Deterministic Cooperating Distributed Grammar Systems

Valeria Mihalache
Victor Mitrana

Faculty of Mathematics, University of Bucharest
Str. Academiei 14, 70109, Bucharest, ROMANIA

Turku Centre for Computer Science
TUCS Technical Report No 63
November 1996

ISBN 951-650-886-3
ISSN 1239-1891

Abstract

Subclasses of grammar systems that can facilitate parser construction appear to be of interest. In this paper, some syntactical conditions considered for strict deterministic grammars are extended to cooperating distributed grammar systems, restricted to the terminal derivation mode. Two variants are considered according to the level to which the conditions address. The local variant, which introduces strict deterministic restrictions for each component of the system apart, results in local unambiguity of the derivations. Rather unexpected, the total variant, which extends the strict deterministic constraints at the level of the entire system, results in some cases in global unambiguity of the derivations.

TUCS Research Group
Mathematical Structures of Computer Science

1 Introduction

Cooperating distributed (CD, for short) grammar systems have been introduced in [3]. A similar generating device was considered in [11], while a particular variant of it appears in [1]. Most of the results known in this area until the middle of 1992 can be found in [4], while newer results are surveyed in [7].

However, there are still lots of classical topics in formal languages theory or in related areas which have not been studied so far in the grammar systems set-up. Constructing parsers is such a topic, which is not only of theoretical interest, but it will make grammar systems more appealing to researchers in applied computer science as well, since it will open the possibility of using grammar systems in domains where just Chomsky-like grammars are currently used (for instance, in natural language processing, or in compiler construction). This will clearly bring to the user all the advantages of having a model which can cope with such phenomena as cooperation and distribution of the work carried out by several processors.

Of interest to this aim are the results of [2] and [6]. Thus, [2] approaches CD grammar systems from the accepting point of view, comparing their accepting capacity to their own generating capacity, or to that of other classes of grammars in the regulated rewriting area. [6] considers pushdown automata systems, with the scope of characterizing the languages generated by grammar systems in terms of recognizers; but their model turns out to be much more powerful from the point of view of the recognized languages class.

We believe that a more involved study of the derivations in a CD grammar system would be very useful to a possible parser constructor for the languages generated by grammar systems. This is the aim of the present paper: we address subclasses of CD grammar sysetms which can facilitate a parser construction, due to some unambiguity properties of the derivations in such systems.

More precisely, the present paper studies the effect on CD grammar systems of syntactical constraints similar to those considered for strict deterministic context-free grammars. It is known that the family of languages generated by strict deterministic context-free grammars is the same as the family of languages generated by $LR(0)$ grammars (see Theorem 11.5.5 in [8] and Theorem 10.12 in [9]), which are ones of the most liked class of grammars for parsing. Therefore, our intention was to see what happens if the conditions for strict deterministic grammars are extended to CD grammar systems. To our great surprise, we obtained that the *unambiguity* of the derivations holds for some classes of grammar systems as well.

When introducing the restrictions for strict determinism in CD grammar systems, two variants should be taken into consideration, depending on the level, *local/global*, to which the restrictions address. In the local level case, the generative capacity of the systems remains the same, but the behaviour of each component is unambiguous. The generative power decreases when global level is considered, whereas for some more restrictive classes the derivation is totally unambiguous.

2 Definitions and examples

We assume the reader accustomed to the basic facts in formal language theory [9]. For details concerning the grammar systems we refer to [4].

For an alphabet V, we denote by V^* the set of all words over V and by λ the empty word; moreover, $V^+ = V^* - \{\lambda\}$, while $|V|$ stands for the cardinality of V. For a string x, denote by $|x|$ the length of x and by $Pref_k(x)$ the prefix of length k of $x, |x| \geq k$.

If π is a partition of V, then we write $a \sim^\pi b$ iff there exists $M \in \pi$ such that $\{a, b\} \subseteq M$.

A *cooperating distributed grammar system* is a construct

$$\Gamma = (N, T, S, P_1, P_2, \ldots, P_n),$$

where N, T are disjoint alphabets, and $P_i, 1 \leq i \leq n$, are finite sets of context-free rules over $N \cup T$.

The sets P_i are called the *components* of Γ. (If we want to point out *grammars* as components, then we can consider grammars without axioms, of the form $G_i = (N, T, P_i), 1 \leq i \leq n$.)

For a component P_i of a grammar system Γ as above, we denote $dom(P_i) = \{A \in N \mid A \rightarrow x \in P_i\}$, and $T_i = (N \cup T) - dom(P_i)$.

Let $\Gamma = (N, T, S, P_1, P_2, \ldots, P_n)$ be a CD grammar system as above and let π_i be a partition of $N \cup T$, for any $1 \leq i \leq n$. The partition π_i is a *local strict partition* if the following conditions hold:

a) $T \in \pi_i$,

b) $T_i - T \in \pi_i$,

c) for any $A, A' \in dom(P_i)$, if $A \sim^{\pi_i} A'$ and $A \longrightarrow xy, A' \longrightarrow xy' \in P_i$, $x, y, y' \in (N \cup T)^*$, then we have either

$(i) y, y' \neq \lambda$ and $Pref_1(y) \sim^{\pi_i} Pref_1(y')$,

or

$(ii) y = y' = \lambda$ and $A = A'$.

We say that a component P_i is a *strict deterministic* component of Γ iff there exists a local strict partition π_i.

The system Γ is said to be *local strict deterministic* iff all its components are strict deterministic.

