Journal of Universal Computer Science, vol. 1, no. 2 (1995), 105-130
submitted: 22/1/95, accepted: 18/2/95, appeared: 28/2/95, ©Springer Pub. Co.

A Variant of Team Cooperation in Grammar Systems

Rudolf Freund
Technical University Wien, Institute for Computer Languages
Resselgasse 3, 1040 Wien, Austria
freund@csdec1.tuwien.ac.at

Gheorghe Păun[1]
Institute of Mathematics of the Romanian Academy of Sciences
PO Box 1-764, 70700 Bucureşti, Romania
gpaun@imar.ro

Abstract: We prove that grammar systems with (prescribed or free) teams (of constant size at least two or arbitrary size) working as long as they can do, characterize the family of languages generated by (context-free) matrix grammars with appearance checking; in this way, the results in [Păun, Rozenberg 1994] are completed and improved.

Keywords: Formal languages, grammar systems, teams
Category: F.4.2, F.4.3

1 Introduction

A cooperating grammar system, as introduced in [Csuhaj-Varjù, Dassow 1990] and [Meersman, Rozenberg 1978], consists of several (usually context-free) grammars, each of them working, by turns, on a common sentential form. A basic protocol of cooperation is the *maximal competence* strategy: a component must rewrite the current sentential form as long as it can do this (and hence never can finish, if it can work forever). In [Csuhaj-Varjù, Dassow 1990] it is proved that in this way exactly the family of $ET0L$-languages can be obtained. In [Meersman, Rozenberg 1978] a variant of this *stop condition* is considered: a component must work until it introduces a non-terminal which cannot be rewritten by the same component.

In [Kari, Mateescu, Păun, Salomaa 1994], a way to increase the power of cooperating grammar systems has been proposed: the cooperation of the components of a grammar system is increased by allowing (or forcing) some of the components of the system to work simultaneously in *teams* on the current sentential form in parallel, i. e. in each step, every member of the currently active team has to apply

[1] Research carried out during the author's visit at the Technical University Wien

a rule. In [Kari, Mateescu, Păun, Salomaa 1994], the condition for a team to stop its work has been the following one: no rule of any member of the team can be used any more. Even with such a strong stop condition, non-$ET0L$-languages can be generated as it is proved in [Kari, Mateescu, Păun, Salomaa 1994] (and moreover, teams of size two are sufficient, as it is shown in [Csuhaj-Varjù, Păun 1993]).

Another stop condition has been considered in [Păun, Rozenberg 1994]: a team stops working if and only if at least one of its members cannot apply one of its rules any more. For this stop condition as well as for that introduced in [Kari, Mateescu, Păun, Salomaa 1994], in [Păun, Rozenberg 1994] it is proved that both using prescribed teams (all of them being of given size or of free size) and using free teams (of given size at least two or of arbitrary size at least two) exactly the family of languages generated by matrix (or programmed) grammars with appearance checking is obtained (thus strenghtening the results proved in [Csuhaj-Varjù, Păun 1993] and [Kari, Mateescu, Păun, Salomaa 1994]).

The stop conditions considered in [Păun, Rozenberg 1994] are not the natural extension of the *maximal competence* strategy from individual components of grammar systems to teams of components: the simplest way for such an extension is to allow a team to become inactive when it is no longer able to rewrite the current sentential form *as a team,* irrespective whether or not some or even all rules of the components can be applied further. For instance, if the current string contains only two occurences of the non-terminal A and we have a team consisting of three components, each consisting of rules of the form $A \rightarrow \alpha$ only, then none of the conditions investigated in [Csuhaj-Varjù, Păun 1993], [Kari, Mateescu, Păun, Salomaa 1994], and [Păun, Rozenberg 1994] is fulfilled, although the team cannot be used any more. Yet the derivation is correctly terminated if we use the natural extension of the maximal competence strategy mentioned above, but not for the variants considered in [Csuhaj-Varjù, Păun 1993], [Kari, Mateescu, Păun, Salomaa 1994], and [Păun, Rozenberg 1994] (the derivation is simply unacceptable for those variants, although it looks quite rationally considered from the point of view of the team).

There is also another reason for considering the new stop condition, namely a mathematical one: grouping sets of rules in teams may remind us of the mode of working of matrix grammars; checking whether rules in a component of a team can be applied may remind us of the appearance checking in matrix grammars. All together, these aspects make the following result somehow non-surprising (although the proof given in [Păun, Rozenberg 1994] is, by no means, obvious): grammar systems with teams (prescribed or free and of given size at least two or of free size at least two) working with the stop conditions considered in [Păun, Rozenberg 1994] characterize the family of languages generated by (context-free) matrix grammars with appearance checking. The new mode of stopping the work of a team is not related to the appearance checking manner of work in such an obvious manner, yet again all languages generated by matrix grammars with appearance checking can be obtained by grammar systems with free teams of given size at least two, but also with free teams of arbitrary size, which is an improvement of the results obtained in [Păun, Rozenberg 1994].

The study of teams, in general the study of classes of grammar systems in which both the sequential and the parallel modes of working are present, requests and deserves further efforts (see also [Csuhaj-Varjù 1994] for motivations of such investigations).

2 Preliminary definitions

We specify only a few notions and notations here; the reader is referred to [Salomaa 1973] for other elements of formal language theory we shall use and to [Dassow, Păun 1989] for the area of regulated rewriting.

For an alphabet V, by V^* we denote the free monoid generated by V under the operation of concatenation; the empty string is denoted by λ, and $V^* - \{\lambda\}$ is denoted by V^+. The length of $x \in V^*$ is denoted by $|x|$, and for any U, $U \subseteq V$, $|x|_U$ denotes the number of occurrences of symbols $a \in U$ in x.

A *matrix grammar (with appearance checking)* is a construct

$$G = (N, T, S, M, F)$$

where N and T are disjoint alphabets (N is the nonterminal alphabet, T is the terminal alphabet), $S \in N$ is the axiom, and M is a finite set of sequences (called *matrices*) of the form $m = (A_1 \to x_1, \ldots, A_s \to x_s)$, $s \geq 1$, $A_i \to x_i$ being a context-free rule over $N \cup T$ with $A_i \in N$ and $x_i \in (N \cup T)^*$, $1 \leq i \leq s$, and F is a subset of the rules occurring in the matrices of M.

For $w, y \in (N \cup T)^*$ we write $w \Longrightarrow y$ if there are strings $w_0, w_1, \ldots, w_s$ in $(N \cup T)^*$ and a matrix $(A_1 \to x_1, \ldots, A_s \to x_s)$ in M such that $w = w_0$, $w_s = y$ and for each i with $1 \leq i \leq s$ either $w_{i-1} = z_i A_i z_i'$ and $w_i = z_i x_i z_i'$ or $w_i = w_{i-1}$, the rule $A_i \to x_i$ is not applicable to w_{i-1}, and $A_i \to x_i$ appears in F. (In words, all the rules in a matrix are applied, one after the other in the given sequence, possibly skipping the rules appearing in F, but only if they cannot rewrite the current string.) If $F = \emptyset$, then the grammar is said to be *without appearance checking* (and the component F can be omitted).

By MAT_{ac}^{λ}, MAT_{ac} we denote the families of languages generated by matrix grammars with arbitrary context-free respectively λ-free context-free rules. The following relations are known ([Dassow, Păun 1989]):

$$ET0L \subset MAT_{ac} \subset CS \subset MAT_{ac}^{\lambda} = RE,$$

where CS and RE denote the families of context-sensitive respectively recursively enumerable languages and $ET0L$ denotes the family of λ-free $ET0L$-languages (i.e. languages generated by extended Lindenmayer systems with tables).

A *cooperating distributed grammar system* (*CD grammar system* for short) is a construct

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

where N and T are disjoint alphabets (N is the nonterminal alphabet, T is the terminal alphabet), $S \in N$ is the axiom, and $P_1, ..., P_n$, $n \geq 1$, are finite sets of context-free rules over $N \cup T$ and are called the *components* of the system Γ.

For each component P_i, $1 \leq i \leq n$, in the CD grammar system Γ we denote

$$dom(P_i) = \left\{ A \in N \mid A \to x \in P_i \text{ for some } x \in (N \cup T)^* \right\}.$$

Given $w, w' \in (N \cup T)^*$ and i, $1 \leq i \leq n$, we write $w \Longrightarrow_{P_i} w'$ if w' can be derived from w by using a rule in P_i in the usual sense: $w = w_1 A w_2$, $w' = w_1 x w_2$, and $A \to x \in P_i$. By $\Longrightarrow_{P_i}^{+}$ and $\Longrightarrow_{P_i}^{*}$ we denote the transitive respectively the reflexive transitive closure of $\Longrightarrow_{P_i}$.

An important derivation relation for CD grammar systems is the *maximal derivation mode t* (see [Csuhaj-Varjù, Dassow 1990]):

$$w \Longrightarrow_{P_i}^{t} w' \text{ if and only if}$$
$$w \Longrightarrow_{P_i}^{*} w' \text{ and there is no } w'' \in (N \cup T)^* \text{ such that } w' \Longrightarrow_{P_i} w''$$

(such a derivation is *maximal* in the component P_i, i.e. no further step can be done). The language generated by the CD grammar system Γ in the maximal derivation mode t is defined by

$$L_t(\Gamma) = \{x \in T^* \mid S \Longrightarrow_{P_{i_1}}^{t} w_1 ... \Longrightarrow_{P_{i_m}}^{t} w_m = x, \\ m \geq 1,\ 1 \leq i_j \leq n \text{ for } 1 \leq j \leq m\}.$$

The family of languages generated in this mode by CD grammar systems with λ-free rules is denoted by $CD(t)$. From [Csuhaj-Varjù, Dassow 1990] we know that $CD(t) = ET0L$.

3 Teams in cooperating grammar systems

In [Kari, Mateescu, Păun, Salomaa 1994] the following extension of CD grammar systems is introduced:
A *CD grammar system with (prescribed) teams (of variable size)* is a construct

$$\Gamma = (N, T, S, P_1, ..., P_n, Q_1, ..., Q_m)$$

where $(N, T, S, P_1, ..., P_n)$ is a usual CD grammar system and $Q_i \subseteq \{P_1, ..., P_n\}$, $1 \leq i \leq m$; the sets $Q_1, ..., Q_m$ are called *teams* and are used in derivations as follows: For $Q_i = \{P_{j_1}, P_{j_2}, ..., P_{j_s}\}$ and $w,\ w' \in (N \cup T)^*$ we write

$$\begin{aligned} w \Longrightarrow_{Q_i} w' \ \text{ if and only if } \ & w = w_1 A_1 w_2 A_2 ... w_s A_s w_{s+1}, \\ & w' = w_1 x_1 w_2 x_2 ... w_s x_s w_{s+1}, \\ & \text{where } w_k \in (N \cup T)^*,\ 1 \leq k \leq s+1, \text{ and} \\ & A_r \rightarrow x_r \in P_{j_r},\ 1 \leq r \leq s \end{aligned}$$

(the team is a set, hence no ordering of the components is assumed).

In [Kari, Mateescu, Păun, Salomaa 1994] the following rule of finishing the work of a team $Q_i = \{P_{j_1}, P_{j_2}, ..., P_{j_s}\}$ has been considered:

$$w \Longrightarrow_{Q_i}^{t_1} w' \text{ if and only if}$$
$$w \Longrightarrow_{Q_i}^{+} w' \text{ and } |w'|_{dom(P_{j_r})} = 0 \text{ for all } r \text{ with } 1 \leq r \leq s.$$

(No rule of any component of the team can be applied to w'.)

