Teams in Grammar Systems: Sub-Context-Free Cases

Maurice H. ter BEEK[1]

Department of Computer Science, Leiden University,
P.O. Box 9512, 2300 RA Leiden, The Netherlands
E-mail: `mtbeek@wi.leidenuniv.nl`

Abstract. The study of teams in grammar systems so far has evolved around teams being formed from a finite number of sets of *context-free* productions. Here, the generative power of teams in grammar systems consisting of regular, linear and metalinear sets of productions is investigated.

For these sub-context-free cases the forming of teams strictly increases the generative power of the underlying grammar systems in many cases.

1. Introduction

When an agent is unable to tackle a complex problem, due to limited capabilities, it seems natural to try to tackle the problem by more than one agent. This results in what they call in *Artificial Intelligence* (AI) a *multi-agent system.* An example of the idea of multi-agent systems in *Distributed AI* are is the so-called *blackboard model of problem solving.*

This model starts with a given problem specified on the blackboard. Several knowledge sources contribute, regulated by a certain strategy, to solving the problem by changing the current state of the blackboard. During the problem solving, the only way in which these knowledge sources can communicate with each other is by using the blackboard. Finally, in the case of successful cooperation, the solution appears on the blackboard.

The link between this blackboard model of problem solving and formal languages was established in [5]. The knowledge sources correspond to grammars, changing the current state of the blackboard corresponds to rewriting the sentential form, the strategy is regulated by so-called derivation modes and the solution is represented by a terminal word. In [3], *cooperating distributed grammar systems*, CD grammars systems for short, have been introduced as a formal realisation of this link. These systems have been investigated intensively. Moreover, they have initiated the development of the theory of grammar systems. This theory has already resulted in the monograph [4], which contains an exhaustive survey of the state of the art in the area until ca. 1992.

Already, several well-motivated enhancements of these CD grammar systems have been introduced, such as *hybrid* CD grammar systems ([15]), *team* CD grammar systems ([13]) and, most recently, *hybrid team* CD grammar systems ([2]). In hybrid CD grammar systems, a more realistic approach to cooperation is considered, by assuming the grammars to have different capabilities. In team CD grammar systems the

---

[1] This research was supported by a scholarship from the Hungarian Ministry of Culture and Education. Moreover, the facilities provided by the Department of General Computer Science of the Eötvös Loránd University and in particular by the Computer and Automation Research Institute of the Hungarian Academy of Sciences were essential.

natural idea of work being done in teams is incorporated in the system by grouping several grammars and use them to rewrite in parallel. The teams are either formed automatically or prescribed and several versions of the maximal competence strategy in CD grammar systems are defined. Hybrid team CD grammar systems combine these two ideas.

In [13] it was shown that there are situations in which the forming of teams enlarges the power of the underlying CD grammar system and that they form an AFL when working in the maximal competence strategy. Moreover, in [6] it was proved that teams of size two suffice. In [19] it was shown that there are situations in which exactly the power of programmed grammars is obtained (recursively enumerable when λ-productions are allowed) and in [11], this result was extended to cover more cases of teams in grammar systems. Another surprising result is that the different maximal competence strategies introduced in each of these papers lead to the same generative power. In [2], finally, it was proved that when hybrid teams are allowed the generative power is not enlarged any further. However, every recursively enumerable language can be generated by a hybrid prescribed team CD grammar system with teams of two members. Moreover, concerning syntactic complexity these systems could well be favoured.

Until now, only team CD grammar systems with context-free productions have been considered. Here, the case of a restriction to regular, linear and metalinear productions is studied. For (hybrid) prescribed team CD grammar systems with teams of constant size and regular productions, the team-forming enlarges their generative power beyond the class of regular languages to the class of regular simple matrix grammars. Hence it extends also beyond the power of regular (hybrid) CD grammar systems. The same holds in the case of a restriction to linear productions. These results lead to several corollaries, one of these being that the class generated by (hybrid) prescribed team CD grammar systems with a restriction to regular or linear productions and teams of constant size is incomparable with the class of context-free languages. On the other hand, (hybrid) (team) CD grammar systems with context-free productions include that class, whereas for the metalinear case, incomparability is only conjectured.

In the case of teams of variable size, no more than the class of regular or linear languages can be generated by (hybrid) prescribed team CD grammar systems with only regular or linear productions, respectively. However, when restricted to metalinear productions, the generative power of (hybrid) prescribed team CD grammar systems extends beyond the class of metalinear languages. Moreover, already the class generated by prescribed team CD grammar systems with this restriction to metalinear productions is equal to the class of programmed grammars with the same restriction and appearance checking in the case of the maximal competence strategies. For the other modes of derivation, the results hold only without appearance checking.

2. Preliminaries

In this section, some prerequisites necessary for understanding the sequel are defined. For details and unexplained notions, the reader is referred to [22] for formal languages, [9] for regulated rewriting, [21] for Lindenmayer systems and [4], [7], [8] and [17] and [2] for (variants of) grammar systems.

The set of all non-empty strings over an *alphabet* V is denoted by V^+. If the *empty* *string*, λ, is included, the notation becomes V^*. The *length* of a string x is denoted by $|x|$.

An *inclusion* is denoted by $\subseteq$, whereas a *proper inclusion* is denoted by $\subset$.

Sometimes, the notation for a family of languages contains a λ between the brackets [and]. This means that the statement holds in the case of allowing λ-productions (indicated by the λ inbetween brackets) as well as in the case of a restriction to λ-free productions (thus neglecting the λ inbetween brackets). Also other symbols between brackets must now be understood.

Without definition, the family of regular (REG), linear (LIN), metalinear ($MLIN$), context-free (CF) and context-sensitive (CS) languages are used in the sequel. Their definitions can be found in, e.g., [9]. The same holds for the family of languages generated by ET0L systems ($ET0L$). Finally, also the family of languages generated by [hybrid] CD grammar systems ($[H]CD$) shall not be defined here. However, their definitions can be found in [4] and will become clear in the sequel.

None of the above families of languages will be used in any construction in the proofs. Those families of languages that are used in (some of) the proofs below, are defined next.

An *unordered scattered context grammar with appearance checking* ([14]) is a construct $G = (N, T, S, P, F)$, where N is the set of nonterminals, T is the set of terminals, $S \in N$ is the axiom, $P = \{p_1, p_2, \ldots, p_n\}$ is a finite set of *rules* (rules are of the form $p_i : (\alpha_1, \alpha_2, \ldots, \alpha_{m_i}) \to (\beta_1, \beta_2, \ldots, \beta_{m_i})$, where $\alpha_j \to \beta_j$ are productions over $N \cup T$) and F is a set of occurrences of productions in P, $1 \le i \le n$. For $w, w' \in (N \cup T)^*$ and $1 \le i \le n$ it is said that w directly derives w', written as

$$\begin{array}{lll} w \Longrightarrow w' & \text{iff} & w = w_1\alpha_{i_1}w_2\alpha_{i_2}\ldots w_m\alpha_{i_m}w_{m+1},\ w' = w_1\beta_{i_1}w_2\beta_{i_2}\ldots w_m\beta_{i_m}w_{m+1}, \\ & & p_i : (\alpha_1, \alpha_2, \ldots, \alpha_p) \to (\beta_1, \beta_2, \ldots, \beta_p) \in P,\ (\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_m}) \text{ is a} \\ & & \text{permutation of a subsequence of } (\alpha_1, \alpha_2, \ldots, \alpha_p),\ w_l \in (N \cup T)^* \\ & & \text{and } 1 \le l \le m+1 \\ & \text{and} & \alpha_j \text{ in } \{\alpha_1, \alpha_2, \ldots, \alpha_p\} \text{ and not in } \{\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_m}\} \text{ implies that} \\ & & \alpha_j \text{ is not contained in } w \text{ and } \alpha_j \to \beta_j \in F. \end{array}$$

If $F = \emptyset$, the unordered scattered context grammar is called an *unordered scattered context grammar without appearance checking* and F is omitted from the construct. Moreover, if F contains all occurrences of productions in P, the unordered scattered context grammar is called *with unconditional transfer.* The language generated by G is $L(G) = \{w \in T^* \mid S \Longrightarrow^* w\}$, where $\Longrightarrow^*$ denotes the reflexive and transitive closure of $\Longrightarrow$.

