sentential forms.

We present in details the definition of a coverability tree and related properties for the $=k$ derivation mode, since this one poses most of the problems. At the end of the section, we point out how this definition should be modified for the other derivation modes, to yield the same properties.

Let A and B be two arbitrary sets. If $\mathscr{T}(V,E,l_1,l_2)$ is an (A,B)-labelled tree (i.e. $l_1 : V \to A$ is the node labelling function and $l_2 : E \to B$ is the edge labelling function), then $d_{\mathscr{T}}(v,v')$ denotes the set of all nodes on the path from v to v'.

Definition 2. Let Γ be a CD grammar system and let the derivation mode be $=k$, $k \geqslant 1$. An $(\mathbb{N}_\omega^{m+2}, \bigcup_{i=1}^n P_i)$-labelled tree, $\mathscr{T}=(V,E,l_1,l_2)$, is called a *coverability tree* of Γ, if the following hold:

(1) the root, denoted by v_0, is labelled by $M_S=(1,0,\dots,0,0,0)$;

(2) for any $v \in V$ having the properties

(a) there exists a production enabled at $l_1(v)$, and

(b) there is no node $\bar{v} \in d_{\mathscr{T}}(v_0,v)$ with $v \neq \bar{v}$ and $l_1(v)=l_1(\bar{v})$,

and for any production r which is enabled at $l_1(v)$, there is a node v' such that:

(2.1) $(v,v') \in E$, $l_2(v,v')=r$,

(2.2) Considering U such that $l_1(v)[r>_\Gamma U$, then $l_1(v')$ is given by:

– $l_1(v')(m+1)=U(m+1)$, $l_1(v')(m+2)=U(m+2)$;

– for any $j, 1 \leqslant j \leqslant m$,

$$l_1(v')(j)=\begin{cases} \omega & \text{if there exists } v'' \in d_{\mathscr{T}}(v_0,v) \text{ such that} \\ & l_1(v'') \leqslant U,\ l_1(v'')(m+1)=U(m+1), \\ & l_1(v'')(m+2)=U(m+2), \text{ and } U(j)>l_1(v'')(j), \\ U(j) & \text{otherwise.} \end{cases}$$

Since any two coverability trees of a CD grammar system Γ working in the $=k$ derivation mode are isomorphic, one can speak about *the coverability tree* of that grammar system, denoted by $\mathscr{T}(\Gamma)$.

Intuitively, an ω-component implies that at that position we can have arbitrarily large numbers, i.e. we can "pump" in the corresponding sentential form arbitrarily many associated non-terminals. Moreover, the operation of "pumping" directs the derivation process in states similar to one another (that is, after "pumping", the process is at the same derivation step and in the same component grammar as before "pumping").

In order to illustrate the above definition, let us consider an example.

Example 1. Let $\Gamma=(N,T,G_1,G_2,G_3,S)$, with

$$
\begin{aligned}
N &= \{S,A,B,C,A',B',S'\},\\
T &= \{a,b,c\},\\
P_1 &= \{1:A\to aA'b,\ 2:B\to CB',\ 3:A\to ab,\ 4:B\to C\},\\
P_2 &= \{5:S\to S',\ 6:S'\to AB,\ 7:A'\to A,\ 8:B'\to B\},\\
P_3 &= \{9:C\to B,\ 10:B\to c\}
\end{aligned}
$$

(the label associated to a production is the number in front of it). One can easily verify that $L_{=2}(\Gamma)=\{a^nb^nc^n \mid n\geqslant 1\}$.

Fig. 1 depicts a fragment of the coverability tree of Γ, with respect to the $=2$ derivation mode.

Nodes are denoted by $v_0,\ldots,v_{16}$, the node labelling function, l_1, is pointed out directly in Fig. 1. As for the edge labelling function, l_2, we have

$$
\begin{array}{llll}
l_2(v_0,v_1)=5, & l_2(v_1,v_2)=6, & l_2(v_2,v_3)=4, & l_2(v_3,v_4)=1,\\
l_2(v_4,v_5)=7, & l_2(v_3,v_6)=3, & l_2(v_6,v_7)=9, & l_2(v_7,v_8)=10,\\
l_2(v_2,v_9)=1, & l_2(v_9,v_{10})=2, & l_2(v_{10},v_{11})=7, & l_2(v_{11},v_{12})=8,\\
l_2(v_{12},v_{13})=1, & l_2(v_{13},v_{14})=2, & l_2(v_{14},v_{15})=7, & l_2(v_{15},v_{16})=8.
\end{array}
$$

From the figure, one can observe that the vector $M=(0,1,1,1,0,0,0,2,2)$ with $l_1(v_{11})[8>_\Gamma M$ (where 8 stands for the production labelled 8) satisfies the requirements for introducing ω in the position corresponding to the non-terminal C, and, therefore, $l_1(v_{12})=(0,1,1,\omega,0,0,0,2,2)$. The backward arrow from node v_{12} to v_2 indicates that the vector M mentioned above is compared to $l_1(v_2)$, in order to define $l_1(v_{12})(4)=\omega$.

Due to the existence of $v_{12}\in d_{\mathscr{T}}(v_0,v_{16})$, $v_{12}\neq v_{16}$, such that $l_1(v_{16})=l_1(v_{12})$, v_{16} is an end node in $\mathscr{T}(\Gamma)$. Also, nodes v_5 and v_8 are leaf nodes in the tree, because no production is enabled at either of their labels. Moreover, v_8 is a node corresponding to a terminal derivation, since $l_1(v_8)(j)=0$ for any $1\leqslant j\leqslant 7$ and $l_1(v_8)(9)=2$.

In order to be able to use the coverability tree, we just have to establish its finiteness.

Theorem 1. *For any CD grammar system Γ, working in the $=k$ derivation mode, $k\geqslant 1$, the coverability tree $\mathscr{T}(\Gamma)$ is finite and can be effectively constructed.*

Proof. First recall that König's Lemma states that *any rooted tree in which each node has only a finite number of successors and there is no infinite path emanating from the root is a finite tree.*

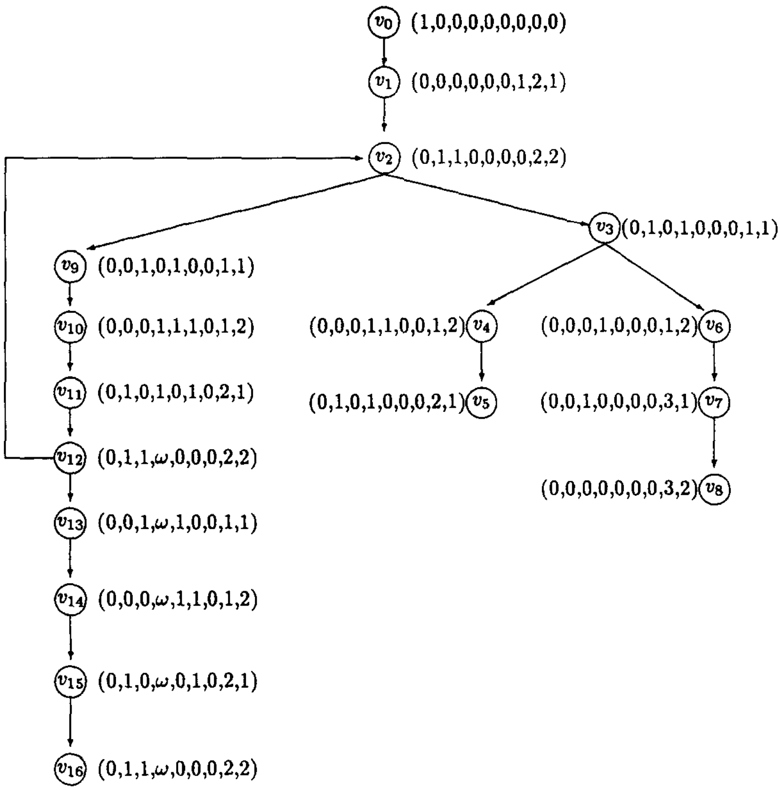

Fig. 1.