Another variant is proposed in [Păun, Rozenberg 1994]:

$$w \Longrightarrow_{Q_i}^{t_2} w' \text{ if and only if}$$
$$w \Longrightarrow_{Q_i}^{+} w' \text{ and } |w'|_{dom(P_{j_r})} = 0 \text{ for some } r \text{ with } 1 \leq r \leq s.$$

(There is a component of the team that cannot rewrite any symbol of the current string.)
The language generated by Γ in one of these modes is denoted by $L_{t_1}(\Gamma)$ and $L_{t_2}(\Gamma)$, respectively.
If all teams in Γ have the same size, then we say that Γ is a CD grammar system with *teams of constant size*. If all possible teams are considered, we say that Γ has *free teams*; the teams then need not be specified. If we allow free teams of only one size, we speak of CD systems with *free teams of constant size*. Obviously, if we only have teams of size $s \geq 2$, then we cannot rewrite an axiom consisting of one symbol only, hence we must start from a string or a set of strings as axioms. Therefore, we consider systems of the form

$$\Gamma = (N, T, W, P_1, .., P_n, Q_1, ..., Q_m),$$

where $W \subseteq (N \cup T)^*$ is a finite set; the terminal strings of W are directly added to the language generated by Γ. The others are used as starting points for derivations. The languages generated by such a system Γ when using free teams of given size s are denoted by $L_{t_1}(\Gamma, s)$ and $L_{t_2}(\Gamma, s)$, respectively; when free teams of arbitrary size are allowed, we write $L_{t_1}(\Gamma, *)$ respectively $L_{t_2}(\Gamma, *)$, and if these free teams must be of size at least two, we write $L_{t_1}(\Gamma, +)$ respectively $L_{t_2}(\Gamma, +)$.
By $PT_sCD(g)$ we denote the family of languages generated in the mode $g \in \{t_1, t_2\}$ by CD grammar systems with prescribed teams of constant size s and λ-free context-free rules; if the size is not constant we replace s by $*$; when the size must be at least 2 (no team consisting of only one component is allowed), then we write $PT_+CD(g)$. If the teams are not prescribed, we remove the letter P, thus obtaining the families $T_sCD(g)$, $T_*CD(g)$, and $T_+CD(g)$, respectively.
As we are interested in the relations with the family MAT_{ac}^{λ}, too, we also consider CD grammar systems with prescribed (arbitrary) teams of constant size s (arbitrary size, of size at least two) and arbitrary context-free rules; the corresponding families of languages generated in the mode $g \in \{t_1, t_2\}$ by such CD grammar systems are denoted by $PT_sCD^{\lambda}(g)$, $PT_*CD^{\lambda}(g)$, $PT_+CD^{\lambda}(g)$, and $T_sCD^{\lambda}(g)$, $T_*CD^{\lambda}(g)$, $T_+CD^{\lambda}(g)$, respectively.

In [Păun, Rozenberg 1994] it is proved that for all $s \geq 2$ and $g \in \{t_1, t_2\}$
$T_sCD(g) = PT_sCD(g) = PT_*CD(g) = T_+CD(g) = MAT_{ac}$ and
$T_sCD^{\lambda}(g) = PT_sCD^{\lambda}(g) = PT_*CD^{\lambda}(g) = T_+CD(g)^{\lambda} = MAT_{ac}^{\lambda}$.

The relations $\Longrightarrow^{t_1}$ and $\Longrightarrow^{t_2}$ as defined in [Csuhaj-Varjù, Păun 1993], [Kari, Mateescu, Păun, Salomaa 1994], and [Păun, Rozenberg 1994] are not the direct extensions of the relation $\Longrightarrow^{t}$ from components to teams. Such an extension looks as follows (where Γ, w, w', Q_i are as above):

$$w \Longrightarrow_{Q_i}^{t_0} w' \text{ if and only if}$$
$$w \Longrightarrow_{Q_i}^{+} w' \text{ and there is no } w'' \in (N \cup T)^* \text{ such that } w' \Longrightarrow_{Q_i} w''.$$

The language generated by the CD grammar system Γ in this mode t_0 is denoted by $L_{t_0}(\Gamma)$, the languages generated by such a system Γ in the mode t_0 when using

free teams of given size s, free teams of arbitrary size, free teams of size at least two are denoted by $L_{t_0}(\Gamma, s)$, $L_{t_0}(\Gamma, *)$. and $L_{t_0}(\Gamma, +)$, respectively.
Obviously, if $w \Longrightarrow_{Q_i}^{t_j} w'$, $j = 1, 2$, then $w \Longrightarrow_{Q_i}^{t_0} w'$, too, but, as we have pointed out in the introduction, the converse is not true; we can have $w \Longrightarrow_{Q_i}^{t_0} w'$ without having $w \Longrightarrow_{Q_i}^{t_j} w'$ for $j = 1, 2$. Consequently, $L_{t_j}(\Gamma) \subseteq L_{t_0}(\Gamma)$, $j = 1, 2$, without necessarily having an equality; the same holds true for the languages $L_{t_j}(\Gamma, s)$, $L_{t_j}(\Gamma, *)$, and $L_{t_j}(\Gamma, +)$. This means that we have no relations directly following from definitions, between families considered above and the corresponding families $PT_sCD(t_0)$, $PT_*CD(t_0)$, $PT_+CD(t_0)$, $T_sCD(t_0)$, $T_*CD(t_0)$, and $T_+CD(t_0)$. However, in the following section we shall prove that again a characterization of the families MAT_{ac} and MAT_{ac}^{λ} is obtained, hence the new termination mode of team work is equally powerful as those considered in [Csuhaj-Varjù, Păun 1993], [Kari, Mateescu, Păun, Salomaa 1994], and [Păun, Rozenberg 1994].

In order to elucidate some of the specific features of the derivation modes t_k, $k \in \{0, 1, 2\}$, we consider some examples. The first example shows that the inclusions, $L_{t_j}(\Gamma) \subseteq L_{t_0}(\Gamma)$, etc., $j \in \{1, 2\}$, can be proper:

Example 1. Let

$$\Gamma_1 = (\{A, B, C\}, \{a\}, \{AB\}, P_1, P_2, P_3, P_4)$$

be a CD grammar system with the sets of rules

$$\begin{aligned} P_1 &= \{A \to B, B \to B\}, \\ P_2 &= \{B \to C, B \to B\}, \\ P_3 &= \{B \to a, B \to B\}, \text{ and} \\ P_4 &= \{C \to a, B \to B\}. \end{aligned}$$

Obviously, $L_t(\Gamma_1) = \emptyset$, because the only way to get rid of the symbol A is to apply the rule $A \to B$ from P_1, but because of the rule $B \to B$ the derivation can never terminate.
If we consider Γ_1 together with the prescribed teams (of size 2)

$$\begin{aligned} Q_1 &= \{P_1, P_2\} \text{ and} \\ Q_2 &= \{P_3, P_4\}, \end{aligned}$$

i.e. if we take the CD grammar system with prescribed teams

$$\Gamma_2 = (\Gamma_1, Q_1, Q_2),$$

then we obtain

$$L_{t_0}(\Gamma_2) = \{aa\}$$

because $AB \Longrightarrow_{Q_1}^{t_0} BC \Longrightarrow_{Q_2}^{t_0} aa$, yet still

$$L_{t_i}(\Gamma_2) = \emptyset$$

for $i \in \{1, 2\}$, because after one derivation step with Q_1, i.e. $AB \Longrightarrow_{Q_1} BC$, Q_1 cannot be applied as a team any more to BC, although the rule $B \to B$, which is in both sets of rules of the team Q_1, is applicable to BC. This means that

the derivation is blocked, although the stop condition for the derivation mode t_i, $i \in \{1,2\}$, is not fulfilled!
As only teams of size at most two can be applied to a string of length two, we also obtain

$$L_{t_j}(\Gamma_1, 2) = L_{t_j}(\Gamma_1, +) = L_{t_j}(\Gamma_1, *) = \emptyset \text{ for } i \in \{1,2\},$$

whereas

$$L_{t_0}(\Gamma_1, 2) = L_{t_0}(\Gamma_1, +) = L_{t_0}(\Gamma_1, *) = \{aa\}.$$

Example 2. Let

$$\Gamma_3 = (\{A, B\}, \{a, b\}, \{AA, BB\}, P_1, P_2, P_3, P_4)$$

be a CD grammar system with the sets of rules

$$\begin{aligned} P_1 &= \{A \to aA, A \to aB, A \to b\}, \\ P_2 &= \{A \to aA, A \to aB, A \to a\}, \\ P_3 &= \{B \to bB, B \to bA, B \to a\}, \text{ and} \\ P_4 &= \{B \to bB, B \to bA, B \to b\}. \end{aligned}$$

If we consider Γ_3 together with the prescribed teams (of size 2)

$$\begin{aligned} Q_1 &= \{P_1, P_2\} \text{ and} \\ Q_2 &= \{P_3, P_4\}, \end{aligned}$$

i.e. if we take the CD grammar system with prescribed teams

$$\Gamma_4 = (\Gamma_3, Q_1, Q_2),$$

then we obtain

$$L_{t_i}(\Gamma_4) = \{wawb, wbwa \mid w \in \{a,b\}^*\}$$

for $i \in \{0,1,2\}$. Although this non-context-free language is obtained in each derivation mode t_i, the intermediate sentential forms (after an application of Q_1 or Q_2) are not the same:
Whereas for $i \in \{1,2\}$ the intermediate sentential forms are $wAwA$ and $wBwB$ with $w \in \{a,b\}^*$, in the derivation mode t_0 we also obtain $wawA$, $wawB$, $wbwA$, $wbwB$, $wAwa$, $wAwb$, $wAwB$, $wBwa$, $wBwb$, and $wBwA$. These strings are somehow *hidden* in the other derivation modes t_1 and t_2, because they can be derived from a sentential form $vAvA$ or $vBvB$ with a suitable $v \in \{a,b\}^*$, by using the derivation relation $\Longrightarrow_{Q_1}$ of the team Q_1, but then further derivations with the team Q_1 are blocked, although the stop conditions of the derivation modes t_1 respectively t_2 are not fulfilled. This additional control on the possible sentential forms is not present with the derivation mode t_0, where a derivation using a team stops *if and only if* the team cannot be applied *as a team* any more, which does not say anything about the applicability of the rules in the components of the team on the current sentential form. Nethertheless the same generative power as with the derivation modes t_1 and t_2 can be obtained by teams using the derivation mode t_0, too, which will be shown in the succeeding section.

4 The power of the derivation mode $\mathbf{t_0}$

In this section we shall prove that CD grammar systems with (prescribed or free) teams (of given size at least two respectively of arbitrary size) together with the derivation mode t_0 again yield characterizations of the families MAT_{ac} respectively MAT_{ac}^{λ}.

The following relations are obvious:

Lemma 1. For all $s \geq 1$ we have

$$T_sCD(t_0) \subseteq PT_sCD(t_0) \subseteq PT_*CD(t_0),$$

$$T_sCD^{\lambda}(t_0) \subseteq PT_sCD^{\lambda}(t_0) \subseteq PT_*CD^{\lambda}(t_0),$$

$$T_*CD(t_0) \subseteq PT_*CD(t_0),$$

$$T_*CD^{\lambda}(t_0) \subseteq PT_*CD^{\lambda}(t_0),$$

$$T_+CD(t_0) \subseteq PT_+CD(t_0) \subseteq PT_*CD(t_0),$$

$$T_+CD^{\lambda}(t_0) \subseteq PT_+CD^{\lambda}(t_0) \subseteq PT_*CD^{\lambda}(t_0).$$

Lemma 2. $PT_*CD(t_0) \subseteq MAT_{ac}$ and $PT_*CD^{\lambda}(t_0) \subseteq MAT_{ac}^{\lambda}$.