The family of languages generated by unordered scattered context grammars with λ-free context-free productions in P is denoted by USC_{ac} in the case of grammars with appearance checking; when grammars without appearance checking are considered the subscript ac is omitted and when grammars with unconditional transfer are considered the subscript ac is replaced by ut.

A *programmed grammar* ([20]) is a construct $G = (N, T, S, P)$, where N is the set of nonterminals, T is the set of terminals, $S \in N$ is the axiom and P is a finite set of productions of the form $(r : \alpha \to \beta, \sigma(r), \varphi(r))$, where $r : \alpha \to \beta$ is a production over $N \cup T$, labelled by r. Denote by $Lab(P) = \{r \mid (r : \alpha \to \beta, \sigma(r), \varphi(r)) \in P\}$ the set of labels of productions of G. Then $\sigma(r) \subseteq Lab(P)$ is called the *success field* of production r and $\varphi(r) \subseteq Lab(P)$ the *failure field*. For $(r_1 : \alpha \to \beta, \sigma(r_1), \varphi(r_1)) \in P$ and $w, w' \in (N \cup T)^*$ it is said that w directly derives w', written as

$$\begin{array}{lll} (w, r_1) \Longrightarrow (w', r_2) & \text{iff} & w = w_1 \alpha w_2,\ w' = w_1 \beta w_2 \text{ and } r_2 \in \sigma(r_1) \\ & \text{or} & w = w',\ \alpha \to \beta \text{ cannot be applied to } w \text{ and } r_2 \in \varphi(r_1). \end{array}$$

If the failure fields are empty for every production, the programmed grammar is called *without appearance checking*; otherwise it is called *with appearance checking*. Moreover, if the success field and the failure field coincide for every labeled production, the programmed grammar is called *with unconditional transfer*. The language generated by G is $L(G) = \{w \in T^* \mid (S, r_0) \Longrightarrow^* (w, r_1),\ r_0, r_1 \in Lab(P)\}$, where $\Longrightarrow^*$ denotes the reflexive and transitive closure of $\Longrightarrow$.

The family of languages generated by programmed grammars with λ-free context-free productions in P is denoted by PR_{ac} in the case of grammars with appearance checking; when grammars without appearance checking are considered the subscript ac is omitted and when grammars with unconditional transfer are considered the subscript ac is replaced by ut.

A *matrix grammar with appearance checking* is a construct $G = (N, T, S, M, F)$, where N is the set of nonterminals, T is the set of terminals, $S \in N$ is the axiom, M is a finite set of *matrices* of the form $m : (r_1, r_2, \ldots, r_n)$, where $r_i : \alpha_i \to \beta_i$ are productions over $N \cup T$ and $|\alpha|_N \geq 1$, $1 \leq i \leq n$ and F, finally, is a set of occurrences of productions in M. For $w, w' \in (N \cup T)^*$ and $m : (\alpha_1 \to \beta_1, \alpha_2 \to \beta_2, \ldots, \alpha_n \to \beta_n) \in M$ it is said that w directly derives w', written as

$$\begin{array}{lll} w \Longrightarrow w' & \text{iff} & \text{there exist } w_0, w_1, \ldots, w_n \in (N \cup T)^* \text{ such that} \\ & & w_0 = w \text{ and } w_n = w' \text{ and for all } 0 \leq i \leq n-1 \\ & \text{either} & w_{i-1} = w'_{i-1} \alpha_i w''_{i-1} \text{ and } w_i = w'_{i-1} \beta_i w''_{i-1} \\ & & \text{for some } w'_{i-1}, w''_{i-1} \in (N \cup T)^* \\ & \text{or} & \text{the production } \alpha_i \to \beta_i \text{ cannot be applied to } w_{i-1}, \\ & & \alpha_i \to \beta_i \in F \text{ and } w_i = w_{i-1}. \end{array}$$

If $F = \emptyset$, the matrix grammar is called a *matrix grammar without appearance checking* and F is omitted from the construct. Moreover, if F contains all occurrences of productions in M, the matrix grammar is called *with unconditional transfer*. The language generated by G is $L(G) = \{w \in T^* \mid S \Longrightarrow^* w\}$, where $\Longrightarrow^*$ denotes the reflexive and transitive closure of $\Longrightarrow$.

The family of languages generated by matrix grammars with λ-free context-free productions in M is denoted by MAT_{ac} in the case of grammars with appearance checking; when grammars without appearance checking are considered the subscript ac is omitted and when grammars with unconditional transfer are considered the subscript ac is replaced by ut.

A *simple matrix grammar* ([12]) of *degree* n, $n \geq 1$, is a construct $G = (N_1, N_2, \ldots, N_n, T, S, M)$, where $N_1, N_2, \ldots, N_n$ (sets of nonterminals) and T (the set of terminals) are pairwise disjoint alphabets, $S \notin (\bigcup_{i=1}^{n} N_i \cup T)$ is the start symbol and M is a finite set of matrices, each of one of the following forms.

(a) $(S \to x)$, for $x \in T^*$,

(b) $(S \to A_1 A_2 \dots A_n)$, for $A_i \in N_i$ and $1 \le i \le n$ or

(c) $(A_1 \to x_1, A_2 \to x_2, \dots, A_n \to x_n)$, for $A_i \in N_i$, $x_i \in (N_i \cup T)^*$ and $|x_i|_{N_i} = |x_j|_{N_j}$ for all $1 \le i, j \le n$.

For $w, w' \in (\bigcup_{i=1}^{n} N_i \cup T \cup \{S\})^*$ it is said that w directly derives w', written as

$$\begin{array}{lll} w \Longrightarrow w' & \text{iff} & w = S \text{ and } (S \to w') \in M \\ & \text{or} & w = v_1 A_1 w_1 v_2 A_2 w_2 \dots v_n A_n w_n,\ w' = v_1 x_1 w_1 v_2 x_2 w_2 \dots v_n x_n w_n, \\ & & A_i \in N_i,\ v_i \in T^*,\ w_i, x_i \in (N_i \cup T)^*,\ 1 \le i \le n \text{ and} \\ & & (A_1 \to x_1, A_2 \to x_2, \dots, A_n \to x_n) \in M. \end{array}$$

The language generated by G is $L(G) = \{w \in T^* \mid S \Longrightarrow^* w\}$, where $\Longrightarrow^*$ denotes the reflexive and transitive closure of $\Longrightarrow$.

A simple matrix grammar is called regular, linear, context-free or λ-free iff the productions appearing in matrices of type (c) in M are all regular, linear, context-free or λ-free, respectively. The family of languages generated by λ-free context-free simple matrix grammars of degree n, $n \ge 1$, is denoted by $SM(n)$. Furthermore, denote $SM = \bigcup_{n \ge 1} SM(n)$ and likewise for the other cases.

For all generative devices mentioned above, only the notation in the case of λ-free context-free productions was given. However, when the productions are of type X, for $X \in \{REG, LIN, MLIN\}$, a subscript X is added to the notation. Moreover, when there is no restriction to λ-free productions a superscript λ is added to the notation.

3. Hybrid Prescribed Teams of Grammars

Definition 1. Let N and T be two disjoint alphabets. A *production* over (N, T) is a pair $(A, x) \in N \times (N \cup T)^*$. Usually, $A \to x$ shall be written instead of (A, x). If $x \ne \lambda$, then $A \to x$ is called a λ-free production. A team over (N, T) is a multiset of sets of productions over (N, T). The sets of productions occurring in a team shall be referred to as *components*.

A team rewrites a string in the following manner.