If Γ is a CD grammar system and π is a partition of $N \cup T$, then π is a *total strict partition* iff:

$a)$ $T \in \pi$,

$b)$ for any $A, A' \in N$ if $A \sim^\pi A'$ and $A \longrightarrow xy, A' \longrightarrow xy' \in \bigcup_{i=1}^{n} P_i$, $x, y, y' \in (N \cup T)^*$, then we have either
$(i) y, y' \neq \lambda$ and $Pref_1(y) \sim^\pi Pref_1(y')$,
or
$(ii) y = y' = \lambda$ and $A = A'$.

The system Γ is said to be *total strict deterministic* iff there exists a total strict partition of $N \cup T$.

We will latter on provide grammar systems which are total strict deterministic but not local strict deterministic, and vice versa.

Among the derivation modes in CD grammar systems, our concern here is restricted to the *terminal* (t) mode. Therefore, we next recall only the definition of this one.

On $(N \cup T)^*$ one can define the usual one step derivation with respect to P_i, denoted by $\underset{P_i}{\Longrightarrow}$. Formally, $x \underset{P_i}{\Longrightarrow} y$ iff $x = x_1 A \alpha,\ y = x_1 z \alpha,\ x_1 \in T_i^*,\ z, \alpha \in (N \cup T)^*$, and $A \longrightarrow z \in P_i$. The reflexive and transitive closure of the relation $\underset{P_i}{\Longrightarrow}$ is denoted by $\overset{*}{\underset{P_i}{\Longrightarrow}}$. Furthermore, we write $x \overset{t}{\underset{P_i}{\Longrightarrow}} y$ if $z \overset{*}{\underset{P_i}{\Longrightarrow}} y$ and there is no $z \in (N \cup T)^*$ such that $y \underset{P_i}{\Longrightarrow} z$.

We will consider in the following that all the derivations involved are leftmost derivations with respect to the component grammar which is active at that moment. This is not a restriction at all, since for the terminal derivation mode in a CD grammar system, leftmost derivation is equivalent to usual derivation, from the point of view of the strings generated. We use the same notation, $\underset{P_i}{\Longrightarrow}$, for the leftmost derivation with respect to P_i $(1 \leq i \leq n)$.

The language generated by the system Γ is

$$\begin{aligned} L(\Gamma) \ = \ & \{w \mid w \in T^*,\ S \overset{t}{\underset{P_{i_1}}{\Longrightarrow}} w_1 \overset{t}{\underset{P_{i_2}}{\Longrightarrow}} \ldots \overset{t}{\underset{P_{i_m}}{\Longrightarrow}} w_m = w, \\ & m \geq 1, 1 \leq i_j \leq n, 1 \leq j \leq m\}. \end{aligned}$$

Example 1.

$$\Gamma_1 = (\{S, A, C, X, Y\}, \{a, b, c\}, S, P_1, P_2, P_3)$$

where

$$\begin{aligned} P_1 &= \{S \longrightarrow AC, X \longrightarrow A, Y \longrightarrow C\}, \\ P_2 &= \{A \longrightarrow aXb, C \longrightarrow cY\}, \\ P_3 &= \{A \longrightarrow ab, C \longrightarrow c\}. \end{aligned}$$

One can observe that Γ_1 is local strict deterministic under the local partitions

$$\begin{aligned} \pi_1 &= \{\{S\}, \{A, C\}, \{X, Y\}, \{a, b, c\}\}, \\ \pi_2 = \pi_3 &= \{\{S, X, Y\}, \{A, C\}, \{a, b, c\}\}, \end{aligned}$$

but it is not total strict deterministic.

The language generated by Γ_1 is $\{a^n b^n c^n | n \geq 1\}$.

Example 2.

$$\Gamma_2 = (\{S, A, B, C, D\}, \{a, b, c\}, S, P_1, P_2)$$

where

$$\begin{aligned} P_1 &= \{S \longrightarrow aA, S \longrightarrow aB, A \longrightarrow aAa, A \longrightarrow bC\}, \\ P_2 &= \{B \longrightarrow aB, B \longrightarrow bD, C \longrightarrow bC, C \longrightarrow a, D \longrightarrow bDc, D \longrightarrow c\}. \end{aligned}$$

The grammar system above is total strict deterministic under the partition

$$\pi = \{\{S\}, \{A, B\}, \{C, D\}, \{a, b, c\}\}$$

which is a total strict partition. However, its first component is not deterministic.

The generated language is

$$L = \{a^n b^k a^n | n, k \geq 1\} \cup \{a^n b^k c^k | n, k \geq 1\}.$$

3 Properties of derivations

Due to the explicit motivation for the introduction of local/total strict deterministic restrictions on CD grammar systems, we first study the derivations in such systems.

In the local case, we obtain that all the components of the system are unambiguous. This is not the situation anymore for total strict deterministic CD grammar systems. However, also in this case the deterministic constraints have clear impact on the derivations: one can eliminate left-recursion from a total strict deterministic CD grammar system. Moreover, if the system is such that no production rule belongs to two different components, then the unambiguity of the derivations (properly defined) at the level of the system holds as well.

A few additional notations and definitions are to be first specified.

Let $\Gamma = (N, T, S, P_1, P_2, \ldots, P_n)$ be a CD grammar system whose rules are labelled. Let $Lab(P_i)$ be the set of labels of the rules in P_i. For a derivation $D : x \overset{t}{\underset{P_i}{\Longrightarrow}} y$ we denote by $C(D)$ the control word associated to the derivation D (that is, the word consisting of the labels of productions used in the derivation D, in the corresponding order). Therefore, $C(D)$ is a string in $(Lab(P_i))^*$ that represents the sequence of rules of P_i used in the derivation D.