Proof. Let $\Gamma = (N, T, W, P_1, ..., P_n, Q_1, ..., Q_m)$ be a CD grammar system with prescribed teams and λ-free rules. We construct a matrix grammar

$$G = (N', T \cup \{c\}, S', M, F)$$

with λ-free rules as follows.
For a team

$$Q_i = \{P_{j_1}, ..., P_{j_s}\}$$

consider all sequences of rules of the form

$$\pi = (A_1 \to x_1, ..., A_s \to x_s)$$

such that from each set P_{j_r} exactly one rule is present in m. Let

$$\{\pi_{i,1}, \pi_{i,2}, ..., \pi_{i,k_i}\} = R_i$$

be all such sequences associated with the team Q_i.
Then

$$N' = N \cup \{A' \mid A \in N\} \cup \{S', \#, X, X'\} \cup \{[Q_i] \mid 1 \leq i \leq m\} \cup \left\{R_{i,j}, R'_{i,j} \mid 1 \leq i \leq m, 1 \leq j \leq k_i\right\}$$

and

$$\begin{aligned} M = &\{(S' \to wX) \mid w \in W\} \cup \\ &\{(X \to [Q_i]) \mid 1 \leq i \leq m\} \cup \\ &\{([Q_i] \to [Q_i], A_1 \to x'_1, ..., A_s \to x'_s) \mid 1 \leq i \leq m, \\ &Q_i = \{P_{j_1}, ..., P_{j_s}\},\ (A_1 \to x_1, ..., A_s \to x_s) \in R_i, \\ &x'_r \text{ is obtained by replacing each nonterminal in } x_r \\ &\text{by its primed version},\ 1 \leq r \leq s\} \cup \end{aligned}$$

$$\{(A' \to A) \mid A \in N\} \cup \{([Q_i] \to R_{i,1}) \mid 1 \le i \le m\} \cup$$
$$\big\{(R_{i,j} \to R'_{i,j+1}, A_1 \to \alpha_1, ..., A_s \to \alpha_s) \mid 1 \le i \le m,\ 1 \le j \le k_i - 1,$$
$$\pi_{i,j} = (A_1 \to x_1, ..., A_s \to x_s),\ \alpha_r \in \{A'_r, \#\},\ 1 \le r \le s,$$
$$\text{and for at least one } r, 1 \le r \le s, \text{ we have } \alpha_r = \#\} \cup$$
$$\{(R_{i,k_i} \to X', A_1 \to \alpha_1, ..., A_s \to \alpha_s) \mid 1 \le i \le m,$$
$$\pi_{i,k_i} = (A_1 \to \alpha_1, ..., A_s \to \alpha_s),\ \alpha_r \in \{A'_r, \#\}, 1 \le r \le s,$$
$$\text{and for at least one } r, 1 \le r \le s, \text{ we have } \alpha_r = \#\} \cup$$
$$\big\{(R'_{i,j} \to R_{i,j}, A'_1 \to \#, ..., A'_p \to \#) \mid 1 \le i \le m,$$
$$1 \le j \le k_i, \{A_1, ..., A_p\} = N\} \cup$$
$$\big\{(X' \to X, A'_1 \to \#, ..., A'_p \to \#) \mid \{A_1, ..., A_p\} = N\big\} \cup$$
$$\{(X \to c)\}.$$

The set F contains all rules of the form $A \to \#$ in the previous matrices.

The derivation starts form wX, $w \in W$. In general, from a sentential form $wX, w \in (N \cup T)^*$, in a non-deterministic way we can pass to $w[Q_i]$ in order to start the simulation of the team Q_i. Using a matrix

$$([Q_i] \to [Q_i], A_1 \to x'_1, ..., A_s \to x'_s)$$

corresponds to a derivation step in Q_i (the primed symbols in $x'_1, ..., x'_s$ ensure the parallel mode of using the rules $A_1 \to x_1, ..., A_s \to x_s$. The primed symbols can be replaced freely by their originals using the matrices $(A' \to A)$. The symbol $[Q_i]$ can be changed only by passing through $R_{i,1}, ..., R_{i,k_i}$, which checks the correct termination of the derivation in Q_i, in the sense of the mode t_0 of derivation: we can pass from $R_{i,j}$ to $R'_{i,j+1}$, if and only if the corresponding sequence $\pi_{i,j}$ of rules cannot be used (otherwise a symbol $\#$ will be introduced, because for each sequence

$$A_1 \to \alpha_1, ..., A_s \to \alpha_s,$$

at least one α_k is $\#$). After obtaining a sequence $\pi_{i,j}$, we introduce the symbol $R'_{i,j+1}$, which is replaced by $R_{i,j+1}$ only after having replaced all primed symbols A' with $A \in N$ by their original A. Then we can pass to checking the sequence $\pi_{i,j+1}$. If none of the sequences $\pi_{i,j}, 1 \le j \le k_i$, can be used, we can introduce the symbol X' and then X; in this way the derivation in Q_i is successfully simulated, and we can pass, in a non-deterministic way, to another team.
When the matrix $(X \to c)$ is used, the string must not contain any further non-terminal, because no matrix can be used any more .
In conclusion, $L(G) = L_{t_0}(\Gamma)\{c\}$. As MAT_{ac} is closed under right derivative, it follows that $L_{t_0}(\Gamma) \in MAT_{ac}$.

A similar construction like that elaborated above shows that for a CD grammar system with prescribed teams and arbitrary context-free rules we can construct a matrix grammar

$$G = (N', T, S', M, F)$$

with arbitrary context-free rules such that $L(G) = L_{t_0}(\Gamma)$; observe that we do not need the additional terminal symbol c, because in the case of arbitrary context-free rules we can simply replace the matrix $(X \to c)$ by the matrix $(X \to \lambda)$. As an immediate consequence, we obtain $L_{t_0}(\Gamma) \in MAT^{\lambda}_{ac}$. □

Lemma 3. $MAT_{ac} \subseteq T_2CD(t_0)$ and $MAT_{ac}^{\lambda} \subseteq T_2CD^{\lambda}(t_0)$.
Proof. Let $L \subseteq V^*$ be a matrix language in MAT_{ac}. We can write

$$L = (L \cap \{\lambda\}) \cup \bigcup_{c \in V} \delta_c^r(L)\,\{c\}\,,$$

where $\delta_x^r(L)$ denotes the right derivative of L with respect to the string x.
The family MAT_{ac} is closed under right derivative, hence $\delta_c^r(L) \in MAT_{ac}$. For each $c \in V$, let $G_c = (N_c, V, S_c, M_c, F_c)$ be a matrix grammar for $\delta_c^r(L)$, and moreover, we suppose that G_c is in the accurrate normal form [Dassow, Păun 1989]:

1. $N_c = N_{c,1} \cup N_{c,2} \cup \{S, \#\}$, where $N_{c,1}, N_{c,2}, \{S, \#\}$ are pairwise disjoint.
2. The matrices in M_c are of one of the following forms:
   a. $(S_c \to w)$, $w \in V^*$;
   b. $(S_c \to AX)$, $A \in N_1$, $X \in N_2$;
   c. $(A \to w, X \to Y)$, $A \in N_1$, $w \in (N_1 \cup V)^+$, $X, Y \in N_2$;
   d. $(A \to \#, X \to Y)$, $A \in N_1$, $X, Y \in N_2$;
   e. $(A \to a, X \to b)$, $A \in N_1$, $X \in N_2$, $a, b \in V$.
3. The set F_c consists of all rules $A \to \#$ appearing in matrices of M_c.

Without loss of generality we may also assume that $|w|_{\{A\}} = 0$ and $X \neq Y$ in matrices of the forms c (if we have a matrix $(A \to w, X \to Y)$ with $|w|_{\{A\}} \neq 0$ or $X = Y$ we can replace it by the sequence of matrices

$$\begin{array}{l}(A \to A_1, X \to X_1),\ (A_k \to w_k, X_k \to X_{k+1}),\ 1 \le k \le m-1,\\ (A_m \to w_m, X_m \to X),\end{array}$$

where $w = w_1 ... w_m$, $w_k \in V$, $1 \le k \le m$, as well as A_k and X_k with $1 \le k \le m$ are new symbols to be added to N_1 and N_2, respectively); in a similar way, we can assume that $X \neq Y$ in a matrix $(A \to \#, X \to Y)$ of form d (a matrix $(A \to \#, X \to X)$ can be replaced by the matrices $(A \to \#, X \to X_1)$ and $(A \to \#, X_1 \to X)$, where X_1 is a new symbol to be added to N_2,).
We take such a matrix grammar G_c for every language $\delta_c^r(L) \neq \emptyset$, $c \in V$; without loss of generality, we may assume that the sets $N_{c,1}$, $N_{c,2}$, $c \in V$, are pairwise disjoint.
Assume all matrices of the forms c, d, e in the sets M_c to be labelled in a one-to-one manner such that the labels used for M_c are different from those used for $M_{c'}$, $c' \neq c$, and let Lab_c, Lab_d, Lab_e, be the set of all the corresponding labels as well as

$$Lab = Lab_c \cup Lab_d \cup Lab_e.$$

Now consider the following sets of symbols

$$\begin{aligned} N_1 &= \bigcup_{c \in V} N_{c,1},\\ N_2 &= \bigcup_{c \in V} N_{c,2},\\ \Pi &= \{A_l, A'_l \mid A \in N_1,\ l \in Lab\}\,, \end{aligned}$$

$$\Sigma = \{X_l \mid X \in N_2,\ l \in Lab\},$$
$$\Delta = \left\{D^{(c)}, D_l^{(c)}, E_l^{(c)}, F_l^{(c)}, G_l^{(c)} \mid c \in V,\ l \in Lab\right\},$$
$$\Psi = \Pi \cup \Sigma \cup \Delta, \text{ and}$$
$$N = N_1 \cup N_2 \cup \Pi \cup \Sigma \cup \Delta.$$

We construct a CD grammar system Γ with $N \cup \{\#\}$ as the set of non-terminal symbols, V as the set of terminal symbols, the set of axioms

$$W = (L \cap \{\lambda\}) \cup \{wc \mid (S_c \to w) \in M_c,\ w \in V^*,\ c \in V\} \cup \left\{AXD^{(c)} \mid (S_c \to AX) \in M_c,\ c \in V,\ \delta_c^r(L) \neq \emptyset\right\}$$

and the components $P_{l,1}$, $Q_{l,1}$, $P_{l,2}$, $Q_{l,2}$ for $l \in Lab$ constructed as follows:

A. If $l : (A \to w, X \to Y)$ is a matrix of type c with $A \in N_1$, $w \in (N_1 \cup V)^+$, $|w|_{\{A\}} = 0$, and $X, Y \in N_2$, $X \neq Y$, then we take the components

$$P_{l,1} = \{X \to X_l, A \to A_l\} \cup \{\beta \to \# \mid \beta \in N\},$$
$$Q_{l,1} = \left\{D^{(c)} \to D_l^{(c)}, D_l^{(c)} \to E_l^{(c)}\right\} \cup \left\{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \left\{A_l, X_l, E_l^{(c)}\right\}\right\},$$
$$P_{l,2} = \{A_l \to w, X_l \to Y\} \cup \{\beta \to \# \mid \beta \in N\},$$
$$Q_{l,2} = \left\{E_l^{(c)} \to F_l^{(c)}, F_l^{(c)} \to D^{(c)}\right\} \cup \left\{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \left\{D^{(c)}, Y\right\}\right\}.$$

B. If $l : (A \to a, X \to b)$ is a matrix of type e, with $A \in N_1$, $a \in V$, $X \in N_2$, $b \in V$, then we take the components

$$P_{l,1} = \{X \to X_l, A \to A_l\} \cup \{\beta \to \# \mid \beta \in N\},$$
$$Q_{l,1} = \left\{D^{(c)} \to D_l^{(c)}, D_l^{(c)} \to E_l^{(c)}\right\} \cup \left\{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \left\{A_l, X_l, E_l^{(c)}\right\}\right\},$$
$$P_{l,2} = \{A_l \to A'_l, A'_l \to a, X_l \to b\} \cup \{\beta \to \# \mid \beta \in N\},$$
$$Q_{l,2} = \left\{E_l^{(c)} \to F_l^{(c)}, F_l^{(c)} \to G_l^{(c)}, G_l^{(c)} \to c\right\} \cup \{\beta \to \# \mid \beta \in N\}.$$

C. If $l : (A \to \#, X \to Y)$ is a matrix of type d (hence with $A \to \# \in F_c$), with $A \in N_1$, $X, Y \in N_2$, $X \neq Y$, then we take the components

$$P_{l,1} = \{X \to X_l\} \cup \{\beta \to \# \mid \beta \in N\},$$
$$Q_{l,1} = \left\{D^{(c)} \to E_l^{(c)}\right\} \cup \left\{\beta \to \# \mid \beta \in (\Psi \cup N_2 \cup \{A\}) - \left\{E_l^{(c)}, X_l\right\}\right\},$$
$$P_{l,2} = \{X_l \to Y\} \cup \{\beta \to \# \mid \beta \in N\},$$
$$Q_{l,2} = \left\{E_l^{(c)} \to D^{(c)}\right\} \cup \left\{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \left\{D^{(c)}, Y\right\}\right\}.$$

Let us give some remarks on these constructions:

- The intended *legal teams* of two components are $\{P_{l,1}, Q_{l,1}\}$ and $\{P_{l,2}, Q_{l,2}\}$ for arbitrary labels $l \in Lab$ (which would already solve the problem for prescribed teams of size two); all other pairs of components cannot work in the mode t_0 without introducing the trap-symbol $\#$.
- The symbol $\#$ is a trap-symbol and every component contains rules $\beta \to \#$ for "almost all" symbols $\beta \in N$; the termination of a derivation sequence with a legal team is only guaranteed by the "exceptions" in the components of type Q.