Definition 2. Let N and T be two disjoint alphabets. Let Q be a team over (N, T) and $x, y \in (N \cup T)^*$. Then x is rewritten by Q into y, written as

$$\begin{array}{lll} x \Longrightarrow_Q y & \text{iff} & x = x_1 A_1 x_2 A_2 \dots x_n A_n x_{n+1},\ y = x_1 y_1 x_2 y_2 \dots x_n y_n x_{n+1}, \\ & & x_i \in (N \cup T)^*,\ 1 \le i \le n+1,\ A_j \to y_j \in P_j,\ 1 \le j \le n \text{ and} \\ & & Q = \{P_1, P_2, \dots, P_n\}. \end{array}$$

A derivation step of a team thus consists of choosing a production from each component of this team and apply these in parallel on the string to be rewritten. If Q is a singleton team, i.e. $Q = \{P\}$ for some set of productions P, then $x \Longrightarrow_P y$ shall be written instead of $x \Longrightarrow_{\{P\}} y$. It is clear that in that case only one symbol in x is rewritten, using a production from P.

So-called modes of derivation are used to prescribe halting requirements on the use of a team. These modes can be divided into three groups. Firstly, mode $*$ has *no restrictions* whatsoever. Any number of derivation steps is allowed. Secondly, modes $\le k$, $= k$ and $\ge k$ restrict the number of derivation steps to *at most*, *exactly* and *at least* k derivation steps, respectively. Thirdly, modes t_0, t_1 f derivation steps. All three prescribe a slightly different condition which needs to be fulfilled before a team is considered to have successfully worked in that mode. In the case of mode t_0 the work of a team ends successfully when *no further derivation step can be done as a team*, in the case of mode t_1 the work ends when *no component of the team can apply one of its productions any longer* and in mode t_2, finally, the work of a team ends when *there is at least one component that can no longer apply one of its productions.*

Definition 3. Let $Q = \{P_1, P_2, \ldots, P_n\}$ be a team over (N, T) and let $f \in \{\leq k, = k, \geq k \mid k \geq 1\} \cup \{*, t_0, t_1, t_2\}$ be a *mode (of derivation).* Furthermore, let $x, y, z \in (N \cup T)^*$ and $k \in \mathbf{N}$. Then x is rewritten by Q, working in mode f, into y, written as

$$
\begin{array}{lcll}
x \Longrightarrow_Q^{\leq k} y & \text{iff} & x \Longrightarrow_Q^{k'} y \text{ for some } k' \leq k, \\
x \Longrightarrow_Q^{= k} y & \text{iff} & x \Longrightarrow_Q^{k} y, \\
x \Longrightarrow_Q^{\geq k} y & \text{iff} & x \Longrightarrow_Q^{k'} y \text{ for some } k' \geq k, \\
x \Longrightarrow_Q^{*} y & \text{iff} & x \Longrightarrow_Q^{k} y \text{ for some } k, \\
x \Longrightarrow_Q^{t_0} y & \text{iff} & x \Longrightarrow_Q^{*} y \text{ and there is no } z \text{ such that } y \Longrightarrow_Q z, \\
x \Longrightarrow_Q^{t_1} y & \text{iff} & x \Longrightarrow_Q^{*} y \text{ and for no component } P_i \in Q \text{ and no } z \\
 & & \text{there is a derivation } y \Longrightarrow_{P_i} z \text{ and} \\
x \Longrightarrow_Q^{t_2} y & \text{iff} & x \Longrightarrow_Q^{*} y \text{ and there is a component } P_i \in Q \\
 & & \text{for which there is no derivation } y \Longrightarrow_{P_i} z.
\end{array}
$$

The three variants of the t-mode of derivation first appeared in [11] (t_0), [13] (t_1) and [19] (t_2); the other modes of derivation are the natural extension of the modes in CD grammar systems (see [4]) to teams of grammars.

Now the definition of hybrid prescribed teams in the theory of grammar systems from [2] can be introduced.

Definition 4. A *hybrid prescribed team CD grammar system* is a construct

$$\Gamma = (N, T, S, P_1, P_2, \ldots, P_n, (Q_1, f_1), (Q_2, f_2), \ldots, (Q_m, f_m)),$$

where N is the set of nonterminals, T is the set of terminals, with $N \cap T = \emptyset$, $S \in N$ is the axiom, $P_1, P_2, \ldots, P_n$ are sets of productions over (N, T), $Q_1, Q_2, \ldots, Q_m$ are teams with components from $P_1, P_2, \ldots, P_n$ and $f_1, f_2, \ldots, f_m$ are modes of derivation.

This definition is more general than those from [13] and [19]. If, in this construct, $f_i = f_j$ for all $1 \leq i, j \leq m$, the definition of a prescribed team CD grammar system as in [19] is obtained.

Note that in this definition, there is no restriction on the size of a team. In the original definition of teams in [13], however, they are of constant size. A natural number $s \geq 1$ is given and the teams are formed such that the number of components of every team is exactly s; these teams are called of constant size s. Moreover, in that definition the teams are not prescribed, but each set of components can be a team (so-called *free* teams) as long as the size restriction is fulfilled.

It is now clear that one can differentiate between the following four variants of teams in the theory of grammar systems. For all four, hybridity is another possibility.

Free teams of constant size: this is the original definition of [13], as explained above.

Free teams of variable size: each subset of components can be a team.

Prescribed teams of constant size: all prescribed teams consist of the same number of components.

Prescribed teams of variable size: these are defined in Definition 4.

In the case of teams of constant size, whether prescribed or free, a finite set of axioms $W \subseteq (N \cup T)^*$, with only one string in it containing nonterminals, is allowed. This is done since otherwise in the case of λ-free productions no string shorter than s could be generated. In the case of free teams with teams of constant size, the construct thus becomes $\Gamma = (N, T, W, P_1, P_2, \ldots, P_n)$. The modifications in the other cases are obvious.

Definition 5. Consider a hybrid prescribed team CD grammar system Γ as in Definition 4. Then the language generated by Γ is

$$L(\Gamma) = \{z \in T^* \mid S \Longrightarrow_{Q_{i_1}}^{f_{i_1}} w_{i_1} \Longrightarrow_{Q_{i_2}}^{f_{i_2}} \cdots \Longrightarrow_{Q_{i_p}}^{f_{i_p}} w_{i_p} = z,\ 1 \leq i_j \leq m,\ 1 \leq j \leq p\}.$$

When dealing with a language generated by teams of constant size, the notation of Definition 5 is modified to $L(\Gamma, s)$. When the teams are not hybrid, the mode of derivation is added as a subscript to this notation.

The family of languages generated by CD grammar systems with hybrid prescribed teams of variable size and λ-free productions of type X is denoted by HPT_*CD_X. When teams are of constant size s, the $*$ in the notation is replaced by s and when there is no restriction to λ-free productions, λ is added to the notation as a superscript. When dealing with context-free productions this need not be specified and the subscript is thus omitted. Finally, when the teams are not hybrid (prescribed) the H (P) in the notation is omitted.