A component P_i is *ambiguous* if there exist two derivations $D_1 : x \overset{t}{\underset{P_i}{\Longrightarrow}} y$ and $D_2 : x \overset{t}{\underset{P_i}{\Longrightarrow}} y$ with $C(D_1) \neq C(D_2)$, for some $x, y \in (N \cup T)^*$. Otherwise, the component is called unambiguous.

For a word $w \in L(\Gamma)$, we say that a derivation

$$S = x_0 \overset{t}{\underset{P_{i_1}}{\Longrightarrow}} x_1 \overset{t}{\underset{P_{i_2}}{\Longrightarrow}} x_2 \overset{t}{\underset{P_{i_3}}{\Longrightarrow}} \ldots \overset{t}{\underset{P_{i_s}}{\Longrightarrow}} x_s = w$$

is a *leftmost derivation* of w if for any j, $1 \leq j \leq s$, the derivation in the component P_{i_j} is such that at the first step the very leftmost nonterminal symbol in x_{j-1} is rewritten, and afterwards, the leftmost possible nonterminal in the resulted sentential form is rewritten by P_{i_j}. Formally, the following conditions hold:

i) $x_{j-1} = \alpha_{j-1} A_j \beta_{j-1}$, for $\alpha_{j-1} \in T^*$,

ii) the first production used in the derivation $x_{j-1} \overset{t}{\underset{P_{i_j}}{\Longrightarrow}} x_j$ is of the form $A_j \to \gamma_{j-1} \in P_{i_j}$, resulting in the string $z = \alpha_{j-1}\gamma_{j-1}\beta_{j-1}$,

iii) afterwards, the derivation $z \overset{t}{\underset{P_{i_j}}{\Longrightarrow}} x_j$ proceeds by rewriting at any step the leftmost possible nonterminal in the current sentential form.

In order to point out that a derivation is leftmost, we will use the notation tl (terminal leftmost) instead of t, when we specify the derivation mode.

A CD grammar system Γ is said to be *unambiguos* if for any $w \in L(\Gamma)$ and for any two leftmost derivations of w in Γ,

$$D_1: \; S = x_0 \underset{P_{i_1}}{\overset{tl}{\Longrightarrow}} x_1 \underset{P_{i_2}}{\overset{tl}{\Longrightarrow}} \dots \underset{P_{i_s}}{\overset{tl}{\Longrightarrow}} x_s = w,$$

$$D_2: \; S = x'_0 \underset{P_{j_1}}{\overset{tl}{\Longrightarrow}} x'_1 \underset{P_{j_2}}{\overset{tl}{\Longrightarrow}} \dots \underset{P_{j_l}}{\overset{tl}{\Longrightarrow}} x'_l = w,$$

the following hold:

$$\begin{array}{ll} (i) & s = l, \\ (ii) & P_{i_r} = P_{j_r}, \; 1 \leq r \leq s, \\ (iii) & x_r = x'_r, \; 0 \leq r \leq s \end{array}$$

We shall shortly say that $D_1 = D_2$ holds.

Remark that the definition of leftmost derivation does not imply that for any $w \in L(\Gamma)$ a leftmost derivation of w exists.

We can now proceed in analysing the derivations in a CD grammar system. We first consider the case of local strict deterministic systems. The next theorem states the *local unambiguity* of such systems.

Theorem 1 *Any local strict grammar system has only unambiguous components.*

Proof. Let $\Gamma = (N, T, S, P_1, \dots, P_n)$ be a CD grammar system, local strict deterministic under the partitions $\pi_i, 1 \leq i \leq n$. We have to prove that for any i, $1 \leq i \leq n$, and for any two derivations

$$\begin{array}{ll} D_1: & \alpha \underset{P_i}{\overset{t}{\Longrightarrow}} x \\ D_2: & \alpha \underset{P_i}{\overset{t}{\Longrightarrow}} x, \end{array}$$

the relation $C(D_1) = C(D_2)$ holds.

We need the following fact which we give without proof. The reader interested in details may consult [8] (Lemma 11.4.1), since the corresponding proof there can be adapted very easily for our situation.

Fact: For any $A, A' \in N$ with $A \sim^{\pi} A'$ such that $A \underset{P_i}{\overset{k}{\Longrightarrow}} xy$ and $A' \underset{P_i}{\overset{k}{\Longrightarrow}} xy'$, one of the following situations holds:

i) $y, y' \neq \lambda$, and $Pref_1(y) \sim^{\pi} Pref_1(y')$,

ii) $y = y' = \lambda$, and $A = A'$.

Here $\overset{k}{\underset{P_i}{\Longrightarrow}}$ means a k-steps derivation in the component P_i. We shall prove now our theorem by induction on the length of α.

Suppose $|\alpha| = 1$, hence $\alpha = A$. Assume that each of the derivations D_i, $1 \leq i \leq 2$, uses q_i productions, respectively. Then $q_1 = q_2$ must hold. Indeed, if $q_1 < q_2$ or $q_2 < q_1$, by the above fact, a symbol in $dom(P_i)$ would result equivalent, with respect to π_i, to a symbol in T_i, which contradicts the definition of local strict determinism. Denote $q = q_1 = q_2$.