One can remark that each node in $\mathcal{T}(\Gamma)$ has only a finite number of successors. If one assumes that there exists an infinite path $v_0, v_1, \ldots$, in $\mathcal{T}(\Gamma)$, then $l_1(v_0), l_1(v_1), \ldots$, is an infinite sequence of elements of $\mathbb{N}_\omega^{m+2}$. But for such an infinite sequence there exists a non-decreasing infinite subsequence, i.e., there exist $v_{i_1}, v_{i_2}, \ldots, v_{i_t}, \ldots$, such that $l_1(v_{i_1}) \leqslant l_1(v_{i_2}) \leqslant \cdots \leqslant l_1(v_{i_t}) \leqslant \cdots$, with $i_1 \leqslant i_2 \leqslant \cdots \leqslant i_t \leqslant \cdots$.

Moreover, since $l_1(v)(m+1) \in \{0, 1, \ldots, n\}$ and $l_1(v)(m+2) \in \{0, 1, \ldots, k\}$, for any $v \in V$ (hence, they both range over a finite set of values), there exists an infinite subsequence $v_{s_1}, v_{s_2}, \ldots, v_{s_l}, \ldots$ of $v_{i_1}, v_{i_2}, \ldots, v_{i_t}, \ldots$, such that

$$l_1(v_{s_1})(m+1) = l_1(v_{s_2})(m+1) = \cdots = l_1(v_{s_l})(m+1) = \cdots,$$
$$l_1(v_{s_1})(m+2) = l_1(v_{s_2})(m+2) = \cdots = l_1(v_{s_l})(m+2) = \cdots.$$

Regarding the sequence of nodes $v_{s_1}, v_{s_2}, \ldots, v_{s_l}, \ldots$, with $l_1(v_{s_1}) \leqslant l_1(v_{s_2}) \leqslant \cdots \leqslant l_1(v_{s_l}) \leqslant \cdots$, two situations may occur:

(i) The sequence of labels of nodes above is not stationary. Then there exists an infinite strictly increasing subsequence of it, i.e., there exists $v_{r_1}, v_{r_2}, \ldots, v_{r_p}, \ldots$, subsequence of $v_{s_1}, v_{s_2}, \ldots, v_{s_l}, \ldots$, such that $l_1(v_{r_1}) < l_1(v_{r_2}) < \cdots < l_1(v_{r_p}) < \cdots$. Because the sequence in question is infinite, it follows that there is a coordinate j, $1 \leqslant j \leqslant m$, such that in the jth coordinate the vectors (in $\mathbb{N}_\omega^{m+2}$) of the sequence form a strictly increasing infinite sequence of natural numbers. But then $l_1(v_{r_2})(j)=\omega$ results, which contradicts the existence of the node v_{r_3} with $l_1(v_{r_3})(j)>l_1(v_{r_2})(j)=\omega$.

(ii) The sequence $l_1(v_{s_1})\leqslant l_1(v_{s_2})\leqslant \cdots \leqslant l_1(v_{s_i})\leqslant \cdots$ is stationary from a position, i.e. there exists $p\geqslant 1$, such that $l_1(v_{s_p})=l_1(v_{s_{p+1}})=\cdots$. By the definition of $\mathscr{T}(\Gamma)$, the node $v_{s_{p+1}}$ should be a leaf node, and therefore it could not have descendants, which contradicts the existence of an infinite path containing it.

So all paths of $\mathscr{T}(\Gamma)$ are finite, and then, by König's Lemma, $\mathscr{T}(\Gamma)$ is finite. The construction of $\mathscr{T}(\Gamma)$ is effective following the definition. □

In order to establish the relation between the derivations in a CD grammar system and its coverability tree, we need some additional notions.

Definition 3. Let Γ be a CD grammar system, let the derivation mode be $=k$, $k\geqslant 1$, and let $M\in\mathbb{N}^{m+2}$, be a vector such that $M(m+1)\leqslant n$ and $M(m+2)\leqslant k$.

- The vector M is called *coverable* in Γ if there exists a derivation $S\overset{\sigma}{\underset{\Gamma}{\Rightarrow}}x$, such that $M_x\geqslant M$ and $M_x(m+t)=M(m+t)$, for $1\leqslant t\leqslant 2$, where σ is an applicable (with respect to the $=k$ derivation mode) sequence of productions.
- The vector M is called *coverable in the coverability tree* $\mathscr{T}(\Gamma)$ if there exists a node v of $\mathscr{T}(\Gamma)$ such that $l_1(v)\geqslant M$ and $l_1(v)(m+t)=M(m+t)$, for $1\leqslant t\leqslant 2$.

The next two theorems state the equivalence between the coverability of a vector in a CD grammar system and in the associated coverability tree.

Theorem 2. *Let Γ be a CD grammar system working in the $=k$ derivation mode, $k\geqslant 1$, and let $M\in\mathbb{N}^{m+2}$. If M is coverable in Γ, then M is coverable in $\mathscr{T}(\Gamma)$.*

Proof. Let M be a vector coverable in Γ. Then $M(m+1)\leqslant n$, $M(m+2)\leqslant k$ and there exists a derivation in Γ

$$S=x_0\overset{r_1}{\Rightarrow}x_1\overset{r_2}{\Rightarrow}x_2\Rightarrow\cdots\overset{r_h}{\Rightarrow}x_h$$

such that $M_{x_h}\geqslant M$, $M_{x_h}(m+t)=M(m+t)$, for $1\leqslant t\leqslant 2$, where $x_i\overset{r_i}{\Rightarrow}x_{i+1}$ stands for the rewriting of the string x_i into x_{i+1} using production r_i. We prove that there exists in $\mathscr{T}(\Gamma)$ a sequence of nodes $v_0,v_1,\ldots,v_h$ (not necessarily pairwise distinct), such that for all i, $1\leqslant i\leqslant h$, $l_1(v_i)\geqslant M_{x_i}$, and $l_1(v_i)(m+s)=M_{x_i}(m+s)$, $1\leqslant s\leqslant 2$.

The node v_0 is the root of $\mathscr{T}(\Gamma)$, and therefore $l_1(v_0)=M_S=M_{x_0}$ holds.

Suppose one has determined the nodes $v_0,\ldots,v_i$, $0\leqslant i<h$, such that $l_1(v_j)\geqslant M_{x_j}$ and $l_1(v_j)(m+s)=M_{x_j}(m+s)$, for all j, $0\leqslant j\leqslant i$, $1\leqslant s\leqslant 2$. One has to determine the node v_{i+1}. Two situations may occur:

(1) *v_i is not a leaf node.* Then the production r_{i+1} is enabled at M_{x_i}. Since $l_1(v_i)\geqslant M_{x_i}$ and $l_1(v_i)(m+s)=M_{x_i}(m+s)$, $1\leqslant s\leqslant 2$, it follows that r_{i+1} is enabled at $l_1(v_i)$ as well. The descendant v' of v_i for which $l_2(v_i,v')=r_{i+1}$ satisfies $l_1(v')\geqslant M_{x_{i+1}}$, and

$l_1(v')(m+s) = M_{x_{i+1}}(m+2)$, for $1 \leqslant s \leqslant 2$. Therefore, the node v_{i+1} is uniquely determined as $v_{i+1} = v'$.

(2) *v_i is a leaf node.* Since there is at least one production enabled at $l_1(v_i)$ (namely r_{i+1}), the only possible case leading to this situation is that there exists a node $v \in d_{\mathcal{T}}(v_0, v_i)$, $v \neq v_i$, with $l_1(v_i) = l_1(v)$. Then r_{i+1} is enabled at $l_1(v)$, and the desired node v_{i+1} is the unique node v' such that $l_2(v, v') = r_{i+1}$. $\square$

Theorem 3. *Let Γ be a CD grammar system working in the $=k$ derivation mode, $k \geqslant 1$, and let $M \in \mathbb{N}^{m+2}$. If M is coverable in $\mathcal{T}(\Gamma)$, then M is coverable in Γ.*

Proof. The proof bears resemblance with the proof of the corresponding theorem for vector addition systems [6] or for Petri nets [10].