– In order to assure the correct pairing of components, we use variants of the control symbol D ($D^{(c)}$, $D_l^{(c)}$, $E_l^{(c)}$, $F_l^{(c)}$, $G_l^{(c)}$), as well as subscripts added to symbols in N_1 (leading to symbols in Π) and to symbols in N_2 (leading to symbols in Σ).

We now show that Γ with free teams of constant size two in the same way as with the prescribed teams of size two described above generates L :

Claim 1. $L \subseteq L_{t_0}(\Gamma, 2)$.

As the "short strings" in L are directly introduced in W, it is enough to prove that every derivation step in a grammar G_c can be simulated by the teams of Γ. More exactly, we shall prove that if $z_1 \Longrightarrow_{G_c} z_2$ is a derivation step in G_c, where z_2 is not a terminal string, then $z_1 D^{(c)} \Longrightarrow_\Gamma^* z_2 D^{(c)}$ in a derivation sequence using teams of size 2 from Γ, and that if z_2 is a terminal string, then $z_1 D^{(c)} \Longrightarrow_\Gamma^* z_2 c$ in a derivation sequence using teams of size 2 from Γ.

If

$$z_1 = x_1 A x_2 X \Longrightarrow_{G_c} x_1 w x_2 Y = z_2$$

by a matrix $l : (A \to w, X \to Y)$ of type c, then

$$x_1 A x_2 X D^{(c)} \Longrightarrow_{\{P_{l,1}, Q_{l,1}\}} x_1 A_l x_2 X D_l^{(c)} \Longrightarrow_{\{P_{l,1}, Q_{l,1}\}} x_1 A_l x_2 X_l E_l^{(c)}$$

and no more step is possible with this team $\{P_{l,1}, Q_{l,1}\}$, hence

$$x_1 A x_2 X D^{(c)} \Longrightarrow_{\{P_{l,1}, Q_{l,1}\}}^{t_0} x_1 A_l x_2 X_l E_l^{(c)}.$$

Now, also in two steps, we obtain

$$x_1 A_l x_2 X_l E_l^{(c)} \Longrightarrow_{\{P_{l,2}, Q_{l,2}\}}^{t_0} x_1 w x_2 Y D^{(c)} = z_2 D^{(c)}.$$

In a similar way, if for a terminal string

$$z_1 = x_1 A x_2 X \Longrightarrow_{G_c} x_1 a x_2 b = z_2$$

by a matrix $l : (A \to a, X \to b)$ of type e, then we obtain

$$x_1 A x_2 X D^{(c)} \Longrightarrow_{\{P_{l,1}, Q_{l,1}\}} x_1 A_l x_2 X D_l^{(c)} \Longrightarrow_{\{P_{l,1}, Q_{l,1}\}} x_1 A_l x_2 X_l E_l^{(c)}$$

and

$$\begin{aligned} x_1 A_l x_2 X_l E_l^{(c)} &\Longrightarrow_{\{P_{l,2}, Q_{l,2}\}} x_1 a x_2 X_l F_l^{(c)} \Longrightarrow_{\{P_{l,2}, Q_{l,2}\}} x_1 a x_2 X_l' G_l^{(c)} \\ &\Longrightarrow_{\{P_{l,2}, Q_{l,2}\}} x_1 a x_2 b c \end{aligned}$$

i.e.

$$x_1 A x_2 X D^{(c)} \Longrightarrow_{\{P_{l,1}, Q_{l,1}\}}^{t_0} x_1 A_l x_2 X_l E_l^{(c)} \Longrightarrow_{\{P_{l,2}, Q_{l,2}\}}^{t_0} x_1 a x_2 b c = z_2 c.$$

If

$$z_1 = xX \Longrightarrow_{G_c} xY = z_2$$

by a matrix $l : (A \to \#, X \to Y)$ of type d, then

$$xXD^{(c)} \Longrightarrow_{\{P_{l,1}, Q_{l,1}\}}^{t_0} xX_l E_l^{(c)} \Longrightarrow_{\{P_{l,2}, Q_{l,2}\}}^{t_0} xYD^{(c)}.$$

Observe that from $xX_lE_l^{(c)}$ no further derivation step with the team $\{P_{l,1}, Q_{l,1}\}$ is possible if and only if $|x|_{\{A\}} = 0$.
In conclusion, every derivation in a grammar G_c can be simulated in Γ by applying a suitable sequence of appropriate teams of pairs of components, which completes the proof of claim 1.

Using *legal teams,* i.e. the teams $\{P_{l,1}, Q_{l,1}\}$ and $\{P_{l,2}, Q_{l,2}\}$, we can only obtain the following sentential forms not containing the trap symbol # (we call them *legal configurations*):

1. $xXD^{(c)}$, with $x \in (N_1 \cup V)^+$, $X \in N_2$, $c \in V$ (initially we have $x \in N_1$).
2. $xA_lx'X_lE_l^{(c)}$, with $x, x' \in (N_1 \cup V)^*$, $A \in N_1$, $X \in N_2$, $c \in V$, $l \in Lab_c \cup Lab_e$, i.e. l being a label of a matrix of type c or e.
3. $xX_lE_l^{(c)}$, with $x \in (N_1 \cup V)^+$, $X \in N_2$, $c \in V$, $l \in Lab_d$, i.e. l being a label of a matrix of type d.

Claim 2. Starting from an arbitrary legal configuration, every illegal team will introduce the symbol #.

First of all we have to notice that in the following we can restrict our attention to components associated with some matrix from M_c, because components associated with some matrix from $M_{c'}$ with $c' \neq c$ already at the first application of a rule force us to introduce the trap symbol #. For the same reasons, we need not take into account teams consisting of two components of type Q : they cannot work together without introducing #, because they can only replace symbols in Δ by symbols different from #.

For the rest of possible illegal teams of size two we consider the following three cases according to the three types of legal configurations:
Case 1: Configuration $xXD^{(c)}$, i.e. of type 1.
Each component being of one of the types $P_{l,2}$ and $Q_{l,2}$ will introduce # at the first application of a rule; therefore it only remains to consider pairs of components of the types $P_{l,1}$ and $Q_{l,1}$ for different labels from Lab associated with matrices from M_c (the labels must be different, because otherwise either the team were legal or else the teams would not be of size two). Hence only the following teams might be possible:

1. $\{P_{l,1}, P_{l',1}\}$, where $l \neq l'$: The intermediate strings coming up during the application of such a team will contain at least one symbol X_l or A_l as well as at least one symbol $X_{l'}$ or $A_{l'}$ for the two different labels l and l', hence before the derivation with the team can terminate, at least one of the rules of the form $\beta \to \#$ (i.e. $X_l \to \#$ or $A_l \to \#$ respectively $X_{l'} \to \#$ or $A_{l'} \to \#$) is forced to be applied in at least one of the components.
2. $\{P_{l,1}, Q_{l',1}\}$, where $l \neq l'$: While the component $Q_{l',1}$ works on symbols from Δ, the other component $P_{l,1}$ introduces at least one symbol X_l or A_l. As $P_{l,1}$ contains all rules $\beta \to \#$ for $\beta \in \Delta$ (and no other rules for $\beta \in \Delta$) and $Q_{l',1}$ contains all rules $\alpha \to \#$ for $\alpha \in \{X_l, A_l\}$ (and no other rules for X_l, A_l), the derivation with the team $\{P_{l,1}, Q_{l',1}\}$ cannot terminate without a step introducing the symbol # by at least one of the components.

In all cases, further derivations are blocked (they never can lead to terminal strings) because the trap-symbol # has been forced to be introduced.

Case 2: Configuration $xA_lx'X_lE_l^{(c)}$, for $l \in Lab_c \cup Lab_e$.
The only components that may not be forced to introduce # by the first rule they can apply are $P_{l',1}$ for any arbitrary $l' \in Lab_c \cup Lab_e$ as well as $P_{l,2}$ and $Q_{l,2}$. Hence only the following teams might be possible:

1. $\{P_{l',1}, P_{l'',1}\}$, where $l' \neq l''$ (otherwise the team would not be of size two): $P_{l',1}$ will introduce some $A_{l'}$, $A \in N_1$, and $P_{l'',1}$ will introduce some $B_{l''}$, $B \in N_1$, therefore further derivations are blocked by introducing the trap symbol # with $A_{l'} \to \#$ or $B_{l''} \to \#$ in $P_{l',1}$ or in $P_{l'',1}$.
2. $\{P_{l',1}, P_{l,2}\}$: $P_{l,2}$ (in two or three steps) can replace A_l and X_l; in the meantime $P_{l',1}$ must introduce some $B_{l'}$, $B \in N_1$.
   (a) If $l' \neq l$, then from $xA_lx'X_lE_l^{(c)}$ in two steps (if a second step in $P_{l',1}$ is possible without introducing #) we obtain $y_1B_{l'}y_2B_{l'}y_3UE_l^{(c)}$, where $U \in N_2$ if $l \in Lab_c$ and $U \in N_2 \cup \{X_l\}$ if $l \in Lab_e$.
      i. If $l \in Lab_c$, then in the third step at least $P_{l,2}$ now must use a rule introducing the trap symbol #, e.g. $B_{l'} \to \#$, whereas $P_{l',1}$, if not also being forced to use such a trap rule, may be able to use $B \to B_{l'}$ once again or $U \to U_{l'}$, if it just happens that U is the right symbol from N_2 that can be handled by $P_{l',1}$.
      ii. If $l \in Lab_e$, then $X_l \to b$ or $A_l' \to a$ from $P_{l,2}$ can be applied in the third step, but even if $P_{l',1}$ can replace a third occurrence of B by B_l', at least in the fourth step $P_{l,2}$ now is forced to introduce the trap symbol #, e.g. by $B_l' \to \#$.