We continue the proof by induction on q. The assertion is trivially true for $q = 1$. Assume that the assertion is true for any derivations of length $q-1$, and let

$$\begin{aligned} D_1: &\quad A \overset{q-1}{\underset{P_i}{\Longrightarrow}} x_1 B x_2 \underset{P_i}{\Longrightarrow} x \\ D_2: &\quad A \overset{q-1}{\underset{P_i}{\Longrightarrow}} x_1' B' x_2' \underset{P_i}{\Longrightarrow} x. \end{aligned}$$

Clearly, $x_1 = x_1'$ (since the derivation is leftmost) and $B \sim_i^{\pi} B'$.

Assume now that $x_1 B x_2 \underset{P_i}{\Longrightarrow} x$ by using a rule $B \to z$, while $x_1 B' x_2' \underset{P_i}{\Longrightarrow} x$ by using a rule $B' \to z'$. Since $x = x_1 z x_2 = x_1 z' x_2'$, it follows that z is a prefix of z', or z' is a prefix of z, or $z = z'$. Due to $B \sim_i^{\pi} B'$, none of the first two cases can hold. Therefore, $z = z'$, which implies $B = B'$, and $x_2 = x_2'$.

By the induction hypothesis, it then follows that the derivation

$$A \overset{q-1}{\underset{P_i}{\Longrightarrow}} x_1 B x_2$$

is unambiguous, hence $C(D_1) = C(D_2)$.

Suppose now the statement holds true for each word α with $|\alpha| < p$, and let

$$\begin{aligned} D_1: &\quad \alpha \overset{t}{\underset{P_i}{\Longrightarrow}} x, \\ D_2: &\quad \alpha \overset{t}{\underset{P_i}{\Longrightarrow}} x, \end{aligned}$$

with $|\alpha| = p$.

If $\alpha = A\alpha'$, with $A \in (N \cup T) - dom(P_i)$, then $C(D_1) = C(D_2)$ holds by the induction hypothesis.

If $A \in dom(P_i)$, then there are $y_1, y_2, z_1, z_2 \in (T_i)^*$ such that

$$\begin{aligned} A \overset{t}{\underset{P_i}{\Longrightarrow}} y_1, &\quad \alpha' \overset{t}{\underset{P_i}{\Longrightarrow}} z_1, \\ A \overset{t}{\underset{P_i}{\Longrightarrow}} y_2, &\quad \alpha' \overset{t}{\underset{P_i}{\Longrightarrow}} z_2, \end{aligned}$$

and $x = y_1 z_1 = y_2 z_2$.

Assume that y_1 is a proper prefix of y_2 (the case of y_2 a proper prefix of y_1 can be treated similarly), and let us point out the lengths of the derivations resulting into y_1, y_2, that is, let $A \overset{q_1}{\underset{P_i}{\Longrightarrow}} y_1$, $A \overset{q_2}{\underset{P_i}{\Longrightarrow}} y_2$. We have neither $q_1 < q_2$, nor $q_2 < q_1$. Consequently, $q_1 = q_2$ and $y_1 = y_2$. Moreover, the derivation $A \overset{t}{\underset{P_i}{\Longrightarrow}} y_1$ is unambiguous. It then follows that $z_1 = z_2$. Since $|\alpha_1| < p$, by the induction hypothesis we infer that the derivation $\alpha' \overset{t}{\underset{P_i}{\Longrightarrow}} z_1$ is unambiguous, which concludes the proof. $\square$

For the case of total strict deterministic systems, some global (i.e. at the level of the system) properties of the derivations follow directly from the case of strict deterministic context-free grammars, see [8]. The two propositions that follow present them.

Proposition 1 *Let $\Gamma = (N, T, S, P_1, \ldots, P_n)$ be a reduced CD grammar system (i. e. all the productions and all the nonterminals of Γ are used during terminal derivations) without λ - productions, and let $P = \bigcup_{i=1}^{n} P_i$. If Γ is total strict deterministic under the partition π, then for any $A, B \in N$ such that $A \sim^{\pi} B$, no derivation $B \overset{+}{\underset{P}{\Longrightarrow}} A\alpha$, $\alpha \in (N \cup T)^*$, is possible.*

We make the observation that the requirement for the system Γ to be reduced in the theorem above is not a restriction at all, due to the fact that for any CD grammar system there exists an equivalent (with respect to the generated language) reduced CD grammar system, [10]. Moreover, the proof of [10] only eliminates the non-necessary productions and nonterminals, and therefore does not affect the eventually total strict deterministic partition of the system.

Yet remark that for the case of local strict deterministic systems, a property similar to Proposition 1 holds locally, in each component grammar.

As a consequence, one can eliminate left recursion from a total strict deterministic CD grammar system.

Proposition 2 *For any total strict deterministic CD grammar system $\Gamma = (N, T, S, P_1, \ldots, P_n)$, there exists a total strict deterministic CD grammar system $\Gamma' = (N', T, S, P'_1, \ldots, P'_n)$, with $L(\Gamma') = L(\Gamma)$, such that no derivation $A \overset{+}{\underset{\Gamma'}{\Longrightarrow}} A\alpha$, $\alpha \in (N' \cup T)^*$ is possible, for any $A \in N'$.*

It is worthwhile mentioning that in case of total strict deterministic systems, a property similar to Theorem 1, with respect to a global derivation in the system, does not hold. For instance, consider the grammar system in Example 2, to which we add a new component, P_3, defined as

$$P_3 = \{C \to bC, C \to a\}.$$

One can observe that for any word of the form $a^n b^k a^n$, $n,k \geq 1$, two distinct leftmost derivations can be pointed out: one in the components P_1 and P_2, and the other in the components P_1 and P_3.