By definition, M being coverable in $\mathcal{T}(\Gamma)$ implies the existence of a node v of $\mathcal{T}(\Gamma)$ such that $l_1(v) \geqslant M$, and $l_1(v)(m+t) = M(m+t)$, for $1 \leqslant t \leqslant 2$. Let $v_0, v_1, \ldots, v_h = v$ be the path from the root to the node v in $\mathcal{T}(G)$, and let r_i be the production which labels the edge (v_{i-1}, v_i), that is, $l_2(v_{i-1}, v_i) = r_i$, for any i, $1 \leqslant i \leqslant h$.

If $l_1(v)$ does not contain components ω, then the derivation $S = x_0 \overset{r_1}{\Rightarrow} x_1 \overset{r_2}{\Rightarrow} \cdots \overset{r_h}{\Rightarrow} x_h$ has the property $M_{x_i} = l_1(v_i)$, for any $1 \leqslant i \leqslant h$. In particular, $M_{x_h} = l_1(v)$ and, hence, there exists $x = x_h$ such that $S \underset{\Gamma}{\overset{\sigma}{\Rightarrow}} x$, and $M_x \geqslant M$, $M_x(m+t) = M(m+t)$, for $1 \leqslant t \leqslant 2$, where $\sigma = r_1 r_2 \ldots r_h$. Therefore, M is coverable in Γ.

In the case that $l_1(v)$ contains components ω, the idea is that there are vectors M_x (x being a sentential form of Γ derivable from the start symbol) which agree with $l_1(v)$ in its finite coordinates and can be made arbitrarily large in the coordinates equal to ω, by repetitions of the sequences of productions which led to occurrences of ω.

Let v' be the ancestor of v whom v was compared to when a component ω of it was introduced, and let π be the sequence of productions which labels the path from v' to v in $\mathcal{T}(\Gamma)$. One can note that the constraints $l_1(v')(m+1) = l_1(v)(m+1)$, $l_1(v')(m+2) = l_1(v)(m+2)$, which had to be satisfied in order to introduce the component ω, assure us that after applying once the sequence of productions π to a sentential form, we can apply this π again, as many times as we wish, since the derivation step and the active component grammar are the same before applying π, as after applying it. Moreover, if in a derivation, after applying π, another sequence τ of production rules had been applied, then τ can be applied also after π^s, for any $s \geqslant 1$.

Without loss of generality, we can assume that the components ω are on the first s positions in $l_1(v)(1 \leqslant s \leqslant m)$. Also, we can assume that the component ω which is on the first position in $l_1(v)$ has been introduced along the path $v_0, \ldots, v_h$ before the second component ω of $l_1(v)$, which, in turn, has been introduced before the third one, and so on, the sth component ω of $l_1(v)$ being the last one introduced along the path $v_0, \ldots, v_h$ (some of them can be introduced at the same time).

The idea is to construct, starting from the sequence of production rules $r_1, \ldots, r_h$, a sequence of productions β, such that $S \underset{\Gamma}{\overset{\beta}{\Rightarrow}} x$, and $M_x \geqslant M$.

Let us point out the sequences of productions $\alpha_1,\dots,\alpha_{s+1}$, such that $r_1r_2\dots r_h = \alpha_1\alpha_2\dots\alpha_s\alpha_{s+1}$, and $\alpha_1\dots\alpha_i$ is the minimum prefix of $r_1\dots r_h$ which led to the introduction of the ith component ω of $l_1(v)$, for any i, $1 \leqslant i \leqslant s$ (some of α_i can be the empty string, if several components ω are introduced simultaneously). Each component ω is introduced as the result of a comparison with an ancestor of the node where it is introduced, hence one can point out sequences $u_1,u_2,\dots,u_s$, which are suffixes of the sequences $\alpha_1,\alpha_1\alpha_2,\dots,\alpha_1\alpha_2,\dots,\alpha_s$, respectively, and which led to the introduction of the components ω in $l_1(v)$. Note that any application of the sequence of productions u_i to a sentential form of Γ (providing an application is possible) increases the ith component of the corresponding vector in $\mathbb{N}_\omega^{m+2}$ with at least one (by the condition to be satisfied when u_i introduces the ith component ω); it does not decrease any of the components $i+1,\dots,m$, since $l_1(v')(j) \leqslant l_1(v)(j) \neq \omega$, for any $i+1 \leqslant j \leqslant m$, but it can decrease some of the components $1,\dots,i-1$ (in the tree the fact that it can decrease some of these components causes no trouble, because they are considered to be ω, and subtracting any number from ω still leaves ω). So, when intending to apply a sequence u_i a number of times, one has first to “pump” the components $1,\dots,i-1$, i.e. the sequences $u_1,\dots,u_{i-1}$, sufficiently to allow the applicability of the sequence u_i as many times as one wishes.

Let us consider numbers $n_1,\dots,n_s$ with the properties:

$$
\begin{aligned}
n_s &\geqslant M(s) + |\Delta\alpha_{s+1}(s)|,\\
n_{s-1} &\geqslant M(s-1) + |\Delta\alpha_{s+1}(s-1)| + n_s\cdot|\Delta u_s(s-1)|,\\
n_{s-2} &\geqslant M(s-2) + |\Delta\alpha_{s+1}(s-2)| + n_s\cdot|\Delta u_s(s-2)| + n_{s-1}\cdot|\Delta u_{s-1}(s-2)|,\\
&\ \vdots\\
n_1 &\geqslant M(1) + |\Delta\alpha_{s+1}(1)| + n_s\cdot|\Delta u_s(1)| + \cdots + n_2\cdot|\Delta u_2(1)|.
\end{aligned}
$$

One can see that the sequence $\beta = \alpha_1u_1^{n_1}\alpha_2\dots\alpha_su_s^{n_s}\alpha_{s+1}$ is a permitted sequence of production rules starting from S in Γ, and hence there exists a sentential form x, such that $S \overset{\beta}{\underset{\Gamma}{\Rightarrow}} x$. Moreover, $M_x \geqslant M$ and $M_x(m+t) = M(m+t)$, $1 \leqslant t \leqslant 2$, hence M is coverable in Γ. □

The next theorem will be essential in proving decidability results for the derivations in a system.

Theorem 4. *Let Γ be a CD grammar system working in the $=k$ derivation mode, $k \geqslant 1$, and let r be a production of Γ. Then r is enabled in Γ if and only if there exists an edge labelled by r in $\mathcal{T}(\Gamma)$.*

Proof. Suppose first that r is enabled in Γ; hence, there exists a derivation $S \overset{\sigma}{\underset{\Gamma}{\Rightarrow}} x \overset{r}{\Rightarrow} x'$, where σ is an applicable sequence of productions. Let us suppose $r \in P_i$ for an i,

$1 \leqslant i \leqslant n$, and the derivation above is $S \overset{\sigma'}{\underset{\Gamma}{\Rightarrow}} \bar{x} \underset{G_i}{\Rightarrow} x_1 \underset{G_i}{\Rightarrow} \cdots \underset{G_i}{\Rightarrow} x_s \overset{r}{\underset{G_i}{\Rightarrow}} x_{s+1} = x', x_s = x$. Two situations may occur, depending on the value of $s\,(<k)$.

(i) $s \geqslant 1$, i.e. production r is not used at the first derivation step in G_i. Let $M = (a_1, a_2, \ldots, a_m, i, s)$ be the smallest vector in $\mathbb{N}^{m+2}$ such that r is enabled at M. Then $M_x \geqslant M$ and $M_x(m+t) = M(m+t)$, $1 \leqslant t \leqslant 2$, i.e. M is coverable in Γ. By Theorem 2, M is coverable in $\mathscr{T}(\Gamma)$, and therefore there exists a node v of $\mathscr{T}(\Gamma)$ such that $l_1(v) \geqslant M$, $l_1(v)(m+t) = M(m+t)$, for $1 \leqslant t \leqslant 2$.