   (b) If $l' = l$, i.e. if we use the team $\{P_{l,1}, P_{l,2}\}$, then again we have to distinguish between two subcases:
      i. If $l \in Lab_c$, then $P_{l,1}$ can replace all occurences of A by A_l, while $P_{l,2}$ can replace X_l by Y and A_l by w. As we have assumed $Y \neq X$, no other rule not introducing # than $A \to A_l$ can be used in $P_{l,1}$. Moreover we also have assumed the rule $A_l \to w$ to be non-recursive, i.e. $|w|_{\{A\}} = 0$, hence after a finite number of derivation steps with the team $\{P_{l,1}, P_{l,2}\}$ the occurrences of A will be exhausted, so finally a rule introducing the trap symbol # must be used by $P_{l,1}$ (one possible candidate is $E_l^{(c)} \to \#$), while $P_{l,2}$ can use $A_l \to w$ or $X_l \to Y$ (if this rule has not yet been used before).
      ii. If $l \in Lab_e$, we face a similar situation as above except that from A_l two steps are needed in $P_{l,2}$ in order to obtain a from A_l.
3. $\{P_{l',1}, Q_{l,2}\}$: While $E_l^{(c)} \to F_l^{(c)}$ is used in $Q_{l,2}$, $P_{l',1}$ must introduce some $A_{l'}$, $A \in N_1$; if no more non-terminal symbol A is available in the current sentential form, when $Q_{l,2}$ uses its rule for replacing $F_l^{(c)}$, $P_{l',1}$ will have to use a trap rule like $A_{l'} \to \#$; if $P_{l',1}$ can introduce one more $A_{l'}$, then finally a trap rule like $A_{l'} \to \#$ must be applied by at least one of the components $P_{l',1}$ and $Q_{l,2}$ before the derivation can terminate.

Case 3: Configuration $xX_lE_l^{(c)}$, for $l \in Lab_d$.

The only components that do not introduce # by the first rule they can apply are $P_{l',1}$ for any arbitrary $l' \in Lab_c \cup Lab_e$ (and therefore $l' \neq l$) as well as $P_{l,2}$ and $Q_{l,2}$. Hence only the following teams might be possible:

1. $\{P_{l',1}, P_{l'',1}\}$, where $l' \neq l''$ (otherwise the team would not be of size two): $P_{l',1}$ will introduce some $A_{l'}$, $A \in N_1$, and $P_{l'',1}$ will introduce some $B_{l''}$, $B \in N_1$, therefore further derivations are blocked by introducing the trap symbol # with $A_{l'} \to \#$ or $B_{l''} \to \#$ in $P_{l',1}$ or in $P_{l'',1}$.
2. $\{P_{l',1}, P_{l,2}\}$: $P_{l,2}$ can only replace X_l by Y; in the meantime $P_{l',1}$ must introduce some $A_{l'}$, $A \in N_1$. In the second derivation step, in $P_{l,2}$ the trap rule $Y \to \#$ can be used, whereas from $P_{l',1}$ at least $A_{l'} \to \#$ can be applied.
3. $\{P_{l',1}, Q_{l,2}\}$: While $E_l^{(c)} \to D^{(c)}$ is used in $Q_{l,2}$, $P_{l',1}$ must introduce some $A_{l'}$, $A \in N_1$; but then $Q_{l,2}$ has to use a trap rule like $X_l \to \#$, while $P_{l',1}$ can use $A \to A_{l'}$ once more or at least $A_{l'} \to \#$.

In conclusion, only the legal teams can be used without introducing the trap symbol #; they simulate matrices in the sets M_c, $c \in V$, hence also the inclusion $L_{t_0}(\Gamma, 2) \subseteq L$ is true, which completes the proof of $MAT_{ac} \subseteq T_2CD(t_0)$.

Now let $L \subseteq V^*$ be a matrix language in MAT_{ac}^{λ}. As λ-rules are allowed in this case, we need not split up the language L in languages $\delta_c^r(L)$, $c \in V$; hence, for a matrix grammar $G = (N', V, S, M, F)$ with $L(G) = L$ we can directly construct a CD grammar system Γ such that $L_{t_0}(\Gamma, 2) = L$. Again the matrix grammar G can be assumed to be in the accurrate normal form [Dassow, Păun 1989] like in the previous case:

1. $N' = N_1 \cup N_2 \cup \{S, \#\}$, where $N_1, N_2, \{S, \#\}$ are pairwise disjoint.
2. The matrices in M are of one of the following forms:
   a. $(S \to w)$, $w \in V^*$;
   b. $(S \to AX)$, $A \in N_1$, $X \in N_2$;
   c. $(A \to w, X \to Y)$, $A \in N_1$, $w \in (N_1 \cup V)^*$, $|w|_{\{A\}} = 0$, $X, Y \in N_2$, $X \neq Y$;
   d. $(A \to \#, X \to Y)$, $A \in N_1$, $X, Y \in N_2$, $X \neq Y$;
   e. $(A \to a, X \to b)$, $A \in N_1$, $X \in N_2$, $a, b \in V \cup \{\lambda\}$.
3. The set F consists of all rules $A \to \#$ appearing in matrices of M.

In contrast to the λ-free case, matrices of the form c can also be of the form

$$(A \to \lambda, X \to Y)$$

and matrices of the form e can also be of the forms

$$(A \to \lambda, X \to \lambda),\ (A \to \lambda, X \to b),\ (A \to a, X \to \lambda), \text{ where } a, b \in V.$$

Assume all matrices of the forms c, d, e in the sets M to be labelled in a one-to-one manner and let Lab_c, Lab_d, Lab_e, be the sets of all the corresponding labels as well as

$$Lab = Lab_c \cup Lab_d \cup Lab_e.$$

Now consider the following sets of symbols

$$\begin{aligned}
\Pi &= \{A_l, A'_l \mid A \in N_1,\ l \in Lab\}\,,\\
\Sigma &= \{X_l \mid X \in N_2,\ l \in Lab\}\,,\\
\Delta &= \{D, D_l, E_l, F_l, G_l \mid l \in Lab\}\,,\\
\Psi &= \Pi \cup \Sigma \cup \Delta, \text{ and}\\
N &= N_1 \cup N_2 \cup \Pi \cup \Sigma \cup \Delta.
\end{aligned}$$

We construct a CD grammar system Γ with $N \cup \{\#\}$ as the set of non-terminal symbols, V as the set of terminal symbols, the set of axioms

$$W = \{w \mid (S \to w) \in M,\ w \in V^*\} \cup \{AXD \mid (S \to AX) \in M\}$$

and the components $P_{l,1}$, $Q_{l,1}$, $P_{l,2}$, $Q_{l,2}$ for $l \in Lab$ constructed like in the previous case:

A. If $l : (A \to w, X \to Y)$ is a matrix of type c with $A \in N_1$, $w \in (N_1 \cup V)^*$, $|w|_{\{A\}} = 0$, and $X, Y \in N_2$, $X \neq Y$, then we take the components

$P_{l,1} = \{X \to X_l, A \to A_l\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,1} = \{D \to D_l, D_l \to E_l\} \cup \{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \{A_l, X_l, E_l\}\}$,

$P_{l,2} = \{A_l \to w, X_l \to Y\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,2} = \{E_l \to F_l, F_l \to D\} \cup \{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \{D, Y\}\}$.

B. If $l : (A \to a, X \to b)$ is a matrix of type e, with $A \in N_1$, $X \in N_2$, $a, b \in V \cup \{\lambda\}$, then we take the components

$P_{l,1} = \{X \to X_l, A \to A_l\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,1} = \{D \to D_l, D_l \to E_l\} \cup \{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \{A_l, X_l, E_l\}\}$,

$P_{l,2} = \{A_l \to A'_l, A'_l \to a, X_l \to b\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,2} = \{E_l \to F_l, F_l \to G_l, G_l \to \lambda\} \cup \{\beta \to \# \mid \beta \in N\}$.

C. If $l : (A \to \#, X \to Y)$ is a matrix of type d (hence with $A \to \# \in F$), with $A \in N_1$, $X, Y \in N_2$, $X \neq Y$, then we take the components

$P_{l,1} = \{X \to X_l\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,1} = \{D \to E_l\} \cup \{\beta \to \# \mid \beta \in (\Psi \cup N_2 \cup \{A\}) - \{E_l, X_l\}\}$,

$P_{l,2} = \{X_l \to Y\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,2} = \{E_l \to D\} \cup \{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \{D, Y\}\}$.

The intended *legal teams* of two components again are $\{P_{l,1}, Q_{l,1}\}$ and $\{P_{l,2}, Q_{l,2}\}$ for arbitrary labels $l \in Lab$, the *legal configurations* are

1. xXD, with $x \in (N_1 \cup V)^+$, $X \in N_2$ (initially we have $x \in N_1$),
2. $xA_lx'X_lE_l$, for $x, x' \in (N_1 \cup V)^*$, $A \in N_1$, $X \in N_2$, $l \in Lab_c \cup Lab_e$, and
3. xX_lE_l, for $x \in (N_1 \cup V)^+$, $X \in N_2$, $l \in Lab_d$.

In contrast to the λ-free case we can use the λ-rule $G_l \to \lambda$ in the component $Q_{l,2}$ associated with a matrix $l : (A \to a, X \to b)$ of type e, which allows us to avoid the splitting up of the language L into the right derivatives $\delta^r_c(L)$, $c \in V$, yet again we obtain $L \subseteq L_{t_0}(\Gamma, 2)$: If $z_1 \Longrightarrow_G z_2$ is a derivation step in G, where z_2 is not a terminal string, then $z_1D \Longrightarrow^*_\Gamma z_2D$ in a derivation sequence using appropriate teams of size 2 from Γ, and if z_2 is a terminal string, then $z_1 D \Longrightarrow_{\Gamma}^{*} z_2$ in a derivation sequence using the appropriate teams of size 2 from Γ.

Similar arguments as in the λ-free case can be used to show that $L_{t_0}(\Gamma, 2) \subseteq L$. Hence again we obtain $L_{t_0}(\Gamma, 2) = L$, which proves $MAT_{ac}^{\lambda} \subseteq T_2CD^{\lambda}(t_0)$, too. □

Lemma 4. $MAT_{ac} \subseteq T_sCD(t_0)$ and $MAT_{ac}^{\lambda} \subseteq T_sCD^{\lambda}(t_0)$ for $s \in \{+, *\}$.

Proof. For a language L in MAT_{ac} respectively MAT_{ac}^{λ} we just take the adequate CD grammar system Γ already constructed in the proof of the previous lemma. As the legal teams of size two still are available, we obviously obtain $L \subseteq L_{t_0}(\Gamma, s)$. On the other hand, we still have $L_{t_0}(\Gamma, s) \subseteq L$, too, although the possibilities for forming teams from the constructed components have increased considerably. Yet we have to adapt our arguments according to this new situation.
As in the previous proof we still have to notice that in the following we can restrict our attention to teams where all components are associated with matrices from only one set M_c, because components associated with some matrix from another set of matrices $M_{c'}$ with $c' \neq c$ already at the first application of a rule force us to introduce the trap symbol #. Hence in the following it is sufficient to consider the case where L in MAT_{ac}^{λ}.

As the *legal teams* of two components again are $\{P_{l,1}, Q_{l,1}\}$ and $\{P_{l,2}, Q_{l,2}\}$ for appropriate labels $l \in Lab$, the *legal configurations* are

1. xXD, with $x \in (N_1 \cup V)^+$, $X \in N_2$ (initially we have $x \in N_1$),
2. $xA_l x' X_l E_l$, for $x, x' \in (N_1 \cup V)^*$, $A \in N_1$, $X \in N_2$, $l \in Lab_c \cup Lab_e$, and
3. $xX_l E_l$, for $x \in (N_1 \cup V)^+$, $X \in N_2$, $l \in Lab_d$.

As teams of type Q cannot work together without introducing #, because they can only replace symbols in Δ by symbols different from #, we need not take into account teams containing at least two components of type Q. Moreover, every team of size one finally is forced to introduce the trap symbol # when started on a legal configuration (i.e. the result for $s = *$ is the same as for $s = +$). Hence in the following we now take a closer look on every possible combination of components yielding a team with at least three components and allowing at least one derivation step on the legal configurations listed above without introducing the trap symbol # :

Case 1: Configuration $xXD^{(c)}$, i.e. of type 1.
Each component being of one of the types $P_{l,2}$ and $Q_{l,2}$ will introduce # at the first application of a rule; therefore it only remains to consider teams T where each component is of one of the types $P_{l,1}$ and $Q_{l,1}$.