Some relations concerning the generative power of several of these grammar systems discussed above are given next. A more complete overview can be found in [1]. In the first paper on teams in grammar systems, [13], it was proved that, for $f \in \{=1, \geq 1, *\} \cup \{\leq k \mid k \geq 1\}$,

$$CF = T_1CD(f) \subset T_2CD(f) \text{ and}$$
$$ET0L = T_1CD(t) \subset T_2CD(t_1).$$

These relations prove that there are modes of derivation for which the forming of teams strictly increases the power of CD grammar systems, since $CD(t) = ET0L$ and $CF = CD(=1) = CD(\geq 1) = CD(*) = CD(\leq k)$ for a $k \geq 1$ were already known to hold (see, e.g., [4]). In [6] it was proved that teams of size two suffice, i.e. for $s \geq 2$

$$T_sCD(t_1) \subseteq T_2CD(t_1).$$

The main results of [19] are, for $s \geq 2$, $f \in \{*\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ and $g \in \{t_1, t_2\}$,

$$PR^{[\lambda]} = PT_sCD^{[\lambda]}(f) = PT_*CD^{[\lambda]}(f) \text{ and}$$
$$PR_{ac}^{[\lambda]} = T_sCD^{[\lambda]}(g) = PT_sCD^{[\lambda]}(g) = PT_*CD^{[\lambda]}(g)$$

and the main result of [11] is, for $s \geq 2$ and $h \in \{t_0, t_1\}$,

$$MAT_{ac}^{[\lambda]} = T_sCD^{[\lambda]}(h) = PT_sCD^{[\lambda]}(h) = PT_*CD^{[\lambda]}(h) = T_*CD^{[\lambda]}(h).$$

In [2] it was proved that $HPT_*CD \subseteq MAT_{ac}$ which, together with the results stated above, leads to the following relations for hybrid teams. For $s \geq 2$

$$PR_{ac}^{[\lambda]} = HPT_sCD^{[\lambda]} = HPT_*CD^{[\lambda]}.$$

4. The Sub-Context-Free Cases

In the previous section, results concerning (hybrid) (prescribed) team CD grammar systems with context-free productions were presented. In this section some results concerning a restriction to regular, linear or metalinear types of productions will be presented.

Recall the fact that, whether free or prescribed, teams with constant size are allowed to have a string axiom, whereas teams of variable size always have a single start symbol.

4.1. The Regular and the Linear Cases

First, a result for the regular case of prescribed team CD grammar systems with constant team-size 1 is presented.

Lemma 1. *For* $f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ *and* $g \in \{= k, \geq k \mid k \geq 2\} \cup \{t_0, t_1, t_2\}$

$$REG = CD_{REG}(f) \subset PT_1CD_{REG}(g).$$

Proof. The equality is proved in [4] and it is obvious that $CD_{REG}(f) \subseteq PT_1CD_{REG}(g)$ for $f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ and $g \in \{*, t_0, t_1, t_2\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$. Furthermore, the prescribed team CD grammar system, with teams of constant size 1,

$$\Gamma_1 = (\{A_0, A_0', A_1, A_1', \ldots, A_{k-2}', B, B'\}, \{a, b\}, AB, P_1, P_2, P_3, \{P_1\}, \{P_2\}, \{P_3\}),$$

where

$$\begin{aligned} P_1 &= \{A_0 \to A_1, A_1 \to A_2, \ldots, A_{k-2} \to aA_0', B \to bB'\}, \\ P_2 &= \{A_0' \to A_1', A_1' \to A_2', \ldots, A_{k-2}' \to A_0, B' \to B\} \text{ and} \\ P_3 &= \{A \to A_1, A_1 \to A_2, \ldots, A_{k-2} \to a, B \to b\}. \end{aligned}$$

contains only regular productions and it generates $L_f(\Gamma_1, 1) = \{a^n b^n \mid n \geq 1\} \in PT_1CD_{REG}(f) \setminus REG$ for $f \in \{= k, \geq k \mid k \geq 2\} \cup \{t_0, t_1, t_2\}$. □

Hence already a prescribed team CD grammar system with only regular productions and teams of size 1 can generate more than the class of regular languages and more than a CD grammar system with only regular productions. The next lemma states that also in the linear case the prescribed team CD grammar systems with teams of any constant size can generate more than the class of linear languages as well as more than CD grammar systems with only linear productions.

Lemma 2. *For* $f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$, $g \in \{= k, \geq k \mid k \geq 2\} \cup \{t_0, t_1, t_2\}$ *and* $g' \in \{*, t_0, t_1, t_2\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$

$$LIN = CD_{LIN}(f) \subset PT_1CD_{LIN}(g) \subseteq PT_1CD(g') = CD(f).$$

Proof. The first equality can be proved with a similar proof as for the regular case (see Lemma 1) and $CD_{LIN}(f) \subseteq PT_1CD_{LIN}(g)$ is obvious, for $f \in \{*, t\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ and $g \in \{*, t_0, t_1, t_2\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$. Furthermore, the prescribed team CD grammar system, with teams of constant size 1,

$$\Gamma_2 = (\{A_0, A_0', A_1, A_1', \ldots, A_{k-2}', B, B'\}, \{a, b, c\}, AB, P_1, P_2, P_3, \{P_1\}, \{P_2\}, \{P_3\}),$$

where

$$\begin{aligned} P_1 &= \{A_0 \to A_1, A_1 \to A_2, \ldots, A_{k-2} \to aA_0'b, B \to cB'\}, \\ P_2 &= \{A_0' \to A_1', A_1' \to A_2', \ldots, A_{k-2}' \to A_0, B' \to B\} \text{ and} \\ P_3 &= \{A \to A_1, A_1 \to A_2, \ldots, A_{k-2} \to ab, B \to c\}. \end{aligned}$$

contains only linear productions and it generates $L_f(\Gamma_2, 1) = \{a^n b^n c^n \mid n \geq 1\} \in PT_1CD_{LIN}(f) \setminus LIN$ for $f \in \{= k, \geq k \mid k \geq 2\} \cup \{t_0, t_1, t_2\}$. The last inclusion in the statement of the lemma is obvious and to prove the last equality, only the inclusion $PT_1CD(g') \subseteq CD(f)$ is not obvious. To prove this inclusion, all teams of size 1 become a component of the CD grammar system and a component $\{S \to S, S \to w \mid w \in W\}$, S being the axiom of the CD grammar system and W being the finite set of string axioms of the prescribed team CD grammar system with teams of constant size, is added. The mode of derivation remains the same, except that for t_0, t_1 and t_2 it becomes t. □

These two results lead to the following corollary for hybrid prescribed team CD grammar systems with teams of constant size 1 and only regular or linear productions.

Corollary 1.

$$\begin{aligned} REG = HCD_{REG} \subset HPT_1CD_{REG} \subseteq HPT_1CD_{LIN} \text{ and} \\ LIN = HCD_{LIN} \subset HPT_1CD_{LIN} \subseteq HPT_1CD = HCD. \end{aligned}$$

Proof. The equality $REG = HCD_{REG}$ is proved in [15], a similar proof can prove this equality for the linear case. The inclusions of hybrid CD grammar systems with only regular or linear productions in hybrid prescribed team CD grammar systems with teams of constant size 1 and only regular or linear productions, respectively, are obvious. Moreover, Lemma 1 and 2 prove their properness. The remaining two inclusions are also obvious and the last equality can be proved with a similar construction as for the proof of $PT_1CD(g') = CD(f)$ in Lemma 2. □

Hence also hybrid prescribed team CD grammar systems with only regular (linear) productions and teams of size 1 can generate more than the class of regular (linear) languages as well as more than hybrid CD grammar systems with only regular (linear) productions can.

In fairness, it must be noted that all proper inclusions proved in this section so far are due to the existence of a "non-linear" axiom, not to the very use of teams.

In fact, the following holds. To be more precise, denote $[H]P_nT_1CD_X(f)$ for the class of [hybrid] prescribed team CD grammar systems with at most n occurrences of nonterminals in the string axiom, teams of size 1, only components of type X and working in mode f (omitted in the hybrid case).

Theorem 1. *For* $n \geq 1$, $X \in \{REG, LIN\}$ *and* $f \in \{t_0, t_1, t_2\} \cup \{=k, \geq k \mid k \geq 2\}$

$$HP_nT_1CD_X = P_nT_1CD_X(f) = SM_X(n) =$$
$$SM_X^\lambda(n) = P_nT_1CD_X^\lambda(f) = HP_nT_1CD_X^\lambda.$$

Proof. In [4], so-called extended CD' grammar systems are defined. In the terminology of this paper, these systems are CD grammar systems with a string axiom. In [10] these extended CD' grammar systems with only regular productions, at most n nonterminals in the string axiom and working in mode $f \in \{t\} \cup \{=k, \geq k \mid k \geq 2\}$ $(E_nCD'_{REG}(f))$ are proved to be equal to the regular simple matrix grammars of degree n.