If v is not a leaf node, r is enabled at $l_1(v)$, and hence v has a successor v' such that $l_2(v, v') = r$. If v is a leaf node, then there exists a node $v' \in d_{\mathscr{T}(\Gamma)}(v_0, v)$, $v' \neq v$, and $l_1(v') = l_1(v)$; we then apply the preceding case to the node v', and thus we find an edge labelled by r in $\mathscr{T}(G)$.

(ii) $s = 0$, i.e. production r is the first one used when G_i becomes active. We point out the component grammar that was active just before G_i has started its work (at this stage of the derivation). If there is such a component, let it be G_j, $1 \leqslant j \leqslant n$, we choose $M = (a_1, a_2, \ldots, a_m, j, k)$ as being the smallest vector in $\mathbb{N}^{m+2}$ such that r is enabled at M. If there is no such component ($x_s = S$), we choose $M = (1, 0, \ldots, 0, 0, 0)$. Afterwards the proof goes as in case (i).

Conversely, let (v, v') be an edge in $\mathscr{T}(\Gamma)$ labelled by r. Then r is enabled at the node v. Considering the vector M as the smallest vector in $\mathbb{N}^{m+2}$ such that r is enabled at it and $M(m+t) = l_1(v)(m+t)$, for $1 \leqslant t \leqslant 2$, it follows $l_1(v) \geqslant M$. This means that M is coverable in $\mathscr{T}(\Gamma)$. By Theorem 3, M is coverable in Γ, i.e. there exists a sentential form x of Γ, $S \overset{\sigma}{\underset{\Gamma}{\Rightarrow}} x$ (σ is a sequence of productions), such that $M_x \geqslant M$, and $M_x(m+t) = M(m+t)$, for $1 \leqslant t \leqslant 2$. Then r is enabled at x, and hence it is enabled in Γ. □

The coverability tree of a CD grammar system can be constructed also for the other derivation modes ($*, t, \leqslant k, \geqslant k$, for any $k \geqslant 1$), and properties similar to the above theorems hold in all cases. Changes in the construction are minor, they concern only the $(m+2)$th coordinate in the node labelling function.

Thus, for the $\geqslant k$ mode, we consider $l_1(v)(m+2) \in \{0, 1, \ldots, k, k+1\}$, where once $l_1(v)(m+2)$ has become $k+1$, and $l_1(v)(m+1) = i$ for an i, $1 \leqslant i \leqslant n$, the $(m+2)$th coordinate will not be changed anymore as long as the derivation continues by the ith component of the system. In addition, when considering the question whether a production is enabled at a node, we allow more than k derivation steps in the same component grammar.

For the $\leqslant k$ mode, the construction is very much the same as for the $= k$ mode, but changes from one component to another during derivations are allowed also after less than k steps (the notion of enabling a production should be modified accordingly).

For the $*$ derivation mode, the number of productions used by a component in a derivation does not matter anymore. Yet, for uniformity of notation, we consider that also in this case any string $x \in (N \cup T)^*$ has an associated vector $M_x \in \mathbb{N}^{m+2}$, but with $M_x = (|x|_{A_1}, \ldots, |x|_{A_m}, i, 0)$, for an i, $1 \leqslant i \leqslant n$, where the $(m+2)$th coordinate is zero and is never changed in the construction of the tree.

Finally, we can agree that for the terminal derivation mode, to a string $x \in (N \cup T)^*$ the vector $M_x = (|x|_{A_1}, \ldots, |x|_{A_m}, i, j)$, $1 \leqslant i \leqslant n$, $j \in \{0, 1\}$, is associated, where

$$j = \begin{cases} 0 & \text{if there exists } y \text{ such that } x \underset{G_i}{\Rightarrow} y, \\ 1 & \text{otherwise.} \end{cases}$$

The enabling of a production to a node and the changing of the $(m+2)$th coordinate during the construction of the tree follow the restrictions of this derivation mode.

4. Coverability tree for the generated language

As pointed out, the coverability tree considered so far depicts *all* the derivations in a CD grammar system. One can modify it as to correspond only to the *terminal* derivations in the system. What is to be done, is to "cut" from the coverability tree all the paths that do not "lead" to a node encoding a terminal string. More precisely, for a system Γ, we have to eliminate from $\mathscr{T}(\Gamma)$ all the edges that are labelled by productions which do not occur in terminal derivations, together with the corresponding nodes. We call the resulting tree the *terminal coverability tree* and denote it by $\overline{\mathscr{T}}(\Gamma)$.

Theorems validating this construction for the $*, t, \leqslant k, \geqslant k$ $(k \geqslant 1)$ modes are presented. In the case of $= k$ mode, there remain open problems.

The definition of $\overline{\mathscr{T}}(\Gamma)$ is actually the algorithm for constructing it from $\mathscr{T}(\Gamma)$.

Definition 4. Let Γ be a CD grammar system and let $\mathscr{T}(\Gamma)$ be the coverability tree of Γ. Denote by *leaves*($\mathscr{T}$) the set of leaf nodes of $\mathscr{T}(\Gamma)$. An $(\mathbb{N}_\omega^{m+2}, \bigcup_{i=1}^n P_i)$-labelled tree, $\overline{\mathscr{T}}(\Gamma) = (\overline{V}, \overline{E}, l_1, l_2)$, is called the *terminal coverability tree* of Γ, if $\overline{\mathscr{T}}(\Gamma)$ is obtained by the following algorithm:

0. Let $V' = V$, $E' = E$, $\mathscr{T}' = (V', E', l_1, l_2)$, i.e. $\mathscr{T}' = \mathscr{T}(\Gamma)$. In the sequel, we will use the same notation for the labelling functions l_1, l_2, as well as for any restrictions of them corresponding to subsets of V, E, respectively. Let *Marked* $= \emptyset$.
1. If *Marked* $=$ *leaves*($\mathscr{T}'$), then
   – $\overline{\mathscr{T}}(\Gamma) = \mathscr{T}'$;
   – End Algorithm
   endif.
2. Let v be an unmarked leaf node of $\mathscr{T}'$, i.e. $v \in$ *leaves*($\mathscr{T}'$) $-$ *Marked*.
3. If $l_1(v) = (0, \ldots, 0, i', k)$ (v corresponds to a terminal string) for an $i' \in \{1, \ldots, n\}$, then
   – *Marked* $:=$ *Marked* $\cup \{v\}$;
   – goto Step 1.
   endif.
4. <u>If</u> there exists node $\bar{v}$ on the path $d_{\mathscr{T}'}(v_0, v)$ with $l_1(\bar{v}) \in \mathbb{N}^{m+2}$ (i.e. without components ω) such that $l_1(\bar{v})(p) = l_1(v)(p)$ for any p, $1 \leqslant p \leqslant m+2$ satisfying $l_1(v)(p) \neq \omega$ (in words, on the path from the root to node v there exists a node $\bar{v}$ without

components ω, which coincides with the node v in the components non-equal to ω of v; in case $l_1(v) \in \mathbb{N}^{m+2}$, this means $l_1(\bar{v}) = l_1(v)$), <u>then</u>
- Let v' be the closest node to the root, among such nodes $\bar{v}$;
- If node v' has a descendant labelled $(0,\ldots,0,i',k)$, for an i', $1 \leqslant i' \leqslant n$, then
- *Marked* := *Marked* $\cup \{v\}$;
- goto Step 1.

else
- Let v'' be the direct ancestor of v in $\mathscr{T}'$;
- $V' := V' - \{v\}$, $E' := E' - \{(v'', v)\}$;
- goto Step 1.

endif.

<u>else</u>
- Let v'' be the direct ancestor of v in $\mathscr{T}'$;
- $V' := V' - \{v\}$, $E' := E' - \{(v'', v)\}$;
- goto Step 1.

<u>endif</u>.

One can note that due to the uniqueness (up to isomorphism) of $\mathscr{T}(\Gamma)$ for a CD grammar system Γ, the terminal coverability tree of the system, $\overline{\mathscr{T}}(\Gamma)$ is uniquely constructed.