Therefore, additional restrictions to the system should be considered in order to cope with such situations. Surprisingly enough, we find out that if the components of the system are mutually disjoint, than the unambiguity of the system follows as well.

Theorem 2 *Any total strict deterministic CD grammar system whose components are mutually disjoint is unambiguous.*

Proof. Let

$$\Gamma = (N, T, S, P_1, \ldots, P_n),$$

be a total strict deterministic CD grammar system such that $P_i \cap P_j = \emptyset$ for any $i, j,\ 1 \leq i, j \leq n,\ i \neq j$.

Denote $P = \bigcup_{i=1}^{n} P_i$, and let $G = (N, T, S, P)$ be the context-free grammar associated to Γ. Clearly $L(\Gamma) \subseteq L(G)$. Due to Γ being total strict deterministic, G is a strict deterministic context-free grammar ([8]). In particular, G is unambiguous follows, i.e. for no word in $L(G)$ two distinct leftmost derivations exist.

We prove that for any word $w \in L(\Gamma)$ such that there exist leftmost derivations

$$D_1 :\ S = x_0 \overset{tl}{\underset{P_{i_1}}{\Longrightarrow}} x_1 \overset{tl}{\underset{P_{i_2}}{\Longrightarrow}} \ldots \overset{tl}{\underset{P_{i_s}}{\Longrightarrow}} x_s = w,$$

$$D_2 :\ S = x'_0 \overset{tl}{\underset{P_{j_1}}{\Longrightarrow}} x'_1 \overset{tl}{\underset{P_{j_2}}{\Longrightarrow}} \ldots \overset{tl}{\underset{P_{j_l}}{\Longrightarrow}} x'_l = w,$$

then $D_1 = D_2$ follows.

We first prove $i_1 = j_1$, by contradiction.

Assume $i_1 \neq j_1$. Let p_1, p'_1 be the productions used at the first derivation steps in P_{i_1}, P_{j_1}, respectively. Since $P_{i_1} \cap P_{j_1} = \emptyset$, then $p_1 \neq p'_1$ results. But this implies that in the grammar G there exists two distinct leftmost derivations of the word w, that is, there exists leftmost derivation

$$S \overset{p_1}{\underset{G}{\Longrightarrow}} \alpha \overset{*}{\underset{G}{\Longrightarrow}} w,$$

$$S \overset{p'_1}{\underset{G}{\Longrightarrow}} \beta \overset{*}{\underset{G}{\Longrightarrow}} w,$$

which contradicts the unambiguity of G.

Hence $i_1 = j_1$ holds. Denote $i = i_1$.

We now prove that any two leftmost derivations

$$\alpha \overset{tl}{\underset{P_i}{\Longrightarrow}} \beta \qquad (1),$$

$$\alpha \underset{P_i}{\overset{tl}{\Longrightarrow}} \gamma \quad (2),$$

which are used in leftmost rewritings of w, actually coincide.

The assumption that the derivations above are used in leftmost rewritings of w implies that there exist leftmost derivations (in Γ)

$$S \underset{\Gamma}{\overset{*}{\Longrightarrow}} \alpha \underset{P_i}{\overset{tl}{\Longrightarrow}} \beta \underset{\Gamma}{\overset{*}{\Longrightarrow}} w,$$

$$S \underset{\Gamma}{\overset{*}{\Longrightarrow}} \alpha \underset{P_i}{\overset{tl}{\Longrightarrow}} \gamma \underset{\Gamma}{\overset{*}{\Longrightarrow}} w.$$

We point out the first productions applied in the rewriting of α, that is, $\alpha = \alpha_1 A \alpha_2$, with $\alpha_1 \in ((N \cup T) - dom(P_i))^*$, $\overline{p}_1$ is a production $\overline{p}_1 : A \rightarrow \delta_1 \in P_i$, $\overline{p}_2$ is a production $\overline{p}_2 : A \rightarrow \delta_2 \in P_i$, and in the derivations above we have

$$\alpha = \alpha_1 A \alpha_2 \underset{P_i}{\overset{\overline{p}_1}{\Longrightarrow}} \alpha_1 \delta_1 \alpha_2 \underset{P_i}{\overset{tl}{\Longrightarrow}} \beta \underset{P_i}{\overset{*}{\Longrightarrow}} w,$$

$$\alpha = \alpha_1 A \alpha_2 \underset{P_i}{\overset{\overline{p}_2}{\Longrightarrow}} \alpha_1 \delta_2 \alpha_2 \underset{P_i}{\overset{tl}{\Longrightarrow}} \gamma \underset{P_i}{\overset{*}{\Longrightarrow}} w.$$

Two situations might occur:
case a): $\alpha_1 \in T^*$.

Then, in a similar manner as in the proof above for $i_1 = j_1$, one can deduce that if $\overline{p}_1 \neq \overline{p}_2$, then a contradiction results.
case b): $\alpha_1 = y_1 B y_2$, for a $B \in N - dom(P_i)$, $y_1 \in T^*, y_2 \in (T \cup (N - dom(P_i))^*$.