Clearly, $E_nCD'_{REG}(f) = P_nT_1CD_{REG}(f)$ for $f \in \{*\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ and $E_nCD'_{REG}(t) = P_nT_1CD_{REG}(g)$ for $g \in \{t_0, t_1, t_2\}$. When observing the proof, it can be seen that it holds for the linear case as well. Moreover, the construction can easily be modified to hold for the hybrid case as well. (One just has to code all nonterminals, thus indicating which mode is currently being simulated.) Finally, $SM_X^\lambda(n) = SM_X(n)$ for $X \in \{REG, LIN\}$ was proved in [16] and the proof thus holds for both the case of forbidding and the case of allowing λ-productions. □

This theorem has some interesting corollaries, since the families of regular and linear simple matrix grammars are well-investigated. A survey of simple matrix grammars can be found in [9], where the proofs of the results corresponding to the coming corollaries can be found.

Corollary 2. *For* $n \geq 1$ *and* $f \in \{=k, \geq k \mid k \geq 2\} \cup \{t_0, t_1, t_2\}$

$$P_nT_1CD_{REG}(f) = HP_nT_1CD_{REG} \subset P_nT_1CD_{LIN}(f) = HP_nT_1CD_{LIN}.$$

Corollary 3. *The number of nonterminal occurrences in the axioms of prescribed team CD grammar systems, with teams of size* 1 *and only regular or only linear productions, defines an infinite hierarchy of languages generated in all modes* $f \in \{=k, \geq k \mid k \geq 2\} \cup \{t_0, t_1, t_2\}$. *The same holds for the hybrid versions of these families of languages.*

Corollary 4. *For* $s \geq 1$ *and* $f \in \{=k, \geq k \mid k \geq 2\} \cup \{t_0, t_1, t_2\}$

$[H]PT_sCD_{REG}(f)$ *is incomparable with* LIN *and*
$[H]PT_sCD_{LIN}(f)$ *is incomparable with* CF.

The question is now what can be said about the generative power of (hybrid) prescribed team CD grammar systems with only regular or linear productions and teams of constant size s, for $s \geq 2$. From the results presented in Section 3, a comparison with the programmed (or matrix) grammars with only regular or linear productions seems natural. These, however, are equal to the classes of regular and linear languages, respectively, even when appearance checking is used. The proofs for these equalities in the regular case can be found in [9] and the proofs for the linear case can be proved similarly. Hence the inclusions $PR_{REG} = PR_{REG,ac} \subset PT_1CD_{REG}(f)$ and $PR_{LIN} = PR_{LIN,ac} \subset PT_1CD_{REG}(f)$, for $f \in \{=k, \geq k \mid k \geq 2\} \cup \{t_0, t_1, t_2\}$, are obvious.

Moreover, the equalities between the programmed grammars and matrix grammars hold in the regular and linear case as well, even with appearance checking. Again, proofs of these equalities can be found in [9]. Note that these proper inclusions hold already for teams of constant size 1. Thus to find a good comparison for the generative power of (hybrid) prescribed team CD grammar systems with only regular or linear productions and teams of constant size s, $s \geq 2$, remains an open problem.

After these results for (hybrid) prescribed team CD grammar systems with teams of constant size, some results for the case of teams of variable size are presented next. The difference is the use of a single nonterminal as axiom in the case of teams of variable size, whereas in the case of teams with constant size a finite set of string axioms, with only one of them containing nonterminals, is used.

Lemma 3. *For* $f \in \{*, t_0, t_1, t_2\} \cup \{\leq k, =k, \geq k \mid k \geq 1\}$

$$HPT_*CD_{REG} = PT_*CD_{REG}(f) = REG \text{ and}$$
$$HPT_*CD_{LIN} = PT_*CD_{LIN}(f) = LIN.$$

Proof. For teams with more than one component at least two nonterminals must be present in a sentential form, in order to use that team to rewrite that sentential form. This is in contradiction with the facts that every $\Gamma \in \{HPT_*CD_{REG}, PT_*CD_{REG}(f), HPT_*CD_{LIN}, PT_*CD_{LIN}(f) \mid f \in \{*, t_0, t_1, t_2\} \cup \{\leq k, =k, \geq k \mid k \geq 1\}\}$ has a single nonterminal as axiom and regular or linear productions, respectively. For teams of size one, the equality with (hybrid) CD grammar systems is obvious for the regular case as well as for the linear case, keeping in mind the use of a single nonterminal as axiom. From [4] ([15]) it is known that (hybrid) CD grammar systems with only regular productions do not generate more than the class of regular languages and similar proofs can be used to prove these results for the linear case as well. □

4.2. The Metalinear Case

It is obvious that the (hybrid) prescribed team CD grammar systems with teams of variable size and metalinear productions are able to generate languages beyond the class of regular or linear languages. What's more, the following lemma holds.

Lemma 4. *For* $f \in \{*, t\} \cup \{\leq k, =k, \geq k \mid k \geq 1\}$ *and* $g \in \{=1, \geq 1, *, t_0, t_1, t_2\} \cup \{\leq k \mid k \geq 1\}\}$

$$MLIN = CD_{MLIN}(f) = HCD_{MLIN} \subset PT_*CD_{MLIN}(g).$$

Proof. The first two equalities can be proved by the proofs of $REG = CD_{REG}(f)$ ([4]) and $REG = HCD_{REG}$ ([15]), with the obvious modifications. Furthermore, it is clear that $MLIN \subseteq PT_*CD_{MLIN}(f)$, for $f \in \{*, t_0, t_1, t_2\} \cup \{\leq k, =k, \geq k \mid k \geq 1\}\}$. Moreover, the prescribed team CD grammar system

$$\Gamma_3 = (\{S, A, B, C\}, \{a, b, c\}, S, P_1, P_2, \ldots, P_7, \{P_1\}, \{P_2, P_3, P_4\}, \{P_5, P_6, P_7\}),$$

with teams of variable size and the metalinear productions

$$
\begin{array}{lll}
P_1 = \{S \to ABC\}, & P_2 = \{A \to aA\}, & P_5 = \{A \to a\}, \\
 & P_3 = \{B \to bB\}, & P_6 = \{B \to b\}, \\
 & P_4 = \{C \to cC\} \text{ and } & P_7 = \{C \to c\},
\end{array}
$$

generates $L(\Gamma_3) = \{a^n b^n c^n \mid n \geq 1\} \in PT_*CD_{MLIN}(g) \setminus CF$, for $g \in \{=1, \geq 1, *, t_0, t_1, t_2\} \cup \{\leq k \mid k \geq 1\}\}$. Since it is known that $\{a^n b^n c^n \mid n \geq 1\} \in CS \setminus CF$ and from the Chomsky hierarchy that $MLIN \subset CF \subset CS$, the inclusion is proper indeed. □

Hence even languages beyond the class of metalinear languages can be generated already by a prescribed team CD grammar system with teams of variable size and metalinear productions for some modes of derivation. For prescribed team CD grammar systems with teams of constant size, no version containing only metalinear productions is defined due to the string axiom they already possess. Note, however, that the lemma above does not cover all modes of derivation, which Lemma 7 below will.

The following theorem is obtained by combining the results of the previous section and the results obtained so far in this section.

Theorem 2. *For* $f \in \{*, t_0, t_1, t_2\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ *and* $g \in \{=1, \geq 1, *, t_0, t_1, t_2\} \cup \{\leq k \mid k \geq 1\}$

$$
\begin{aligned}
PT_*CD_{REG}(f) = HPT_*CD_{REG} \subset PT_*CD_{LIN}(f) = HPT_*CD_{LIN} \subset \\
MLIN \subset PT_*CD_{MLIN}(g) \subseteq HPT_*CD_{MLIN}.
\end{aligned}
$$

The question is now how far these (hybrid) prescribed team CD grammar systems with teams of variable size and metalinear productions extend beyond the class of metalinear languages. Before presenting a theorem that answers this question, two lemmas are needed.

The proofs of these lemmas are given because the metalinear case is not covered in [9]. Moreover, since the proofs are based on the proofs for the context-free case in [9], they explain the techniques that are used to prove those frequently used results of the next lemmas in the context-free case.