We want to point out that, in the definition above, the action of marking nodes is performed only to leaf nodes of the temporary tree $\mathscr{T}'$ built in the algorithm. When a node v is marked, it means that the node will remain in the terminal coverability tree $\overline{\mathscr{T}}$, together with the path from the root to it. When a leaf node v of $\mathscr{T}'$ is not suitable to be marked, it will be eliminated from the tree, together with the edge connecting it to its direct ancestor.

Theorem 5. *Let Γ be a CD grammar system deriving in one of the modes $f \in \{=k, \leqslant k, \geqslant k, *, t \mid k \geqslant 1\}$. Let $\mathscr{T} = (V, E, l_1, l_2)$ be the associated coverability tree with respect to the considered derivation mode. Let $v \in V$ be any node with $l_1(v) = (a_1, a_2, \ldots, a_m, i, \bar{j})$, $a_p \in \mathbb{N}$, $1 \leqslant p \leqslant m$, such that the product $a_1 a_2 \ldots a_m \neq 0$, $1 \leqslant i \leqslant n$, while the domain of $\bar{j}$ will be specified below. Then if v has a descendant v' with $l_1(v') = (0, \ldots, 0, i', \bar{k})$, where $1 \leqslant i' \leqslant n$, while $\bar{k}$ will be specified below, one of the following situations holds:*

(i) *If f is "$=k$" and $1 \leqslant \bar{j} \leqslant k$, $\bar{k} = k$, then for any string $\alpha \in (N \cup T)^*$, $M_\alpha = (kl_1 + a_1, kl_2 + a_2, \ldots, kl_m + a_m, i, \bar{j})$, for some $l_1, \ldots, l_m \in \mathbb{N}$, there exists a derivation*

$$\alpha \overset{k-\bar{j}}{\underset{G_i}{\Longrightarrow}} \beta_1 \overset{=k}{\underset{G_{i_1}}{\Longrightarrow}} \beta_2 \overset{=k}{\underset{G_{i_2}}{\Longrightarrow}} \cdots \overset{=k}{\underset{G_{i_s}}{\Longrightarrow}} \beta_{s+1} \in T^*$$

for some $i_1, \ldots, i_s \in \{1, \ldots, n\}$.

(ii) *If f is "$\geqslant k$" and $\bar{j} \in \{1, \ldots, k, k+1\}$, $\bar{k} \in \{k, k+1\}$, then for any string $\alpha \in (N \cup T)^*$, $M_\alpha \geqslant (a_1, a_2, \ldots, a_m, i, \bar{j})$, $M_\alpha(m+1) = i$, $M_\alpha(m+2) = \bar{j}$, satisfying the additional constraint that for any p, $1 \leqslant p \leqslant m$, for which $a_p = 0$, also $M_\alpha(p) = 0$*

holds, there exists a derivation

$$\alpha \underset{G_i}{\overset{p}{\Rightarrow}} \beta_1 \underset{G_{i_1}}{\overset{\geq k}{\Rightarrow}} \beta_2 \underset{G_{i_2}}{\overset{\geq k}{\Rightarrow}} \cdots \underset{G_{i_s}}{\overset{\geq k}{\Rightarrow}} \beta_{s+1} \in T^*$$

for some $i_1,\ldots,i_s \in \{1,\ldots,n\}$, *where*

$$p \geq \begin{cases} k-\bar{j} & \text{if } \bar{j} \leq k, \\ 0 & \text{if } \bar{j} = k+1. \end{cases}$$

(iii) *If* f *is "t" and* $\bar{j} \in \{0,1\}$, $\bar{k}=1$, *then for any string* $\alpha \in (N \cup T)^*$, $M_\alpha \geq (a_1,a_2,\ldots,a_m,i,\bar{j})$, $M_\alpha(m+1)=i$, $M_\alpha(m+2)=\bar{j}$, *satisfying the additional constraint that for any* p, $1 \leq p \leq m$, *for which* $a_p=0$, *also* $M_\alpha(p)=0$ *holds, there exists a derivation*

$$\alpha \underset{G_{i_1}}{\overset{t}{\Rightarrow}} \beta_1 \underset{G_{i_2}}{\overset{t}{\Rightarrow}} \beta_2 \underset{G_{i_3}}{\overset{t}{\Rightarrow}} \cdots \underset{G_{i_s}}{\overset{t}{\Rightarrow}} \beta_s \in T^*$$

for some $i_1,\ldots,i_s \in \{1,\ldots,n\}$, *where*

$$i_1 = \begin{cases} i & \text{if } \bar{j}=0, \\ i' \neq i & \text{if } \bar{j}=1 \ (1 \leq i' \leq n). \end{cases}$$

(iv) *If* f *is "*" and* $\bar{j}=\bar{k}=0$, *then for any string* $\alpha \in (N \cup T)^*$, *with* $M_\alpha \geq (a_1,a_2,\ldots,a_m,i,0)$, $M_\alpha(m+1)=i$, $M_\alpha(m+2)=0$, *satisfying the additional constraint that for any* p, $1 \leq p \leq m$, *for which* $a_p=0$, *also* $M_\alpha(p)=0$ *holds, there exists a derivation*

$$\alpha \underset{G_i}{\overset{*}{\Rightarrow}} \beta_1 \underset{G_{i_1}}{\overset{*}{\Rightarrow}} \beta_2 \underset{G_{i_2}}{\overset{*}{\Rightarrow}} \cdots \underset{G_{i_s}}{\overset{*}{\Rightarrow}} \beta_{s+1} \in T^*$$

for some $i_1,\ldots,i_s \in \{1,\ldots,n\}$.

(v) *If* f *is "*$\leq k$*" and* $\bar{j} \in \{1,\ldots,k\}$, $\bar{k} \in \{1,\ldots,k\}$, *then for any string* $\alpha \in (N \cup T)^*$, *with* $M_\alpha \geq (a_1,a_2,\ldots,a_m,i,\bar{j})$, $M_\alpha(m+1)=i$, $M_\alpha(m+2)=\bar{j}$, *satisfying the additional constraint that for any* p, $1 \leq p \leq m$, *for which* $a_p=0$, *also* $M_\alpha(p)=0$ *holds, there exists a derivation*

$$\alpha \underset{G_{i_1}}{\overset{\leq k}{\Rightarrow}} \beta_1 \underset{G_{i_2}}{\overset{\leq k}{\Rightarrow}} \beta_2 \underset{G_{i_3}}{\overset{\leq k}{\Rightarrow}} \cdots \underset{G_{i_s}}{\overset{\leq k}{\Rightarrow}} \beta_s \in T^*$$

for some $i_1,\ldots,i_s \in \{1,\ldots,n\}$.

Proof. We prove the assertion in case of $=k$ derivation mode ($k \geq 1$). For the other situations, the proof goes in a similar manner.

We call a production of the form $A \to \alpha$ with $|\alpha|_A = 0$ an *erasing* production for the non-terminal A; an application of such a production will be referred to as an *erasing* of A.

We have v a node in $\mathscr{T}(\Gamma)$, v' a descendant of v, $l_1(v)=(a_1,a_2,\ldots,a_m,i,\bar{j})$, with at least an $a_p \neq 0$, $(1 \leq p \leq m)$, $l_1(v')=(0,0,\ldots,0,i',k)$, where $1 \leq i,\ i' \leq n$.

Since at least one of the components $a_p \neq 0$, $1 \leqslant p \leqslant m$, while $l_1(v')(p) = 0$, it means that on the path $d_{\mathscr{T}}(v, v')$, at least one erasing production for the non-terminal A_p is used.

Let $A_{i_1}, \ldots, A_{i_s}$, $s \geqslant 1$ be all the non-terminals for which erasing productions are applied on the path $d_{\mathscr{T}}(v, v')$. For the simplicity of the notation, we assume $i_1 = 1, \ldots, i_s = s$. We still can assume, without loss of generality, that $l_1(v')(1)$ is the first component of $l_1(v')$ which becomes 0 on the path $d_{\mathscr{T}}(v, v')$, then $l_1(v')(2)$ becomes 0, and so on, $l_1(v')(s)$ being the last one that becomes 0.