1. T contains only components of the type $P_{l,1}$, where obviously all labels of these components have to be different. Then at most one label can be from Lab_d, because such a component only once can use the rule $X \rightarrow X_l$, whereas all the other components $P_{l,1}$ with $l \in Lab_c \cup Lab_e$ can also apply a rule to the symbol X at most once as well as the rule $A \rightarrow A_l$ to any occurrence of the corresponding symbol A. Hence, before the derivation with the team can terminate, at least one of the rules of the form $\beta \rightarrow \#$ (e.g. $X_l \rightarrow \#$ or $A_l \rightarrow \#$) is forced to be applied in at least one of the components.

2. T contains exactly one component $Q_{m,1}$, wheras all the other components are of the type $P_{l,1}$, i.e.

$$T = \{Q_{m,1}\} \cup \{P_{l_i,1} \mid 1 \leq i \leq k\},$$

where $k \geq 2$. Denote $Lab_P(T) = \{l_i \mid 1 \leq i \leq k\}$. Again, at most one label in $Lab_P(T)$ can be from Lab_d. In the first derivation step with the team T, $D \to D_m$ for $m \in Lab_c \cup Lab_e$ respectively $D \to E_m$ for $m \in Lab_d$ from $Q_{m,1}$ is used, while at most one component P_{l_j} can use the rule $X \to X_{l_j}$, whereas all the others have to use the rules $A \to A_{l_i}$ for the corresponding non-terminal symbols $A \in N_1$.

(a) If $m \in Lab_d$, in the next step $Q_{m,1}$ has to use a trap rule.

i. If the rule $X \to X_{l_j}$ has been applied in the first step (observe that $l_j \in Lab_d$ if $Lab_P(T) \cap Lab_d \neq \emptyset$), then even if $l_j = m$, every component of T can apply a rule, i.e. for $Q_{m,1}$ we choose $A_{l_{i_0}} \to \#$ for some $l_{i_0} \in Lab_P(T) - Lab_d$, for $P_{l_j,1}$ we can take $X_{l_j} \to \#$, from $P_{l_{i_0},1}$ at least $E_m \to \#$ can be applied, and in the remaining components $P_{l_i,1}$, $l_i \in Lab_P(T) - \{l_j, l_{i_0}\}$ at least $A_{l_i} \to \#$ is applicable.

ii. If the rule $X \to X_{l_j}$ has not been applied in the first step, i.e. for each $l_i \in Lab_P(T)$ the rule $A \to A_{l_i}$ has been taken from $P_{l_i,1}$ (which implies $Lab_P(T) \cap Lab_d = \emptyset$ and therefore $m \notin Lab_P(T)$, too), again a second step with T is possible: We can choose $X \to \#$ from $Q_{m,1}$, while at least $A_{l_i} \to \#$ is applicable in the remaining components P_{l_i}, $l_i \in Lab_P(T)$.

(b) If $m \in Lab_c \cup Lab_e$, then one more derivation step may be possible without $Q_{m,1}$ being forced to use a trap rule, but again in any case the trap symbol $\#$ must be introduced before the derivation can terminate, even if T contains the legal team $\{Q_{m,1}, P_{m,1}\}$:

i. If $Lab_P(T) \cap Lab_d \neq \emptyset$, i.e. $l_j \in Lab_P(T) \cap Lab_d$, then only one derivation step with T is possible without introducing $\#$, and in this step $P_{l_j,1}$ has used the rule $X \to X_{l_j}$, whereas all the other $P_{l_i,1}$, $l_i \in Lab_P(T) - \{l_j\}$, had to use $A \to A_{l_i}$ for the corresponding symbols $A \in N_1$, and $Q_{m,1}$ used $D \to D_m$. $P_{l_i,1}$ now is forced to use a trap rule like $X_{l_j} \to \#$ (observe that $m \neq l_j$), whereas $Q_{m,1}$ can use its rule for replacing D_m and the other components $P_{l_i,1}$, $l_i \in Lab_P(T) - \{l_j\}$, at least can apply $A_{l_i} \to \#$.

ii. If $Lab_P(T) \cap Lab_d = \emptyset$, then at most two derivation steps with the team T are possible without introducing the trap symbol $\#$, where at most once one component $P_{l_j,1}$ can apply $X \to X_{l_j}$, whereas otherwise the components $P_{l_i,1}$, have to use the corresponding rules $A \to A_{l_i}$, while $Q_{m,1}$ can use $D \to D_m$ and $D_m \to E_m$.

A. If two derivation steps without applying a trap rule have been possible, then also a third step with the team T is possible, where $Q_{m,1}$ is forced to apply a trap rule, e.g. for $Q_{m,1}$ we can choose $A_{l_{i_0}} \to \#$, where $l_{i_0} \in Lab_P(T)$ such that X has not been replaced by $X_{l_{i_0}}$, whereas all the components $P_{l_i,1}$, $l_i \in Lab_P(T)$, can at least apply $A_{l_i} \to \#$.

B. If only one derivation step without introducing # has been possible, i.e. at least one component $P_{l_{i_0},1}$ cannot apply a rule not introducing # any more, then a second step with T is possible, where $Q_{m,1}$ uses $D_m \to E_m$ and $P_{l_{i_0},1}$ is forced to apply a trap rule. If X has not been replaced in the first derivation step, $P_{l_{i_0},1}$ can apply $A_{l_{i_0}} \to \#$, while also the other components $P_{l_i,1}$ with $l_i \in Lab_P(T) - \{l_{i_0}\}$ at least can use $A_{l_i} \to \#$; if $X \to X_{l_{i_0}}$ has been applied in the first step, in the second step from $P_{l_{i_0},1}$ we can choose $X_{l_{i_0}} \to \#$ instead of $A_{l_{i_0}} \to \#$; if $X \to X_{l_j}$ has been applied in the first step for some $l_j \neq l_{i_0}$, then we can choose $A_{l_{i_0}} \to \#$ from $P_{l_{i_0},1}$, from $P_{l_j,1}$ at least $X_{l_j} \to \#$ can be applied, while from $P_{l_i,1}$ with $l_i \in Lab_P(T) - \{l_{i_0}, l_j\}$ at least $A_{l_i} \to \#$ is applicable.

In all cases, further derivations are blocked (they never can lead to terminal strings), because the trap-symbol # has been forced to be introduced.

Case 2: Configuration $xA_lx'X_lE_l$, for $l \in Lab_c \cup Lab_e$.
The only components that do not introduce # by the first rule to be applied are $P_{l',1}$ for any arbitrary $l' \in Lab_c \cup Lab_e$ as well as $P_{l,2}$ and $Q_{l,2}$. Hence only the following teams T might be possible:

1. $T = \{P_{l_i,1} \mid 1 \leq i \leq k\}$, where $k \geq 3$ and $\{l_i \mid 1 \leq i \leq k\} \subseteq Lab_c \cup Lab_e$.
   The components $P_{l_i,1}$ cannot replace the symbols A_l, X_l, E_l without introducing #, hence they will introduce A_{l_i} and therefore finally at least one component will have to use the trap rule $A_{l_i} \to \#$.
2. $T = \{P_{l,2}\} \cup \{P_{l_i,1} \mid 1 \leq i \leq k\}$, where $k \geq 2$ and $\{l_i \mid 1 \leq i \leq k\} \subseteq Lab_c \cup Lab_e$.
   Denote $Lab_P(T) = \{l_i \mid 1 \leq i \leq k\}$.
   (a) $l \notin Lab_P(T)$. While $P_{l,2}$ can replace A_l or X_l in the first step, the components $P_{l_i,1}$, $l_i \in Lab_P(T)$, can only use the corresponding rules $A \to A_{l_i}$ in order not to introduce #. If some $P_{l_{i_0},1}$ cannot use a rule not introducing # any more after this first step, at least this component is forced to use a trap rule like $A_{l_{i_0}} \to \#$, while also the other components $P_{l_i,1}$, $l_i \in Lab_P(T)$, can apply at least $A_{l_i} \to \#$ (and $P_{l,2}$ can replace the symbol from$\{A_l, X_l\}$ not affected in the first step). If two steps without introducing # are possible with the team T, then all together $2k$ symbols A_{l_i} have been introduced. As $2k > k+1$, these symbols guarantee that a trap rule must be applied, before the derivation with T can terminate.
   (b) $l \in Lab_P(T)$, i.e. $\{P_{l,1}, P_{l,2}\} \subset T$.
      i. If $l \in Lab_c$, then $P_{l,1}$ can replace all occurrences of A by A_l, while $P_{l,2}$ can replace X_l by Y and A_l by w. As we have assumed $Y \neq X$, no other rule not introducing # than $A \to A_l$ can be used in $P_{l,1}$. Moreover, as we also have assumed the rule $A_l \to w$ to be non-recursive, i.e. $|w|_{\{A\}} = 0$, the occurrences of the symbol A will be exhausted after a finite number of steps with the team T, so finally at least $P_{l,1}$ will be forced to use a trap rule. The other components $P_{l_i,1}$, $l_i \in Lab_P(T) - \{l\}$, in the first step can only apply the corresponding rules $A \to A_{l_i}$ and in the succeeding steps one of these components once also might be able to apply a rule to Y.

Now let s be the number of steps that are possible with the team T without introducing $\#$. If $s = 1$, then $P_{l,2}$ has replaced X_l by Y or A_l by w, whereas all the components $P_{l_i,1}$, $l_i \in Lab_P(T)$, have introduced one symbol A_{l_i}. Hence, in the current sentential form $k-1$ symbols A_{l_i} for $l_i \in Lab_P(T) - \{l\}$ are present as well as Y and two symbols A_l respectively X_l and A_l, i.e. at least $k+1$ non-terminal symbols, which guarantees that a second derivation step is possible, where at least one trap symbol $\#$ is introduced. If $s \geq 2$, then at least one symbol from $N_2 \cup \sum$, one symbol from Δ and $s(k-1)+1$ symbols A_m with $m \in Lab_P(T)$ occur in the current sentential form. As $s(k-1)+1+2 \geq 2(k-1)+3 \geq k+1$, again another derivation step introducing $\#$ is possible in any case.

ii. If $l \in Lab_e$, we face a similar situation except that for A_l we need two steps in order to obtain a from A_l by using $A_l \to A_l'$ and $A_l' \to a$ in $P_{l,2}$ (i.e. the symbols A_l cannot be "consumed" so fast by $P_{l,2}$ as in the previous case) and moreover, after one step the symbol from $N_2 \cup \sum$ may have vanished, so no other $P_{l_i,1}$ can use a rule on a symbol from $N_2 \cup \sum$. Let s again denote the number of steps possible with T without introducing $\#$; for all s exactly $s(k-1)$ symbols A_{l_i}, $l_i \in Lab_P(T) - \{l\}$, appear in the current sentential form. For $s \geq 3$, $s(k-1) \geq 3k-3 \geq k+1$, which guarantees that after these s steps another derivation step introducing the trap symbol $\#$ can be applied. For $s \leq 2$, we have at least $k-1$ such symbols as well as additional non-terminal symbols appearing in the current sentential form, i.e. one symbol from Δ as well as at least one symbol X_l, A_l' or A_l.