Lemma 5.

$$USC^{[\lambda]}_{MLIN} \subseteq PR^{[\lambda]}_{MLIN} \text{ and } USC^{[\lambda]}_{MLIN,ac} \subseteq PR^{[\lambda]}_{MLIN,ac}.$$

Proof. Only the second statement is proved here (for the λ-free case), the others can be proved in a similar way. Consider an unordered scattered context grammar

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

with appearance checking and only metalinear productions. Define the homomorphism h from $(N \cup T)^*$ into $(\{A' \mid A \in N\} \cup T)^*$ by

$$h(a) = a \text{ for } a \in T \text{ and } h(A) = A' \text{ for } A \in N.$$

Next, for a rule $r : (\alpha_1, \alpha_2, \ldots, \alpha_n) \to (\beta_1, \beta_2, \ldots, \beta_n) \in P$, denote $h(\beta_1 \beta_2 \ldots \beta_n) = w_1 B'_1 w_2 B'_2 \ldots w_m B'_m w_{m+1}$ with $w_i \in T^*$ for $1 \leq i \leq m+1$. To simulate this unordered scattered context grammar with appearance checking, construct the programmed grammar with appearance checking

$$G' = (N', T, S', P'),$$

where

$N' \quad = \quad N \cup \{A' \mid A \in N\} \cup \{S'\}$ and

P' contains for S' the starting productions $(s : S' \to S, \{[r,1,0] \mid r \in P\}, \emptyset)$ and for every production $r : (\alpha_1, \alpha_2, \ldots, \alpha_n) \to (\beta_1, \beta_2, \ldots, \beta_n)$ in P the productions

$$([r,i,0] : \alpha_i \to h(\beta_i), \{[r,i+1,0]\}, \{\emptyset \mid \alpha_i \to \beta_i \notin F\} \cup \{[r,i+1,0] \mid \alpha_i \to \beta_i \in F\}),$$

$$([r,n,0] : \alpha_n \to h(\beta_n), \{[r,1,1]\}, \{\emptyset \mid \alpha_n \to \beta_n \notin F\} \cup \{[r,1,1] \mid \alpha_n \to \beta_n \in F\}),$$

$([r,j,1] : B'_j \to B_j), \{[r,j+1,1]\}, \{[r,j+1,1]\})$ and

$([r,m,1] : B'_m \to B_m), \{[p,1,0] \mid p \in P\}, \{[p,1,0] \mid p \in P\})$

for $1 \leq i \leq n$ and $1 \leq j \leq m$.

Since the scattered context grammar contains only metalinear productions, it is clear that also the productions in this programmed grammar are all metalinear. Moreover, the productions in the programmed grammar simulating the productions in a scattered context rule are applied in a fixed order, possibly passing over a production in case it is contained in F. The use of primes guarantees that the simulating productions are applied only to nonterminals already appearing in the sentential form to be rewritten and not to the ones introduced by a former production of the scattered context rule that is being simulated.

This allows the parallel fashion of a scattered context rule to be simulated by the sequential order of programmed grammar productions. Note that the proof requires the unordered characteristic of the scattered context grammar, for a production $\alpha \to \beta$ can rewrite any occurrence of α in the current sentential form. Obviously, $L(G) = L(G')$ and thus $USC_{MLIN,ac} \subseteq PR_{MLIN,ac}$ holds. □

Lemma 6.

$$MAT^{[\lambda]}_{MLIN} \subseteq USC^{[\lambda]}_{MLIN} \text{ and } MAT^{[\lambda]}_{MLIN,ac} \subseteq USC^{[\lambda]}_{MLIN,ac}.$$

Proof. Again, only the second statement is proved (for the λ-free case), the others can be proved in a similar way. Consider a matrix grammar

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

with appearance checking and only metalinear productions. Denote

$$Lab(M) = \{m_{i,j} \mid m_i : (\alpha_1 \to \beta_1, \alpha_2 \to \beta_2, \ldots, \alpha_n \to \beta_n) \in M, \\ M = \{m_1, m_2, \ldots, m_m\}, 1 \leq i \leq m, 1 \leq j \leq n\}.$$

To simulate this matrix grammar with appearance checking, construct the unordered scattered context grammar with appearance checking

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

where

$$\begin{array}{lll} N' & = & N \cup \{[\alpha,\beta] \mid \alpha \in (N \cup T), \beta \in Lab(M)\} \cup \{S'\} \text{ and} \\ P' & \text{contains} & \text{for } S' \text{ the starting rules } (S') \rightarrow ([S, m_{i,1}]) \text{ for } 1 \leq i \leq n \text{ and} \\ & & \text{for every matrix (with only metalinear productions)} \\ & & m_i : (\alpha_1 \rightarrow \beta_1\gamma_1, \alpha_2 \rightarrow \beta_2\gamma_2, \ldots, \alpha_n \rightarrow \beta_n\gamma_n) \in M, \\ & & \beta_j \in (N \cup T),\ \gamma_j \in (N \cup T)^* \text{ and } 1 \leq j \leq n \\ & & \text{the scattered context rules} \\ & & ([\alpha_j, m_{i,j}]) \rightarrow ([\beta_j, m_{i,j+1}]\gamma_j), \\ & & ([\alpha_n, m_{i,n}]) \rightarrow ([\beta_n, m_{k,1}]\gamma_n), \\ & & ([\delta, m_{i,j}], \alpha_j) \rightarrow ([\delta, m_{i,j+1}], \beta_j\gamma_j), \\ & & ([\delta, m_{i,n}], \alpha_n) \rightarrow ([\delta, m_{k,1}], \beta_n\gamma_n) \text{ and} \\ & & ([\tau, m_{i,1}]) \rightarrow (\tau) \\ & & \text{for } 1 \leq i, k \leq m,\ \delta \in (N \cup T) \text{ and } \tau \in T^*. \end{array}$$

The matrices can be simulated by unordered scattered context rules by adding a label to every symbol of the alphabet. The matrices are split and for every production of it some scattered context rules are created. The labels define an ordering on the use of the various scattered context rules, thus simulating the strict order of matrices by unordered scattered context rules.

At any moment in time, the number of symbols with a label in the sentential form is zero or one. This can be seen from the definitions. If the number is zero, either the sentential form is a terminal one and the derivation is terminated or the sentential form contains a nonterminal and the derivation is blocked since every rule requires a labeled symbol (except the initial rules). If the number is one, it can be replaced by another labeled symbol (the label being the one of the next production in the matrix or the one of the first production of a new matrix if it was the last production of the matrix) while rewriting the symbol according to the production of a matrix being simulated.

Naturally, a production of a scattered context rule can be "passed over" if the same production could be passed over in the matrix grammar, in which case the other production of the scattered context rule replaces the label by the label from the next production in the matrix or the one from the first production of a new matrix if this was the last production of the matrix. Naturally, terminating rules eliminating the labels are present, to be used only when a matrix has been completely simulated. It can now be seen that $L(G) = L(G')$ and $MAT_{MLIN,ac} \subsetneq USC_{MLIN,ac}$ holds. □

Theorem 3. *For* $f \in \{*\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ *and* $g \in \{t_0, t_1, t_2\}$

$$USC^{[\lambda]}_{MLIN} = PR^{[\lambda]}_{MLIN} = MAT^{[\lambda]}_{MLIN} = PT_*CD^{[\lambda]}_{MLIN}(f) \subseteq HPT_*CD^{[\lambda]}_{MLIN} \text{ and}$$

$$USC^{[\lambda]}_{MLIN,ac} = PR^{[\lambda]}_{MLIN,ac} = MAT^{[\lambda]}_{MLIN,ac} = PT_*CD^{[\lambda]}_{MLIN}(g) = HPT_*CD^{[\lambda]}_{MLIN}.$$

Proof. It can be seen from the proofs of $PR^{[\lambda]}_{ac} \subseteq T_sCD^{[\lambda]}(t_1)$ and $PR^{[\lambda]}_{ac} \subseteq T_sCD^{[\lambda]}(t_2)$ in [19] that these results continue to hold in the metalinear case. The same holds for the proofs, in [19] as well, of $PR^{[\lambda]} \subseteq PT_sCD^{[\lambda]}(f)$ and $PT_*CD^{[\lambda]}(f) \subseteq$

$MAT^{[\lambda]}$, for $f \in \{*\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$. Finally, also the proof of $HPT_*CD^{[\lambda]} \subseteq MAT_{ac}^{[\lambda]}$ in [2] can be seen to hold in the case of a restriction to metalinear productions as well. The theorem is hence a simple combination of these results and some obvious inclusions with Lemma 5 and 6. □

Note that this theorem strengthens Theorem 2 to an equality for modes t_0, t_1 and t_2 instead of the last inclusion there. An open problem remains, however, for the other modes of derivation. The next lemma strengthens Lemma 4 and thereby Theorem 2, even more.