Let us point out, on the path $d_{\mathscr{T}}(v, v')$, for each of the non-terminals $A_1, \ldots, A_s$, when an erasing production was used for the last time, respectively. Thus, denoting by π the sequence of productions labelling $d_{\mathscr{T}}(v, v')$, we can write

$$\pi = \pi_1 r_1 \pi_2 r_2 \ldots \pi_s r_s \pi_{s+1},$$

where for any p, $1 \leqslant p \leqslant s$, r_p is the last erasing production rule used for A_p on the path $d_{\mathscr{T}}(v, v')$.

Note that by our assumptions, none of the rules r_p, $1 \leqslant p \leqslant s$, introduces any of the A_q, $1 \leqslant q \leqslant p-1$. Moreover, none of the sequences of productions π_p, $1 \leqslant p \leqslant s+1$, introduces any of the non-terminals A_q, $1 \leqslant q \leqslant p-1$. Yet it might be the case that an r_p (for a p, $1 \leqslant p \leqslant s$) introduces some non-terminals A_l, $p+1 \leqslant l \leqslant s$, or a π_p $(1 \leqslant p \leqslant s+1)$ introduces some A_l, $p+1 \leqslant l \leqslant s$. Therefore, assume for each p, $1 \leqslant p \leqslant s$, and for each l, $p+1 \leqslant l \leqslant s$, that production r_p introduces b_{lp} occurrences of the non-terminal A_l, where $b_{lp} \geqslant 0$.

Let $x_0 \in (N \cup T)^*$ be a string with $M_{x_0} = (a_1, \ldots, a_m, i, \bar{j})$, which is actually $M_{x_0} = (a_1, \ldots, a_s, 0, \ldots, 0, i, \bar{j})$, and let $x \in T^*$ be the string such that $x_0 \underset{\Gamma}{\overset{\pi}{\Rightarrow}} x$.

Let G_{i_1} be the component which uses the instance of r_1 pointed out in the sequence π. Several erasing rules (out of $r_1, \ldots, r_s$) can be applied in the same derivation in a component grammar (a derivation in a component meaning k rewriting steps). Then let $r_1, \ldots, r_q$, for $q \leqslant 1$, be all the erasing productions that are applied in the same activation of G_{i_1}. We can write

$$\pi = \pi_1' \pi_1'' r_1 \pi_2 r_2 \ldots \pi_q r_q \pi_{q+1}' \pi_{q+1}'' r_{q+1} \pi_{q+2} \ldots \pi_s r_s \pi_{s+1}.$$

Denote $\sigma_1 = \pi_1'' r_1 \pi_2 r_2 \ldots \pi_q r_q \pi_{q+1}'$, $|\sigma_1| = k$. Let

$$x_0 \underset{\Gamma}{\overset{\pi_{1'}}{\Rightarrow}} x_1 \underset{G_{i_1}}{\overset{\sigma_1}{\Rightarrow}} x_2.$$

It then follows

$$M_{x_2} = (0, \ldots, 0, a_{q+1} - \Delta(\pi_1'\sigma_1)(q+1), \ldots, a_s - \Delta(\pi_1'\sigma_1)(s), 0, \ldots, 0, i_1, k),$$

where the non-zero coordinates are on the positions $q+1, q+2, \ldots, s$ (and, of course, $m+1, m+2$). One can observe $\Delta(\pi_1'\sigma_1)(p) = -a_p$, for each p, such that $1 \leqslant p \leqslant q$.

We then consider the derivation (which is a valid derivation in Γ)

$$\alpha \underset{\Gamma}{\overset{\pi_{1'}}{\Rightarrow}} \beta_1 \underset{G_{i_1}}{\overset{\sigma_1}{\Rightarrow}} \beta_2 \underset{G_{i_1}}{\overset{r_1^k}{\Rightarrow}} \beta_{21} \underset{G_{i_1}}{\overset{r_1^k}{\Rightarrow}} \beta_{22} \underset{G_{i_1}}{\overset{r_1^k}{\Rightarrow}} \cdots \underset{G_{i_1}}{\overset{r_1^k}{\Rightarrow}} \beta_{2l_1},$$

where $x \underset{G_{i_j}}{\overset{r_j^k}{\Rightarrow}} y$ is the notation for a rewriting of the string x into string y in the grammar G_{i_j}, by applying k times production r_j (for any $1 \leqslant j \leqslant q$). Then

$$\begin{aligned} M_{\beta_{2l_1}} = (&0, (l_2 + l_1 b_{12})k, (l_3 + l_1 b_{13})k, \ldots, (l_q + l_1 b_{1q})k, \\ &(l_{q+1} + l_1 b_{1,q+1})k + a_{q+1} - \Delta(\pi_1' \sigma_1)(q+1), \ldots, \\ &(l_s + l_1 b_{1s})k + a_s - \Delta(\pi_1' \sigma_1)(s), 0, \ldots, 0, i_1, k). \end{aligned}$$

We continue by activating $u = l_2 + l_1 b_{12}$ times in a row the component grammar G_{i_1} (this means $(l_2 + l_1 b_{12})k$ derivation steps), using only the production r_2, i.e. we derive

$$\beta_{2l_1} \underset{G_{i_1}}{\overset{r_1^k}{\Rightarrow}} \beta_{31} \underset{G_{i_1}}{\overset{r_1^k}{\Rightarrow}} \beta_{32} \underset{G_{i_1}}{\overset{r_1^k}{\Rightarrow}} \cdots \underset{G_{i_1}}{\overset{r_1^k}{\Rightarrow}} \beta_{3u},$$

thus resulting in the string β_{3u} with

$$\begin{aligned} M_{\beta_{3u}} = (&0, 0, (l_3 + l_2 + l_1 b_{13} + l_1 b_{12})k, \ldots, (l_q + l_2 + l_1 b_{1q} + l_1 b_{12})k, \\ &(l_{q+1} + l_2 + l_1 b_{1,q+1} + l_1 b_{12})k + a_{q+1} - \Delta(\pi_1' \sigma_1)(q+1), \ldots, \\ &(l_s + l_2 + l_1 b_{1s} + l_1 b_{12})k + a_s - \Delta(\pi_1' \sigma_1)(s), 0, \ldots, 0, i_1, k). \end{aligned}$$

We continue in the same way to erase symbols $A_3, A_4, \ldots, A_q$, resulting in a string γ for which

$$\begin{aligned} M_\gamma = (&0, \ldots, 0, l_1' k + a_{q+1} - \Delta(\pi_1' \sigma_1)(q+1), \ldots, l_s' k \\ &+ a_s - \Delta(\pi_1' \sigma_1)(s), 0, \ldots, 0, i_1, k) \end{aligned}$$

(the non-zero components are on positions $q+1, \ldots, s$, and, of course, $m+1, m+2$).

One can note that the string γ is in the same relation with the string x_2 as the string α is with x_0. By repeating the algorithm above with x_2 instead of x_0 and with γ instead of α, after a finite number of steps we obtain a string w satisfying $M_w = (0, \ldots, 0, i', k)$, and hence that way we generate a derivation

$$\alpha \underset{\Gamma}{\overset{\bar{\sigma}}{\Rightarrow}} w \in T^*,$$

where $\bar{\sigma}$ is an applicable sequence of productions. This proves the assertion for the $=k$ derivation mode (point i).

For the other derivation modes, the proofs can be done in a similar manner. It is to be noted that the derivation considered is context free, and therefore at each application of an erasing production rule, exactly one non-terminal occurrence is eliminated from the string in question.

For the case of $\geqslant k, *, t$ $(k \geqslant 1)$ derivation modes, the erasing of any of the non-terminals will be performed during the same activation of the component grammar which erases it (for the last time) in the derivation $x_0 \overset{\pi}{\underset{\Gamma}{\Rightarrow}} x \in T^*$ (hence, in the above notations, grammar G_{i_1} will be activated only once (but for a higher number of steps) to erase all of $A_1, A_2, \ldots, A_q$).

For the derivation in the $\leqslant k$, $k \geqslant 1$, mode, the erasing of the additional non-terminals (of α in comparison to x_0) can be done by activating several (the necessary number of) times, in a row, the components which erase the non-terminals in the derivation $x_0 \overset{\pi}{\underset{\Gamma}{\Rightarrow}} x \in T^*$, each activation being for only one derivation step. □

The construction of the terminal coverability tree is validated by the next two theorems.