3. $T = \{Q_{l,2}\} \cup \{P_{l_i,1} \mid 1 \leq i \leq k\}$, where $k \geq 2$ and $\{l_i \mid 1 \leq i \leq k\} \subseteq Lab_c \cup Lab_e$.
While $Q_{l,2}$ uses $E_l \to F_l$ etc. the components $P_{l_i,1}$, $l_i \in Lab_P(T)$, can only apply the corresponding rules $A \to A_{l_i}$. Even if $l \in Lab_e$, the symbol from Δ can only vanish in the third derivation step with the team T, i.e. in any case, after at most two (for $l \in Lab_c$) respectively at most three (for $l \in Lab_e$) derivation steps without introducing $\#$ we are forced to use a trap rule in a further derivation step, which is always possible, because the number of non-terminal symbols in the current sentential form in all cases is at least $k+1$ (observe that also for $l \in Lab_c$ we can always find a non-terminal symbol $\notin \{D, Y\}$ for $Q_{l,2}$).
4. $T = \{P_{l,2}, Q_{l,2}\} \cup \{P_{l_i,1} \mid 1 \leq i \leq k\}$, where $k \geq 1$ and $\{l_i \mid 1 \leq i \leq k\} \subseteq Lab_c \cup Lab_e$, i.e. T contains the legal team $\{P_{l,2}, Q_{l,2}\}$.
Denote $Lab_P(T) = \{l_i \mid 1 \leq i \leq k\}$.
Because of the presence of $Q_{l,2}$, the legal subteam $\{P_{l,2}, Q_{l,2}\}$ can only make two (for $l \in Lab_c$) respectively three (for $l \in Lab_e$) derivation steps without introducing $\#$.
If at least one derivation step without introducing $\#$ is possible, besides A_l, X_l, and E_l in $xA_lx'X_lE_l$ at least k non-terminal symbols must be present for allowing the components $P_{l_i,1}$, $l_i \in Lab_P(T)$, to use the corresponding rules $B \to B_{l_i}$. Even after applying $A_l \to w$ (if $l \in Lab_c$) respectively $X_l \to b$ (if $l \in Lab_e$) in $P_{l,2}$, at least $k+2$ non-terminal symbols are left to guarantee another derivation step, if at least one component is already forced to apply a trap rule after the first derivation step.

(a) $l \in Lab_c$. Then at most a second derivation step without introducing $\#$ is possible. After this second derivation step, again at least $k+2$ non-terminal symbols are left in the current sentential form:

i. $l \in Lab_P(T)$:

A. If we have applied $A_l \rightarrow w$ in the first step from $P_{l,2}$, in the second step again we may apply $A_l \rightarrow w$, but all together we have $2k-1 \geq k$ non-terminal symbols from Π left in the current sentential form, i.e. together with X_l and D these are $k+2$ non-terminal symbols allowing a third derivation step introducing $\#$.

B. If we have applied $X_l \rightarrow Y$ in the first derivation step, the current sentential form contains *two* symbols A_l and k symbols A_{l_i} for the labels $l_i \in Lab_P(T)$ as well as the control symbol F_l. Even if some $l_j \in Lab_P(T)$ can apply the rule $Y \rightarrow Y_{l_j}$ in the second step, $Q_{l,2}$ has to use $F_l \rightarrow D$, $P_{l,2}$ has to apply $A_l \rightarrow w$ (which consumes only one of the two symbols A_l), and all the other components $P_{l_i,1}$, $l_i \in Lab_P(T) - \{l_j\}$, have to use $A \rightarrow A_{l_i}$, so that at least $k+3+k-1-1 \geq k+2$ non-terminal symbols are left in the current sentential form after two derivation steps, which again allows a third derivation step introducing $\#$.

ii. $l \notin Lab_P(T)$: The only difference to the previous case is that the components $P_{l_i,1}$, $l_i \in Lab_P(T)$, cannot generate A_l, i.e. similar arguments like those used above show that the derivation with the team T cannot terminate without introducing the trap symbol $\#$.

(b) $l \in Lab_e$. In this case, at most three derivation steps without introducing $\#$ are possible.
Like in the case with $l \in Lab_c$, if after the first derivation step at least one component $P_{l_i,1}$, $l_i \in Lab_P(T)$, can only use a trap rule, a further derivation step is possible, because at least $k+2$ non-terminal symbols are available in the current sentential form. Whereas the components $P_{l_i,1}$, $l_i \in Lab_P(T)$, in every step "produce" a non-terminal symbol A_{l_i}, $P_{l,2}$ can use $X_l \rightarrow b$, $A_l \rightarrow A'_l$ and $A'_l \rightarrow a$, and $Q_{l,2}$ uses its rules on the symbols from Δ. After two steps, a symbol from Δ is still occurring in the current sentential form, and $P_{l,2}$ can only have been responsible for the changing of X_l *or* of A_l to a terminal symbol.
In the third step, the symbol from Δ is eliminated by $Q_{l,2}$ and $P_{l,2}$ has the possibility to have eliminated X_l as well as one symbol A_l. Yet in three steps by the components $P_{l_i,1}$, $l_i \in Lab_P(T)$, $3k \geq k+2$ symbols A_{l_i} have been generated, which guarantees a fourth step with introducing $\#$ to be possible before the derivation with the team T can terminate.

The special case of a team $\{P_{l,1}, P_{l,2}, Q_{l,2}\}$ for some $l \in Lab_e$ also shows the necessity of delaying the generation of a from A_l by $P_{l,2}$, $l \in Lab_e$ (i.e. l being the label of a terminal matrix $l : (A \rightarrow a, X \rightarrow b)$), with two rules $A_l \rightarrow A'_l$ and $A'_l \rightarrow a$ instead of using only one single rule $A_l \rightarrow a$.

Case 3: Configuration xX_lE_l, for $l \in Lab_d$.
The only components that do not introduce $\#$ by the first rule they can apply are $P_{l',1}$ for any arbitrary $l' \in Lab_c \cup Lab_e$ (and therefore $l' \neq l$) as well as $P_{l,2}$ and $Q_{l,2}$. Hence only the following teams might be possible:

1. $T = \{P_{l_i,1} \mid 1 \leq i \leq k\}$, where $k \geq 3$ and $\{l_i \mid 1 \leq i \leq k\} \subseteq Lab_c \cup Lab_e$.
Each component $P_{l_i,1}$ introduces symbols A_{l_i}, until the non-terminal symbols for at least one component are exhausted, therefore finally one rule of the form $A_{l_i} \rightarrow \#$ has to be applied in at least one of the components of the team.
2. $T = \{P_{l,2}\} \cup \{P_{l_i,1} \mid 1 \leq i \leq k\}$, where $k \geq 2$ and $\{l_i \mid 1 \leq i \leq k\} \subseteq Lab_c \cup Lab_e$.
$P_{l,2}$ can only replace X_l by Y; in the meantime, the other components $P_{l_i,1}$, $1 \leq i \leq k$, must introduce symbols A_{l_i}. In the second derivation step, in $P_{l,2}$ the trap rule $Y \rightarrow \#$ can be used, whereas all the other components at least can apply $A_{l_i} \rightarrow \#$.
3. $T = \{Q_{l,2}\} \cup \{P_{l_i,1} \mid 1 \leq i \leq k\}$, where $k \geq 2$ and $\{l_i \mid 1 \leq i \leq k\} \subseteq Lab_c \cup Lab_e$.
While $E_l \rightarrow D$ is used in $Q_{l,2}$, the other components $P_{l_i,1}$, $1 \leq i \leq k$, introduce symbols A_{l_i}, but in the second derivation step $Q_{l,2}$ has to use a trap rule, e. g. $X_l \rightarrow \#$, while the other components $P_{l_i,1}$ at least can apply $A_{l_i} \rightarrow \#$.
4. $T = \{P_{l,2}, Q_{l,2}\} \cup \{P_{l_i,1} \mid 1 \leq i \leq k\}$, where $k \geq 1$ and $\{l_i \mid 1 \leq i \leq k\} \subseteq Lab_c \cup Lab_e$, i.e. T contains the legal team $\{P_{l,2}, Q_{l,2}\}$.
While $P_{l,2}$ uses $X_l \rightarrow Y$ and $Q_{l,2}$ uses $E_l \rightarrow D$, the other components $P_{l_i,1}$, $1 \leq i \leq k$, introduce symbols A_{l_i}. $Q_{l,2}$ now has to use a trap rule like $A_{l_{i_0}} \rightarrow \#$ for some $l_{i_0} \in \{l_i \mid 1 \leq i \leq k\}$, $P_{l,2}$ can use $D \rightarrow \#$, $P_{l_{i_0},1}$ can apply at least some rule on Y, and all the other components $P_{l_i,1}$, $l_i \in \{l_i \mid 1 \leq i \leq k\} - \{l_{i_0}\}$ can at least apply $A_{l_i,1} \rightarrow \#$.

In conclusion, again we have proved that only the legal teams can be used without introducing the trap symbol $\#$, which completes the proof. □

Lemma 5. $MAT_{ac} \subseteq T_sCD(t_0)$ and $MAT_{ac}^{\lambda} \subseteq T_sCD^{\lambda}(t_0)$ for every $s \geq 3$.

Proof. Let $L \subseteq V^*$ be a matrix language in MAT_{ac}^{λ} and let $G = (N', V, S, M, F)$ be a matrix grammar with $L(G) = L$. Again the matrix grammar G can be assumed to be in the strengthened accurrate normal form described in the preceeding two lemmas:

1. $N' = N_1 \cup N_2 \cup \{S, \#\}$, where $N_1, N_2, \{S, \#\}$ are pairwise disjoint.
2. The matrices in M are of one of the following forms:
   a. $(S \rightarrow w)$, $w \in V^*$;
   b. $(S \rightarrow AX)$, $A \in N_1$, $X \in N_2$;
   c. $(A \rightarrow w, X \rightarrow Y)$, $A \in N_1$, $w \in (N_1 \cup V)^*$, $|w|_{\{A\}} = 0$, $X, Y \in N_2$, $X \neq Y$;
   d. $(A \rightarrow \#, X \rightarrow Y)$, $A \in N_1$, $X, Y \in N_2$, $X \neq Y$;
   e. $(A \rightarrow a, X \rightarrow b)$, $A \in N_1$, $X \in N_2$, $a, b \in V \cup \{\lambda\}$.
3. The set F consists of all rules $A \rightarrow \#$ appearing in matrices of M.

We can construct a CD grammar system Γ such that $L_{t_0}(\Gamma, s) = L$ using the ideas already known from the preceeding proofs for the case $s = 2$, i.e. we add $s - 2$ additional control variables in every legal sentential form as well as $s - 2$ additional control components to every legal team.

Assume all matrices of the forms c, d, e in the sets M to be labelled in a one-to-one manner and let Lab_c, Lab_d, Lab_e, be the set of all the corresponding labels as well as

$$Lab = Lab_c \cup Lab_d \cup Lab_e.$$

Now consider the following sets of symbols

$$\begin{aligned}
\Pi &= \{A_l, A'_l \mid A \in N_1,\ l \in Lab\}, \\
\Sigma &= \{X_l \mid X \in N_2,\ l \in Lab\}, \\
\Delta &= \{D, D_l, E_l, F_l, G_l \mid l \in Lab\} \cup \\
&\quad \{H_k, H_{k,l,i} \mid 1 \leq k \leq s-2,\ l \in Lab,\ 1 \leq i \leq 4\}, \\
\Psi &= \Pi \cup \Sigma \cup \Delta, \text{ and} \\
N &= N_1 \cup N_2 \cup \Pi \cup \Sigma \cup \Delta.
\end{aligned}$$

We construct a CD grammar system Γ with $N \cup \{\#\}$ as the set of non-terminal symbols, V as the set of terminal symbols, the set of axioms

$$W = \{w \mid (S \to w) \in M,\ w \in V^*\} \cup \{AXDH_1...H_{s-2} \mid (S \to AX) \in M\}$$

and the components $P_{l,1}, Q_{l,1}, R_{1,l,1}, ..., R_{s-2,l,1}$, and $P_{l,2}, Q_{l,2}, R_{1,l,2}, ..., R_{s-2,l,2}$, for $l \in Lab$:

A. If $l : (A \to w, X \to Y)$ is a matrix of type c with $A \in N_1$, $w \in (N_1 \cup V)^*$, $|w|_{\{A\}} = 0$, and $X, Y \in N_2$, $X \neq Y$, then we take the components

$P_{l,1} = \{X \to X_l, A \to A_l\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,1} = \{D \to D_l, D_l \to E_l\} \cup$

$\{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \{A_l, X_l, E_l, H_{1,l,2}, ..., H_{s-2,l,2}\}\}$,

$R_{k,l,1} = \{H_k \to H_{k,l,1}, H_{k,l,1} \to H_{k,l,2}\} \cup \{\beta \to \# \mid \beta \in N\}$, $1 \leq k \leq s-2$,

$P_{l,2} = \{A_l \to w, X_l \to Y\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,2} = \{E_l \to F_l, F_l \to D\} \cup \{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \{D, Y, H_1, ..., H_{s-2}\}\}$,

$R_{k,l,2} = \{H_{k,l,2} \to H_{k,l,3}, H_{k,l,3} \to H_{k,l,4}\} \cup \{\beta \to \# \mid \beta \in N\}$, $1 \leq k \leq s-2$.