Lemma 7. *For* $f \in \{*\} \cup \{\leq k, = k, \geq k \mid k \geq 1\}$ *and* $g \in \{t_0, t_1, t_2\}$

$$MLIN \subset USC_{MLIN} = PR_{MLIN} = MAT_{MLIN} = PT_*CD_{MLIN}(f) \subseteq PT_*CD_{MLIN}(g) = HPT_*CD_{MLIN}.$$

Proof. The first three equalities are proved in Theorem 3. Moreover, the unordered scattered context grammar

$$G_1 = (\{S, A, B, C\}, \{a, b, c\}, S, \{p_1, p_2, p_3\}),$$

with the rules (consisting of only metalinear productions)

$$\begin{aligned} p_1 &: (S) \rightarrow (ABC), \\ p_2 &: (A, B, C) \rightarrow (aA, bB, cC) \text{ and} \\ p_3 &: (A, B, C) \rightarrow (a, b, c) \end{aligned}$$

generates $L(G_1) = \{a^n b^n c^n \mid n \geq 1\} \in USC_{MLIN} \setminus CF$. Since it is known that $\{a^n b^n c^n \mid n \geq 1\} \in CS \setminus CF$ and from the Chomsky hierarchy that $MLIN \subset CF \subset CS$, the first inclusion is proper indeed. The last inclusion is obvious, since clearly $USC_{MLIN} \subseteq USC_{MLIN,ac}$ and, according to Theorem 3, $USC_{MLIN,ac} = PT_*CD_{MLIN}(f)$ for $f \in \{t_0, t_1, t_2\}$. Finally, the last equality was proved in Theorem 3. □

Hence already an unordered scattered context, programmed or matrix grammar without appearance checking and with λ-free productions and only metalinear productions can generate languages beyond the class of metalinear languages. Moreover, also a prescribed team CD grammar system (for all modes of derivation) and a hybrid prescribed team CD grammar system, both with teams of variable size and only metalinear productions, can already generate languages beyond this class of metalinear languages.

For matrix and programmed grammars (with appearance checking) and regular, linear, context-sensitive or recursively enumerable productions only, it is known (see, e.g., [9]) that these cannot generate more than the class of regular, linear, context-sensitive or recursively enumerable languages, respectively. Hence, no interesting results may be expected for unordered scattered context grammars in these cases either.

The next lemma says something about the relation between prescribed team CD grammar systems with teams of bounded size and exactly 1 metalinear production per component and linear simple matrix grammars of degree n. Because of Theorem 1 and its corollaries, this establishes a relation, presented in Corollary 5 right after the coming proof, with the (hybrid) prescribed team CD grammar systems with teams of size 1 and linear or regular productions only.

To be more precise, denote $PT_{(1,n)}CD_{MLIN,1}(f)$ for the class of prescribed team CD grammar systems with teams of size 1 for the teams containing a production with the axiom as its left-hand side, teams of size n for the teams containing other productions, only 1 production per component (the second 1 in the notation), working in mode f and containing only metalinear productions.

Lemma 8. *For* $n \geq 1$ *and* $f \in \{=1, \geq 1, *, t_0, t_1, t_2\} \cup \{\leq k \mid k \geq 1\}$

$$SM_{LIN}(n) \subseteq PT_{(1,n)}CD_{MLIN,1}(f).$$

Proof. Consider the linear simple matrix grammar

$$G = (N_1, N_2, \ldots, N_k, T, S, M)$$

of degree k, $k \geq 1$. To simulate this linear simple matrix grammar, construct the prescribed team CD grammar system

$$\Gamma = (N, T, S, P_1, P_2, \ldots, P_n, Q_1, Q_2, \ldots, Q_m),$$

where

$$N = \bigcup_{i=1}^{n} N_i,$$

$P_1, P_2, \ldots, P_n$ are the components $\{\alpha \to \beta\}$ for every $\alpha \to \beta$ in matrices of M, $\alpha \in \{S\} \cup \bigcup_{i=1}^{n} N_i$ and $\beta \in (\bigcup_{i=1}^{n} N_i \cup T)^*$ and

$Q_1, Q_2, \ldots, Q_m$ are the teams $\{\{S \to \beta\}\}$ for every matrix $(S \to \beta) \in M$ and $\beta \in \bigcup_{i=1}^{n} N_i \cup T^*$ and
the teams $\{\{A_j \to x_j\}\}$ for every matrix of the form $(A_1 \to x_1, A_2 \to x_2, \ldots, A_k \to x_k) \in M, A_i \in N_i$
$x_i \in (N_i \cup T)^*$ and $1 \leq i, j \leq k$.

Note that due to the pairwise disjoint alphabets of simple matrix languages a production $A_j \to x_j$, $1 \leq j \leq k$, does not rewrite a nonterminal introduced by a production $A_i \to x_j$, $1 \leq i < j \leq k$, of the same matrix, but a nonterminal already present in the sentential form before applying this particular matrix to it. It is this property of simple matrix grammars that allows the strict sequential order of rewriting of them to be simulated by one parallel rewriting step of a team of a prescribed team CD grammar system.

Do note also that a characteristic of linear simple matrix grammars is that there can never be two of the same nonterminals in any sentential form. Hence leftmost rewriting is equal to free rewriting in linear simple matrix grammars, thus free rewriting in the simulating prescribed team CD grammar system suffices. These notes imply the restriction to the modes of derivation as stated in the lemma. Moreover, the metalinear productions of the prescribed team CD grammar system allow exactly the axiom to be the left-hand side of non-linear productions, which are precisely the only non-linear productions in a linear simple matrix grammar.

It is clear that $L(G) = L(\Gamma)$. Furthermore, it is easy to see that teams with one component are constructed for productions with the axiom as left-hand side and teams of k components, k being the degree of the simple matrix grammar, are constructed for the other productions. Moreover, all these components of the prescribed team CD grammar system contain only one production. □

Compare the following relations with Theorem 2.

Corollary 5. *For* $f \in \{t_0, t_1, t_2\} \cup \{=k, \geq k \mid k \geq 2\}$, $g \in \{*\} \cup \{\leq k, =k, \geq k \mid k \geq 1\}$ *and* $g' \in \{t_0, t_1, t_2\}$

$$\begin{aligned} PT_1CD_{REG}(f) = HPT_1CD_{REG} \subset PT_1CD_{LIN}(f) = HPT_1CD_{LIN} \subseteq \\ PT_*CD_{MLIN}(g) \subseteq PT_*CD_{MLIN}(g') = HPT_*CD_{MLIN}. \end{aligned}$$

Remark 1. According to Corollary 4, there is a language that can be generated by a (hybrid) prescribed team CD grammar system with teams of size 1 and only linear productions, but which cannot be generated by a context-free grammar. Hence the inclusion $PT_1CD_{LIN}(f) \subseteq PT_*CD_{MLIN}(g)$ in the above corollary is either proper and $CF \subseteq PT_*CD_{MLIN}(g)$ holds or the family of languages generated by prescribed team CD grammar systems with teams of variable size and only metalinear productions is incomparable with the class of context-free languages.