Theorem 6. *Let Γ be a CD grammar system working in a derivation mode $f \in \{\geqslant k, \leqslant k, *, t \mid k \geqslant 1\}$, and let $\overline{\mathscr{T}}(\Gamma) = (\overline{V}, \overline{E}, l_1, l_2)$ be its terminal coverability tree. Then for any production r labelling an edge of $\overline{\mathscr{T}}$, there exists a terminal derivation in Γ which uses production r.*

Proof. Due to $r \in \overline{E}$, there exist nodes $v_1', v_2' \in \overline{V}$, with v_2' direct descendant of v_1', such that $r = l_2(v_1', v_2')$. Two situations might occur:

(a) v_2' has a descendant v with $l_1(v) = (0, \ldots, 0, i', \bar{k})$, for an i', $1 \leqslant i' \leqslant n$, where $\bar{k}$ is related to the derivation mode as in Theorem 5.

Let $d_{\overline{\mathscr{T}}}(v_0, v) = v_0, v_1, \ldots, v_p$, with $v_p = v$ and let $l_2(v_{i-1}, v_i) = r_i$, for any i, $1 \leqslant i \leqslant p$. This implies $v_1' = v_j, v_2' = v_{j+1}, r = r_{j+1}$, for a $j, 0 \leqslant j \leqslant p-1$. It also follows that there exists a derivation

$$S \overset{r_1}{\underset{\Gamma}{\Rightarrow}} x_1 \overset{r_2}{\underset{\Gamma}{\Rightarrow}} x_2 \overset{r_3}{\underset{\Gamma}{\Rightarrow}} \cdots \overset{r_p}{\underset{\Gamma}{\Rightarrow}} x_p$$

(according to the f mode of derivation) with $M_{x_i} = l_1(v_i)$, for any i, $0 \leqslant i \leqslant p$. In particular, $M_{x_p} = l_1(v_p) = l_1(v)$ and, therefore, $x_p \in T^*$ follows.

But due to $r = r_{j+1}$, the derivation above is actually

$$S \overset{r_1}{\underset{\Gamma}{\Rightarrow}} x_1 \overset{r_2}{\underset{\Gamma}{\Rightarrow}} \cdots \overset{r_j}{\underset{\Gamma}{\Rightarrow}} x_j \overset{r}{\underset{\Gamma}{\Rightarrow}} x_{j+1} \cdots \overset{r_{j+2}}{\underset{\Gamma}{\Rightarrow}} \cdots \overset{r_p}{\underset{\Gamma}{\Rightarrow}} x_p \in T^*,$$

hence is a terminal derivation (of type f) in Γ, using production r.

(b) v_2' does not have any descendant labelled $(0, \ldots, 0, i', \bar{k})$, $1 \leqslant i' \leqslant n$.

Let us extend the path $d_{\overline{\mathscr{T}}}(v_0, v_2')$ in $\overline{\mathscr{T}}(\Gamma)$ to a path $d_{\overline{\mathscr{T}}}(v_0, v)$, where v is a leaf node (there can be several such extensions; we just randomly choose one of them; of interest is that for all of them v_1', $v_2' \in d_{\overline{\mathscr{T}}}(v_0, v)$). From the construction of $\overline{\mathscr{T}}(\Gamma)$, one can see that the only possibility is that v has been marked as to remain in $\overline{\mathscr{T}}(\Gamma)$ at Step 5 of the algorithm. More precisely, we have

(i) $l_1(v)$ contains ω-components;

(ii) there exists node $\bar{v}$ on the path $d_{\overline{\mathscr{T}}}(v_0, v)$, $l_1(\bar{v})$ without ω-components, $l_1(\bar{v})$ coincides with $l_1(v)$ in the finite components of $l_1(v)$;

(iii) considering v' the closest node to the root among such nodes $\bar{v}$ (i.e. $l_1(v') \in \mathbb{N}^{m+2}$, $l_1(v')(p) = l_1(v)(p)$ for any p, $1 \leqslant p \leqslant m+2$, for which $l_1(v)(p) \neq \omega$), we have v' has a descendant $v'', l_1(v'') = (0, \ldots, 0, i', \bar{k})$.

Similarly, as in the proof of Theorem 2, one can construct a derivation starting from S, using all the production rules labelling the path $d_{\mathscr{T}}(v_0, v)$, by "pumping" the sequences of productions that lead to the introduction of the ω-components of $l_1(v)$, such that the resulting string has the associated vector in $\mathbb{N}^{m+2}$ greater or equal than $l_1(v')$, i.e., one can construct a derivation

$$S \underset{\Gamma}{\overset{\sigma_1}{\Rightarrow}} \alpha_1 \underset{\Gamma}{\overset{r}{\Rightarrow}} \alpha_2 \underset{\Gamma}{\overset{\sigma_2}{\Rightarrow}} \alpha,$$

with $M_\alpha \geqslant l_1(v')$, $M_\alpha(s) = l_1(v')(s)$, $m+1 \leqslant s \leqslant m+2$, where σ_1, σ_2 are sequences of productions, such that the derivation above is with respect to the f mode.

In addition, since $l_1(v')(p) = l_1(v)(p)$ for any p, $1 \leqslant p \leqslant m$, for which $l_1(v)(p) \neq \omega$, and since the "pumping" procedure does not affect the components non-equal to ω, it follows $M_\alpha(p) = l_1(v')(p) = l_1(v)(p)$, for any p, $1 \leqslant p \leqslant m$, with $l_1(v)(p) \neq \omega$. In particular, for any q, $1 \leqslant q \leqslant m$, such that $l_1(v')(q) = 0$, also $M_\alpha(q) = 0$. By Theorem 5, it then follows that there exists derivation $\alpha \underset{\Gamma}{\overset{\sigma_3}{\Rightarrow}} \beta \in T^*$, with respect to the f mode (σ_3 is a sequence of productions). By combining the two fragments of derivation, we obtain

$$S \underset{\Gamma}{\overset{\sigma_1}{\Rightarrow}} \alpha_1 \underset{\Gamma}{\overset{r}{\Rightarrow}} \alpha_2 \underset{\Gamma}{\overset{\sigma_2}{\Rightarrow}} \alpha \underset{\Gamma}{\overset{\sigma_3}{\Rightarrow}} \beta \ \in T^*,$$

where the entire derivation is with respect to the f mode. □

Remark. The proof of Theorem 6 does not work for the $=k$ derivation mode since in this case it is not enough to generate an α with $M_\alpha \geqslant l_1(v')$ (in the notations of the proof above). We would need M_α to be as in the statement of Theorem 5, point i), or at least the sum of the non-terminals introduced in α in comparison to a string x with $M_x = l_1(v')$ would have to be divisible by k.

Theorem 7. *Assume that the production rule r is used in a terminal derivation in a CD grammar system Γ working according to the derivation mode $f \in \{=k, \leqslant k, \geqslant k, *, t \mid k \geqslant 1\}$. Hence,*

$$S \underset{\Gamma}{\overset{\sigma_1}{\Rightarrow}} w_1 \underset{\Gamma}{\overset{r}{\Rightarrow}} w_2 \underset{\Gamma}{\overset{\sigma_2}{\Rightarrow}} w_3 \in T^*,$$

where σ_1, σ_2 are sequences of productions, such that the derivation above is with respect to the f mode. Then there exists an edge (v_1, v_2) in $\overline{\mathscr{T}}(\Gamma)$, such that $l_2(v_1, v_2) = r$.

Proof. Due to Theorem 4, there exists in $\mathscr{T}(\Gamma)$ an edge $(v_1, v_2) \in E$ labelled by r, $l_2(v_1, v_2) = r$. We prove by contradiction that we have not eliminated it while constructing $\overline{\mathscr{T}}(\Gamma)$ from $\mathscr{T}(\Gamma)$. Hence, let us assume that we have cut this occurrence of the production r as a label of an edge in $\mathscr{T}(\Gamma)$. In other words, we have eliminated, in the construction of $\overline{\mathscr{T}}(\Gamma)$, the node v_2 together with the edge (v_1, v_2) (yet other occurrences of production r as labels of an edge in $\mathscr{T}(\Gamma)$ might be preserved in $\overline{\mathscr{T}}(\Gamma)$).