Then one can point out leftmost derivations in the associated grammar G of the form

$$S \underset{G}{\overset{*}{\Longrightarrow}} \alpha = \alpha_1 A \alpha_2 = y_1 B y_2 A \alpha_2 \underset{G}{\overset{*}{\Longrightarrow}} y_1 z_1 A \alpha_2 \underset{G}{\overset{\overline{p}_1}{\Longrightarrow}} y_1 z_1 \delta_1 \alpha_2 \underset{G}{\overset{*}{\Longrightarrow}} w,$$

$$S \underset{G}{\overset{*}{\Longrightarrow}} \alpha = \alpha_1 A \alpha_2 = y_1 B y_2 A \alpha_2 \underset{G}{\overset{*}{\Longrightarrow}} y_1 z_2 A \alpha_2 \underset{G}{\overset{\overline{p}_2}{\Longrightarrow}} y_1 z_2 \delta_2 \alpha_2 \underset{G}{\overset{*}{\Longrightarrow}} w,$$

where $B y_2 \underset{G}{\overset{*}{\Longrightarrow}} z_1$ and $B y_2 \underset{G}{\overset{*}{\Longrightarrow}} z_2$ are leftmost terminal derivations in G. Since either z_1 is a prefix of z_2 or z_2 is a prefix of z_1, it follows that $z_1 = z_2$ holds. But this again leads to a contradiction if $\overline{p}_1 \neq \overline{p}_2$ is assumed.

In any of the two cases, $\overline{p}_1 = \overline{p}_2$ must hold. By inductively repeating this argument, we obtain that the derivations (1) and (2) above are actually the same.

But this assertion allows us to repeat the reasoning which lead to concluding $i_1 = j_1$, and we then obtain that in derivations D_1 and D_2 we have $s = l$, $i_k = j_k$ for any k, $1 \leq k \leq s$, and any two leftmost derivations in a component concide, and therefore the theorem follows. $\square$

4 The generative power

The subject matter of this section is the language generated by local/total strict deterministic CD grammar systems. We first prove that a total system is local as well, if some additional constraints are satisfied. Then we show that the problem of whether or not a system is local or total strict deterministic is decidable. Also the hierachy on the number of components is addressed, proving that it collapeses to systems with only 3 components for the local case. Finally, the generative capacity of the systems is compared to that of usual CD grammar systems, or to that of context-free grammars.

Theorem 3 *Let $\Gamma = (N, T, S, P_1, \ldots, P_n)$, be a CD grammar sysetm, strict deterministic under the partition π, satisfying the additional restriction that for any $A, B \in N$, $A \sim^{\pi} B$, such that $A \in dom(P_i)$ for an $i, 1 \leq i \leq n$, then $B \in dom(P_i)$ holds as well. Then Γ is local deterministic.*

Proof. Let Γ be a CD grammar system as in the hypotheses of the theorem and let π be the partition. One can observe that partition π can be written as

$$\pi = \{T\} \cup \{[A]_\pi \mid A \in N\}.$$

The requirement that for any $A, B \in N$, $A \sim^{\pi} B, A \in dom(P_i)$, for an $i, 1 \leq i \leq n$, implies $B \in dom(P_i)$ as well, means that for any $A \in N$, and for a ny $i, 1 \leq i \leq n$, either $[A]_\pi \subseteq dom(P_i)$, or $[A]_\pi \subseteq N - dom(P_i)$. Then for any $i, 1 \leq i \leq n$, one can consider the partition

$$\pi_i = \{T\} \cup \{N - dom(P_i)\} \cup \{[A]_\pi \mid A \in dom(P_i).\}$$

Now Γ is local deterministic under the partitions π_i, $1 \leq i \leq n$, follows. □

Theorem 4 *Let $\Gamma = (N, T, S, P_1, P_2, \ldots, P_n)$ be a CD grammar system and let $k = | \cup_{i=1}^{n} P_i|$.*

1. It is decidable in $O(k^2 \cdot |N|)$ whether or not Γ is total strict deterministic.

2. It is decidable in $O(\sum_{i=1}^{n} |dom(P_i)| \cdot |P_i|^2)$ whether or not Γ is local strict deterministic.

Proof. The assertions above can be proved by constructing algorithms which rely on the same idea. We shall briefly explain it in the hypothesis of the first point. See [8] for more details.

For two partitions of $N \cup T$ we say that π_1 is smaller or equal than π_2, if $[X]_{\pi_1} \subseteq [X]_{\pi_2}$, for any $X \in N \cup T$. The notation $[X]_\pi$ stands for the equivalence class of X with respect to the partition π.

One starts with the smallest possible partition $\pi = \{T\} \cup \{\{A\} | A \in N\}$. By examinating the rules of $\bigcup_{i=1}^{n} P_i$ whose left-hand sides are in the same equivalence classes, with respect to the current partition, the following situations may occur:

1. A nonterminal and a terminal are to be put in the same equivalence class. But this is not possible, due to the definition of the total strict deterministic partition, therefore the grammar system is not total strict deterministic follows. This is a consequence of choosing the sequence of current partitions in increasing order.

2. Two nonterminals, say A and B, should be considered in the same equivalence class, and they are not equivalent with respect to the current partition. Then, the current partition becomes

$$\pi \ = \ (\pi \setminus \{[A]_\pi, [B]_\pi\}) \cup \{[A]_\pi \cup [B]_\pi\}$$

   Now, the process goes on by examinating again the productions of Γ.

3. No change of the current partition had to be performed during the last examination of the productions of Γ. Then it follows that the grammar system is total strict deterministic, and the current partition is a total strict partition.

The correctness and the complexity of the algorithm are easy to be checked. □

We now turn our attention to the generative capacity of total/local strict deterministic CD grammar systems.

Theorem 5 *For any local strict deterministic CD grammar system of degree n, there exists an equivalent (with respect to the generated language) local strict deterministic CD grammar system of degree* 3.