B. If $l : (A \to a, X \to b)$ is a matrix of type e, with $A \in N_1$, $X \in N_2$, $a, b \in V \cup \{\lambda\}$, then we take the components

$P_{l,1} = \{X \to X_l, A \to A_l\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,1} = \{D \to D_l, D_l \to E_l\} \cup$

$\{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \{A_l, X_l, E_l, H_{1,l,2}, ..., H_{s-2,l,2}\}\}$,

$R_{k,l,1} = \{H_k \to H_{k,l,1}, H_{k,l,1} \to H_{k,l,2}\} \cup \{\beta \to \# \mid \beta \in N\}$, $1 \leq k \leq s-2$,

$P_{l,2} = \{A_l \to A'_l, A'_l \to a, X_l \to b\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,2} = \{E_l \to F_l, F_l \to G_l, G_l \to \lambda\} \cup \{\beta \to \# \mid \beta \in N\}$,

$R_{k,l,2} = \{H_{k,l,2} \to H_{k,l,3}, H_{k,l,3} \to H_{k,l,4}, H_{k,l,4} \to \lambda\} \cup$

$\{\beta \to \# \mid \beta \in N\}$, $1 \leq k \leq s-2$.

C. If $l : (A \to \#, X \to Y)$ is a matrix of type d (hence with $A \to \# \in F$), with $A \in N_1$, $X, Y \in N_2$, $X \neq Y$, then we take the components

$P_{l,1} = \{X \to X_l\} \cup \{\beta \to \# \mid \beta \in N\}$,

$Q_{l,1} = \{D \to E_l\} \cup \{\beta \to \# \mid \beta \in (\Psi \cup N_2 \cup \{A\}) - \{E_l, X_l, H_{1,l,2}, ..., H_{s-2,l,2}\}\}$,
$R_{k,l,1} = \{H_k \to H_{k,l,2}\} \cup \{\beta \to \# \mid \beta \in N\}$, $1 \leq k \leq s-2$,
$P_{l,2} = \{X_l \to Y\} \cup \{\beta \to \# \mid \beta \in N\}$,
$Q_{l,2} = \{E_l \to D\} \cup \{\beta \to \# \mid \beta \in (\Psi \cup N_2) - \{D, Y, H_1, ..., H_{s-2}\}\}$,
$R_{k,l,2} = \{H_{k,l,2} \to H_k\} \cup \{\beta \to \# \mid \beta \in N\}$, $1 \leq k \leq s-2$.

The intended *legal teams* of two components again are

$$\{P_{l,1}, Q_{l,1}, R_{1,l,1}, ..., R_{s-2,l,1}\} \text{ as well as } \{P_{l,2}, Q_{l,2}, R_{1,l,2}, ..., R_{s-2,l,2},\}$$

for arbitrary labels $l \in Lab$, the *legal configurations* are

1. $xXDH_1...H_{s-2}$, with $x \in (N_1 \cup V)^+$, $X \in N_2$ (initially we have $x \in N_1$),
2. $xA_lx'X_lE_lH_{1,l,2}...H_{s-2,l,2}$, for $x, x' \in (N_1 \cup V)^*$, $A \in N_1$, $X \in N_2$, $l \in Lab_c \cup Lab_e$, and
3. $xX_lE_lH_{1,l,2}...H_{s-2,l,2}$, for $x \in (N_1 \cup V)^+$, $X \in N_2$, $l \in Lab_d$.

Again we obtain $L_{t_0} \subseteq L(\Gamma, s)$: If $z_1 \Longrightarrow_G z_2$ is a derivation step in G, where z_2 is not a terminal string, then $z_1DH_1...H_{s-2} \Longrightarrow_\Gamma^* z_2DH_1...H_{s-2}$ in a derivation sequence using appropriate teams of size s from Γ, and if z_2 is a terminal string, then $z_1DH_1...H_{s-2} \Longrightarrow_\Gamma^* z_2$ in a derivation sequence using the appropriate teams of size s from Γ.
As the additional components of type R contain the trap rules $\beta \to \#$ for every $\beta \in N$, these additional components will never be responsible for the termination of a derivation sequence with a team containing such components. Hence, similar arguments as in the previous proofs can be used to show that $L_{t_0}(\Gamma, s) \subseteq L$; thus again we obtain $L_{t_0}(\Gamma, s) = L$, which proves $MAT_{ac}^\lambda \subseteq T_sCD^\lambda(t_0)$.
If $L \subseteq V^*$ is a matrix language in MAT_{ac}, we have to split up L :

$$L = (L \cap \left(\bigcup_{0 \leq i \leq s-2} V^i \right)) \cup \bigcup_{c, c_1, ..., c_{s-2} \in V} \delta^r_{cc_1...c_{s-2}}(L)\{cc_1...c_{s-2}\}.$$

The family MAT_{ac} is closed under right derivation, hence $\delta^r_{cc_1...c_{s-2}}(L) \in MAT_{ac}$. For each of these languages $\delta^r_{cc_1...c_{s-2}}(L)$ we consider a matrix grammar $G_{cc_1...c_{s-2}}$ in the strengthened accurrate normal form in order to construct a CD grammar system Γ with $L_{t_0}(\Gamma, s) = L$ following the ideas described in the first part of this proof and of Lemma 3. The details of this construction for proving $MAT_{ac} \subseteq T_sCD(t_0)$ are obvious and therefore left to the interested reader. □

As it is quite obvious, the proofs of the preceeding lemmas cannot be used for obtaining the results proved in [Păun, Rozenberg 1994] for the derivation mode t_2, e.g. the components $P_{l,1}$ for $l \in Lab_c$ contain the rules $\beta \to \#$ for every $\beta \in N$, which means that $P_{l,1}$ still is applicable to every legal configuration even after the termination of a derivation sequence with the legal team $\{P_{l,1}, Q_{l,1}\}$. On the other hand, the CD grammar systems Γ in the proofs of Lemma 3, Lemma 4 and Lemma 5 were elaborated in such a way that they also work correctly in the dervation mode t_1, which not only allows a new proof of some of the results already obtained in [Păun, Rozenberg 1994] for the derivation mode t_1, but also yields an improvement of these results, because we now can allow teams of arbitrary size without the restriction for these teams to be of size at least two.

Corollary. For every $s \in \{*, +\} \cup \{2, 3, 4, ...\}$,

$$MAT_{ac} \subseteq T_sCD(t_1) \text{ and } MAT_{ac}^{\lambda} \subseteq T_sCD^{\lambda}(t_1).$$

Proof. As we have already pointed out in the previous section, $L_{t_1}(\Gamma, s) \subseteq L_{t_0}(\Gamma, s)$ for every $s \in \{*, +\} \cup \{2, 3, 4, ...\}$ and every CD grammar system Γ. Therefore this relation also holds true for the CD grammar systems Γ constructed in the previous proofs for the matrix languages in MAT_{ac} and MAT_{ac}^{λ}. Moreover, whenever $z_1 \Longrightarrow_T^{t_0} z_2$ with a legal team T from Γ, where z_1 is a legal configuration and z_2 is a legal configuration or a terminal string, then we also have $z_1 \Longrightarrow_T^{t_1} z_2$, because in any case from the component of type Q in the team T no rule can be applied any more when the derivation sequence started from z_1 terminates with z_2 according to the derivation mode t_0, which implies that the derivation sequence terminates in the derivation mode t_1, too. Therefore we conclude $L_{t_0}(\Gamma, s) \subseteq L_{t_1}(\Gamma, s)$, which all together implies $L_{t_1}(\Gamma, s) = L_{t_0}(\Gamma, s)$ and completes the proof of the corollary. □

Combining the main results obtained in this paper we get the following

Theorem. For every $s \in \{*, +\} \cup \{2, 3, 4, ...\}$ and $i \in \{0, 1\}$,

$$MAT_{ac} = PT_sCD(t_i) = T_sCD(t_i) \text{ and}$$

$$MAT_{ac}^{\lambda} = PT_sCD^{\lambda}(t_i) = T_sCD^{\lambda}(t_i).$$

Proof. For the derivation mode t_1 all the results stated in the theorem follow from the results already proved in [Păun, Rozenberg 1994] as well as from the corollary proved above.
For the derivation mode t_0 all the results stated in the theorem follow from the results proved in this section: From Lemma 1 we know that $PT_*CD^{\lambda}(t_0)$ (respectively $PT_*CD(t_0)$) is an upper bound for all the other families of languages generated by CD grammar systems (without λ-rules) with teams in the derivation mode t_0, and in Lemma 2 we have proved $PT_*CD^{\lambda}(t_0) \subseteq MAT_{ac}^{\lambda}$ (respectively $PT_*CD(t_0) \subseteq MAT_{ac}$). On the other hand, in Lemma 3, in Lemma 4 and in Lemma 5 we have proved that $MAT_{ac}^{\lambda} \subseteq T_sCD^{\lambda}(t_0)$ (and $MAT_{ac} \subseteq T_sCD(t_0)$) for every $s \in \{*, +\} \cup \{2, 3, 4, ...\}$, which all together proves the results stated in theorem. □

5 References

[Csuhaj-Varjù 1994] Csuhaj-Varjù, E.: "Cooperating Grammar Systems: Power and Parameters"; in: Results and Trends in Theoretical Computer Science (Karhumäki, J., Maurer, H., Rozenberg, G., eds.), LNCS, 812, 67 - 84; Springer-Verlag, Berlin (1994).

[Csuhaj-Varjù, Dassow 1990] Csuhaj-Varjù, E., Dassow, J.: "On Cooperating/Distributed Grammar Systems"; J. Inform. Process. Cybern. EIK 26 (1990), 49-63.

[Csuhaj-Varjù, Dassow, Kelemen, Păun 1994] Csuhaj-Varjù, E., Dassow, J., Kelemen, J., Păun, Gh.: "Grammar Systems"; Gordon and Breach, London (1994).

[Csuhaj-Varjù, Păun 1993] Csuhaj-Varjù, E., Păun, Gh.: "Limiting the Size of Teams in Cooperating Grammar Systems"; Bulletin of the EATCS, 51 (1993), 198-202.

[Dassow, Păun 1989] Dassow, J., Păun, Gh.: "Regulated Rewriting in Formal Language Theory"; Springer-Verlag, Berlin (1989).

[Kari, Mateescu, Păun, Salomaa 1994] Kari, L., Mateescu, A., Păun, Gh., Salomaa, A.: "Teams in Cooperating Grammar Systems"; J. Experimental and Theoretical AI, to appear.

[Meersman, Rozenberg 1978] Meersman, R., Rozenberg, G.: "Cooperating Grammar Systems"; Proceedings MFCS'78, LNCS, 64, Springer-Verlag, Berlin (1978), 364-374.

[Păun, Rozenberg 1994] Păun, Gh., Rozenberg, G.: "Prescribed Teams of Grammars"; Acta Informatica, to appear.

[Salomaa 1973] Salomaa, A.: "Formal Languages"; Academic Press, New York (1973).