Let LV be the set of all leaf nodes v such that v_1, v_2 are nodes on the path $d_{\mathscr{T}}(v_0, v)$. One can note that if all nodes $v \in LV$ have the associated vector $l_1(v) \in \mathbb{N}^{m+2}$ then, according to the construction of $\overline{\mathscr{T}}(\Gamma)$, in order to eliminate node v_2 from $\mathscr{T}(\Gamma)$, it must be the case that for any $v \in LV$, one cannot find $i, \bar{k}$, such that $l_1(v) = (0, \dots, 0, i, \bar{k})$.

But then obviously any sequence of productions labelling a path $d_{\mathscr{T}}(v_0, v)$ (for any $v \in LV$) leads to the blocking of the derivation, and therefore, none of them can represent the productions used in a terminal derivation.

Therefore, due to the existence of a terminal derivation using production r, there must exist in $\mathscr{T}(\Gamma)$ at least one edge (v_1', v_2'), labelled r, such that at least one leaf node, v', on an extension of the path $d_{\mathscr{T}}(v_0, v_2')$ contains ω-components in the associated vector.

From the proof of Theorem 4, one can see that the derivation in the hypothesis of the present theorem must be performed by using productions on the path $d_{\mathscr{T}}(v_0, v')$, with "returning points" at the nodes where components ω have been introduced. By analyzing the cases when the edge (v_1', v_2') could be eliminated in the construction of $\overline{\mathscr{T}}(\Gamma)$, one can observe that none of the paths mentioned earlier can lead to a terminal derivation.

Since our assumption has lead to contradictions in all cases, there must exist an edge labelled r in $\overline{\mathscr{T}}(\Gamma)$. □

From Theorems 6 and 7 the following property results. It gives an exhaustive characterization of the "reachability" of a production.

Corollary 1. *Let Γ be a CD grammar system, let the derivation mode be $f \in \{\geqslant k, \leqslant k, *, t \mid k \geqslant 1\}$, and let r be a production of Γ. Then r is used in a terminal derivation in Γ if and only if there exists an edge labelled by r in $\overline{\mathscr{T}}(\Gamma)$.*

5. Decidability results

Consider a CD grammar system Γ as in Definition 1, and denote $D = \{*, t\} \cup \{\leqslant k, \geqslant k \mid k \geqslant 1\}$. By Theorem 4 (when *all* the derivations in a system are of concern), and by Corollary 1 (in case of *terminal* derivations), the decidability of the basic, rather challenging problems follows.

Corollary 2. (i) *It is decidable whether a production $r \in P_i$ (for an $i, 1 \leqslant i \leqslant n$), of a component grammar in a given CD grammar system is enabled in the system, for any derivation mode.*

(ii) *It is decidable whether a production $r \in P_i$ (for an $i, 1 \leqslant i \leqslant n$), of a component grammar in a given CD grammar system working in one of the modes $f \in D$ is enabled in the terminal derivations of the system.*

Corollary 3. (i) *It is decidable whether or not a CD grammar system Γ is non-terminally bounded with respect to its sentential forms, for any derivation modes in the system. In the affirmative case, the constant s such that any sentential form of Γ contains at most s non-terminal symbols can be effectively determined.*

(ii) *It is decidable whether or not a CD grammar system Γ is non-terminally bounded with respect to the sentential forms produced during terminal derivations, for any mode $f \in D$. In the affirmative case, the constant s for which any such sentential form of Γ contains at most s non-terminals can be effectively determined.*

Proof. We check whether or not in the (terminal) coverability tree of Γ there exist nodes in whose labels components ω appear. If so, then the system is not non-terminal bounded with respect to its sentential forms. Otherwise, the desired constant s is the maximum among the coordinates in the labels of the nodes in the tree. $\square$

Corollary 4. (i) *It is decidable whether a component grammar of a CD grammar system, working in any of the derivation modes, is ever active in a derivation in the system.*

(ii) *It is decidable whether a component grammar of a CD grammar system, deriving in a mode $f \in D$, is ever active in a terminal derivation in the system.*

Proof. We check in the (terminal) coverability tree of the system whether there exist edges (v, v'), such that $l_2(v, v')(m+1) = i$, where i is the index of the component grammar we are interested in. If so, then the answer to the problem is affirmative, otherwise it is negative. $\square$

Corollary 5. (i) *It is decidable whether a component grammar is activated only a bounded number of times in any of the derivations in a CD grammar system, for any derivation mode.*

(ii) *It is decidable whether a component grammar is activated only a bounded number of times in any of the terminal derivations in a CD grammar system, for any derivation mode $f \in D$.*

Proof. We check in the (terminal) coverability tree whether there exists a node v with $l_1(v)(m+1) = i$, such that v has a descendant in whose label components ω occur or there exist nodes v', v'', with $l_1(v') = l_1(v'')$, such that v, v' are nodes on the path from v_0 to v'' (again i is the index of the component grammar we are interested in). If so, then the answer to the problem is negative, otherwise is positive. $\square$

Corollary 6. (i) *It is decidable whether for a component G_i $(1 \leqslant i \leqslant n)$ of a CD grammar system working in any of the derivation modes, and for any $q \geqslant 1$, there* *exists a derivation in the system such that G_i is activated at least q times during that derivation.*

(ii) *It is decidable whether for a component G_i of a CD grammar system working in any of the derivation modes $f \in D$, and for any $q \geqslant 1$, there exists a terminal derivation in the system such that G_i is activated at least q times during that derivation.*

Proof. We check in the (terminal) coverability tree whether there exists a node v with $l_1(v)(m+1) = i$, such that v is on a path which led to the introduction of a component ω or v is on a path (v', v''), with $l_1(v') = l_1(v'')$. □

Observe that Corollary 2 can be used to reduce a CD grammar system.

Corollary 7. *Given a CD grammar system Γ, one can effectively construct a CD grammar system Γ' which generates the same language with respect to a derivation mode $f \in D$, such that all productions and all non-terminals of Γ' are used in terminal derivations.*

References

[1] A. Atanasiu, V. Mitrana, The modular grammars, Internat. J. Comput. Math. 30 (1989) 17–35.

[2] E. Csuhaj-Varju, J. Dassow, On cooperating distributed grammar systems, J. Inform. Process. Cybernet. EIK 26 (1990) 49–63.

[3] E. Csuhaj-Varju, J. Dassow, J. Kelemen, Gh. Păun, Grammar systems: A Grammatical Approach to Distribution and Cooperation, Gordon and Breach, London, 1994.

[4] J. Dassow, Gh. Păun, Regulated Rewriting in Formal Language Theory, Springer, Berlin, 1989.

[5] J. Dassow, Gh. Păun, G. Rozenberg, Grammar systems, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Springer, Berlin, 1997, pp. 155–213.

[6] R. Karp, R. Miller, Parallel program schemata, J. Comput. System Sci. 3 (1969) 147–195.

[7] R. Meersman, G. Rozenberg, Cooperating grammar systems, in: Proc. MFCS'78 Symp., Lecture Notes in Computer Science, vol. 64, Springer, Berlin, 1978, pp. 364–374.

[8] Gh. Păun, L. Sântean, Parallel communicating grammar systems: the regular case, Ann. Univ. Buc. Ser. Matem.-Inform. 38 (1989) 55–63.

[9] J.L. Peterson, Petri Net Theory and the Modeling of Systems, Prentice-Hall, London, 1981.

[10] W. Reisig, Petri Nets. An Introduction, Springer, Berlin, 1985.

[11] A. Salomaa, Formal Languages, Academic Press, New York, 1973.

[12] F.L. Ţiplea, C. Ene, A coverability structure for parallel communicating grammar systems, J. Inform. Process. Cybernet. EIK 29 (5) (1993) 303–315.