Proof. The proof parallels the reduction of the number of components to 3 in the case of usual CD grammar systems (deriving in the terminal mode), see [1]. Let $\Gamma = (N, T, P_1, P_2, \ldots, P_n, S)$ be a system of degree n, which is local strict deterministic under the partitions π_i, $1 \leq i \leq n$. Consider

$$N' = \bigcup_{i=0}^{n} \{A^{(i)} \mid A \in N\},$$

and for any i, $0 \leq i \leq n$, define the morphisms $h_i : N \cup T \longrightarrow N' \cup T$ by

$$h_i(X) = \begin{cases} X, & \text{for } X \in T \\ X^{(i)}, & \text{for } X \in N. \end{cases}$$

For the sake of simplicity, for a set M, we denote $h_i(M) = \{h_i(A) \mid A \in M\}$. Consider $\Gamma' = (N', T, P_1', P_2', P_3', S^{(0)})$, where

$$
\begin{aligned}
P_1' &= \bigcup_{i=1}^{n} \{A^{(i)} \to h_i(\alpha) \mid A \to \alpha \in P_i\},\\
P_2' &= \begin{cases} \{A^{(i)} \to A^{(i+1)} \mid A \in N, 0 \le i < n,\ i \text{ even number}\}, \text{ if } n \text{ is odd},\\ \{A^{(i)} \to A^{(i+1)} \mid A \in N, 0 \le i < n,\ i \text{ even number}\} \cup \\ \cup \{A^{(n)} \to A^{(1)}\}, \text{ if } n \text{ is even} \end{cases}\\
P_3' &= \begin{cases} \{A^{(i)} \to A^{(i+1)} \mid A \in N, 0 \le i < n,\ i \text{ odd number}\}, \text{ if } n \text{ is even},\\ \{A^{(i)} \to A^{(i+1)} \mid A \in N, 0 \le i < n,\ i \text{ odd number}\} \cup \\ \cup \{A^{(n)} \to A^{(0)}\}, \text{ if } n \text{ is odd} \end{cases}
\end{aligned}
$$

Then Γ' is local strict deterministic under the partitions

$$
\begin{aligned}
\pi_1' &= \{T\} \cup \bigcup_{i=1}^{n} \{h_i([A]_{\pi_i}) \mid A \in dom(P_i)\} \cup \{N' - dom(P_1'),\\
\pi_2' = \pi_3' &= \{T\} \cup \{\ \{A^{(i)}\} \mid A \in N, 0 \le i \le n,\ i \text{ even number}\}\\
&\cup\ \{\ \{A^{(i)}\} \mid A \in N, 0 \le i \le n,\ i \text{ odd number}\}.
\end{aligned}
$$

On the other hand, $L(\Gamma') = L(\Gamma)$ can be proved in a similar manner as in [1]. □

Theorem 6 *For any CD grammar system there exists an equivalent local strict deterministic CD grammar system.*

Proof. Let $\Gamma = (N, T, S, P_1, P_2, \ldots, P_n)$ be a CD grammar system in which all rules are labelled by distinct labels. Without loss of generality, we may assume that no P_i contains erasing rules. If the rules of P_i are labelled by $r_{i,1}, r_{i,2}, \ldots, r_{i,k_i}$, then consider the set of new nonterminals

$$N' = (\bigcup_{i=1}^{n} \{[i, j, X] | 1 \le j \le k_i, X \in N \cup T\}) \cup \{X_i | 1 \le i \le n, X \in N\} \cup \{F\}$$

We construct the grammar system

$$\Gamma' = (N \cup N', T, S, P_1', P_2', \ldots, P_n', P_{n+1}),$$

where

$$
\begin{aligned}
P_i' &= \{A \longrightarrow [i, j, X]y \mid r_{i,j} : A \longrightarrow Xy \in P_i\} \cup \{A_i \longrightarrow A | A \in N\}\\
&\cup\ \{[t, s, X] \longrightarrow F \mid 1 \le t \le n, t \ne i, 1 \le s \le k_t, X \in N \cup T\}\\
&\cup\ \{A_j \longrightarrow F \mid A \in N, 1 \le j \le n, j \ne i\},\ 1 \le i \le n,\\
P_{n+1} &= \{[i, j, a] \longrightarrow a \mid a \in T, 1 \le i \le n, 1 \le j \le k_i\}\\
&\cup\ \{[i, j, A] \longrightarrow A_i \mid A \in N, 1 \le i \le n, 1 \le j \le k_i\}
\end{aligned}
$$

For any $1 \le i \le n$, denote by Q_i the set of all A_i such that A is a nonterminal in which appears on the first position of a right-hand side of a rule in P_i. Consider the partitions

$$\begin{aligned}
\pi_i &= \{dom(P_i) \cup Q_i \cup \{A_j \mid j \neq i, A \in N\}\} \\
&\cup \{N' \setminus (dom(P_i) \cup Q_i \cup \{A_j \mid j \neq i, A \in N\}), T\}, 1 \le i \le n, \\
\pi_{n+1} &= \{T, N \cup \{A_i \mid A \in N, 1 \le i \le n\}\} \\
&\cup \{\{[i,j,X]\} \mid 1 \le i \le n, 1 \le j \le k_i, X \in N \cup T\}.
\end{aligned}$$

Obviously, the above partitions are local strict partitions. Since any derivation in a component P_i is simulated by an arbitrary number of derivations in the pair of components P_i', P_{n+1}, it follows that $L(\Gamma) = L(\Gamma')$. □

As far as the generative power of total strict deterministic CD grammar systems is concerned, one can observe that this decreases as shown by the next theorem.

Theorem 7 *Every language generated by a total strict deterministic CD grammar system is prefix-free.*

Proof. It is known that every language accepted by an $LR(0)$ grammar is prefix-free. If $\Gamma = (N, T, S, P_1, P_2, \ldots, P_n)$ is a total strict deterministic CD grammar system, then $L(\Gamma) \subseteq L(G)$, where $G = (N, T, S, \cup_{i=1}^{n} P_i)$.

But $L(G)$ generates a language that can be generated by an $LR(0)$ grammar, hence prefix-free and the proof is complete. □

On the other hand, there are non-context-free languages that can be generated by total strict deterministic CD grammar systems. An example is provided by the CD grammar system with the components given by

$$\begin{aligned}
P_1 &= \{S \to AC, X \to A, Y \to C\} \\
P_2 &= \{A \to aXb, C \to cY\} \\
P_3 &= \{A \to d, C \to d\}
\end{aligned}$$

and the total strict partition

$$\pi = \{\{S\}, \{A\}, \{C\}, \{X\}, \{Y\}, \{a, b, c, d\}\}$$

The language generated by the above grammar system is $\{a^n d b^n d c^n d | n \ge 1\}$. Therefore, we have proved:

Theorem 8 *The family of context-free languages and the family of languages generated by total strict deterministic CD grammar systems are incomparable.*

5 Final Remarks

It is worthwhile mentioning that the restrictions we have imposed on CD grammar systems have meaningful interpretations in terms of the *blackboard architecture* of problem solving, which CD grammar systems are a formal representation for, [5]. Briefly stated, the blackboard model consists of three basic parts ([12]):

- the *knowledge* needed to solve a given problem, which is partitioned into separate and independent *knowledge sources*;
- the *global data base*, the blackboard, representing the current state of the problem solving process, and in which the knowledge sources can make changes;
- the *control* of the opportunistic responds of the knowledge sources to make changes in the blackboard.

In the formal modelization, the knowledge sources correspond to grammars and the blackboard to a sentential form. Rewriting a nonterminal can be interpreted as a developmental step of the information contained in the current state of the blackboard, a solution of the problem corresponds to a terminal word, and the control mechanism states the conditions under which a grammar can start or stop its work.

Therefore, it is natural to try to encapture in the CD grammar systems' architecture other features of their Artificial Intelligence counterpart. To this aim, if one regards nonterminal symbols as corresponding to subproblems to be solved, then it is of interest to study in details CD grammar systems in which a relation partioning the subproblems, according to their state of the art with respect to the process of finding solutions, holds. But this is exactly what the strict determinism features considered here for CD grammar systems formally model. More precisely, the partitions of the symbol set of a system can be regarded as follows:

i) all the terminal symbols (at the level of the system, and also at the level of each component, when a local relation is considered) are related, since they all are (partial) solutions to some problems, the presence of a terminal symbol on the blackboard does not require any subsequent study of this symbol;

ii) nonterminal symbols (corresponding to subproblems to be solved) are considered related if, once one has started to build a solution of a subproblem, this is a beginning of a solution of a related subproblem as well.

References

[1] A. Atanasiu, V. Mitrana, The Modular Grammars, *Internat. J. Comp. Math.* 30 (1989), 17–35

[2] H. Fernau, M. Holzer, H. Bordihn, Accepting Multi-Agent Systems: The Case of Cooperating Grammar Systems, *Computers and Artificial Intelligence* 15 (1996), No. 2-3, 123-139

[3] E. Csuhaj-Varju, J. Dassow, On Cooperating Distributed Grammar Systems, *J. Inform. Process. Cybern., EIK*, 26 (1990), 49 – 63.

[4] E. Csuhaj-Varju, J. Dassow, J. Kelemen, Gh. Păun, *Grammar Systems*, Gordon and Breach, 1993.

[5] E. Csuhaj-Varju, J. Kelemen, Cooperating Grammar Systems: A Syntactical Framework for the Blackboard Model of Problem Solving, *Proc. Artificial Intelligence and Information-Control Systems of Robots '89*, North-Holland Publ. Co. (I. Plander, ed.), 1989, 121 – 127.

[6] J. Dassow, V. Mitrana, Cooperating Pushdown Automata Systems, *Submitted*, 1995.

[7] J. Dassow, Gh. Păun, G. Rozenberg, Grammar Systems, in *The Handbook of Formal Languages* (G. Rozenberg, A. Salomaa, eds.), Springer-Verlag, to appear

[8] M. Harrison, *Introduction to Formal Language Theory*, Addison-Wesley Publ. Co., 1978

[9] J. E. Hopcroft, J. D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading Mass., 1979.

[10] V. Mihalache, Coverability Trees for Grammar Systems: A Strong Decidability Tool, *submitted*

[11] R. Meersman, G. Rozenberg, Cooperating Grammar Systems, *Proc. MFCS '78 Symp., LNCS 64*, Springer-Verlag, 1978, 364 – 374.

[12] P. H. Nii, Blackboard Systems, in *The Handbook of AI*, vol. 4 (A. Barr, P. R. Cohen, E. A. Feigenbaum, eds.), Addison-Wesley, 1989.

Turku Centre for Computer Science
Lemminkäisenkatu 14
FIN-20520 Turku
Finland

http://www.tucs.abo.fi

University of Turku
- **Department of Mathematical Sciences**

Åbo Akademi University
- **Department of Computer Science**
- **Institute for Advanced Management Systems Research**

Turku School of Economics and Business Administration
- **Institute of Information Systems Science**