My conjecture is an incomparability result. An intuition supporting a possible proof is the following. It is clear that the context-free grammar $G_2 = (\{S, A\}, \{a, b\}, S, \{S \rightarrow aAbS, S \rightarrow aAb, S \rightarrow ab, A \rightarrow aAb, A \rightarrow ab\})$ generates $L(G_2) = \{a^n b^n \mid n \geq 1\}^+ \in CF$. A characteristic of this language is its unknown width and depth, i.e. the number of $a^n b^n \mid n \geq 1$'s next to each other and for each the amount of n are unknown. Obviously, metalinear productions can simulate the depth with productions similar to the last four productions in G_2. The width, however, has to be known in advance in the case of metalinear productions since the axiom is the only production which can have more than one nonterminal on its right-hand side and should thus introduce a sufficient amount of them. This amount has to be known in advance and the set of productions is finite, hence it seems that $\{a^n b^n \mid n \geq 1\}^+ \notin (PT_*CD_{MLIN}(f) \cup PT_*CD^{\lambda}_{MLIN}(f))$ for $f \in \{*, t_0, t_1, t_2\} \cup \{\leq k, =k, \geq k \mid k \geq 1\}$.

Finally, note that if this conjecture holds, then also the classes of unordered scattered context, matrix and programmed grammars (even with appearance checking) with only metalinear productions are incomparable with the class of context-free languages. For a proof of this, consider for example Theorem 3 and the proof of Lemma 7.

Hence for linear (and regular) simple matrix grammars, a prescribed team CD grammar system with only metalinear productions can be constructed generating the same language. Whether this also holds the other way around and for other modes of derivation, is an open problem.

Another open problem is the relation between simple matrix grammars with only context-free productions and prescribed team CD grammar systems and hence also matrix grammars with only context-free productions (and appearance checking). The leftmost rewriting of simple matrix grammars makes it unlikely to have a similar relation between them and prescribed team CD grammar systems, though. For matrix grammars with only metalinear productions, however, the following corollary does present an interesting relation with regular and linear simple matrix grammars.

Corollary 6. *For* $n \geq 1$

$$SM_{REG}(n) = SM^{\lambda}_{REG}(n) \subset SM_{LIN}(n) = SM^{\lambda}_{LIN}(n) \subseteq MAT_{MLIN} \subseteq MAT^{\lambda}_{MLIN}.$$

Proof. For $n \geq 1$, $SM_{REG}(n) = SM^{\lambda}_{REG}(n) \subset SM_{LIN}(n) = SM^{\lambda}_{LIN}(n)$ (see, e.g., [9]) From Lemma 8 follows that $SM_{LIN}(n) \subseteq PT_{(1,n)}CD_{MLIN,1}(f)$ for $n \geq 1$ and $f \in \{=1, \geq 1, *, t_0, t_1, t_2\} \cup \{\leq k \mid k \geq 1\}$. Moreover, it is clear that $PT_{(1,n)}CD_{MLIN,1}(f) \subseteq PT_*CD_{MLIN}(f)$ for all modes of derivation and thus $SM_{LIN}(n) \subseteq PT_*CD_{MLIN}(f)$ is obtained for $n \geq 1$ and $f \in \{=1, \geq 1, *\} \cup \{\leq k \mid k \geq 1\}$. Finally, Theorem 3 finishes the proof of the corollary. □

$$\begin{tikzcd}
CF = CD(*) = CD(=1) = CD(\geq 1) = CD(\leq k) \text{ (for a } k \geq 1) & \\
& \boldsymbol{PT_*CD_{MLIN}(h_2) = HPT_*CD_{MLIN} = USC_{MLIN,ac} = PR_{MLIN,ac} = MAT_{MLIN,ac}} \\
MLIN = [H]CD_{MLIN}(f) \arrow[uu] \arrow[ur, "h_2 \in \{t_0, t_1, t_2\}"] \arrow[dr, "h_1 \in \{=1, \geq 1, *\}"'] & \\
& \boldsymbol{PT_*CD_{MLIN}(h_1) = USC_{MLIN} = PR_{MLIN} = MAT_{MLIN}} \arrow[uu, dashed, "f \in \{*, t\} \cup \{\leq k, =k, \geq k \mid k \geq 1\}"'] \\
LIN = [H]CD_{LIN}(f) = [\boldsymbol{H}]\boldsymbol{PT_*CD_{LIN}(f)} \arrow[uu] & {[\boldsymbol{H}]\boldsymbol{PT_1CD^{[\lambda]}_{LIN}(g) = SM^{[\lambda]}_{LIN}}} \arrow[u, dashed] \\
REG = [H]CD_{REG}(f) = [\boldsymbol{H}]\boldsymbol{PT_*CD_{REG}(f)} \arrow[u, "f \in \{*, t\} \cup \{\leq k, =k, \geq k \mid k \geq 1\}"'] & {[\boldsymbol{H}]\boldsymbol{PT_1CD^{[\lambda]}_{REG}(g) = SM^{[\lambda]}_{REG}}} \arrow[u, "g \in \{t_0, t_1, t_2\} \cup \{=k, \geq k \mid k \geq 2\}"']
\end{tikzcd}$$

A very interesting corollary indeed, knowing that $SM_{LIN}(n) \subseteq SM(n)$ for $n \geq 1$ (see, e.g., [9]) and keeping in mind the unknown relation between simple matrix grammars and matrix grammars, in the case of context-free productions only, already mentioned above.

5. Summary

A summary of the results presented here will be given in the form of a diagram. A hierarchy along the lines of (the sub-context-free part of) the Chomsky hierarchy (see, e.g., [9]), is chosen. In this way, readers will obtain a clear insight into the power of teams in grammar systems in the sub-context-free cases.

In this diagram, a dashed arrow indicates an inclusion which is not known to be proper, whereas a straight arrow indicates a proper inclusion; in both cases the class the arrow leaves is included in the class the arrow points at. Families which are not connected are not necessarily incomparable. Moreover, all relations of which a proof is included in here are printed in boldface.

Observing this diagram, it is clear that some open problems remain, even though a good insight into the power of (hybrid) prescribed teams in CD grammar systems is offered. One such an open problem concerns a possible hierarchy that can be found in this diagram and it is formulated next.

What is the generative power of prescribed (hybrid) team CD grammar systems with only regular or linear productions and teams of constant size s, for $s \geq 2$?

Acknowledgements. This work has benefited from discussions with and comments and suggestions from E. Csuhaj-Varjú, H.C.M. Kleijn and Gh. Păun. This paper is an excerpt from Part III: *Teams in CD grammar systems* of my master's thesis ([1]).

References

1. M. H. ter Beek, Teams in grammar systems, *IR-96-32 (master's thesis)*, Leiden University, 1996.
2. M. H. ter Beek, Teams in grammar systems: hybridity and weak rewriting. To appear in *Proceedings Workshop Grammar Systems: Recent Results and Perspectives, Budapest, 26-27 July 1996.*
3. E. Csuhaj-Varjú and J. Dassow, On cooperating distributed grammar systems. *J. Inf. Process. Cybern. EIK* 26 (1990), 49 - 63.
4. E. Csuhaj-Varjú, J. Dassow, J. Kelemen and Gh. Păun, *Grammar Systems. A Grammatical Approach to Distribution and Cooperation*, Gordon and Breach, London, 1994.
5. E. Csuhaj-Varjú and J. Kelemen, Cooperating grammar systems: a syntactical framework for the blackboard model of problem solving. In *Proc. AI and information-control systems of robots '89* (I. Plander, ed.), North-Holland Publ. Co., 1989, 121 - 127.
6. E. Csuhaj-Varjú and Gh. Păun, Limiting the size of teams in cooperating grammar systems. *Bulletin EATCS* 51 (1993), 198 - 202.
7. E. Csuhaj-Varjú, Eco-grammar systems: recent results and perspectives. In [18] (1995), 79 - 103.
8. J. Dassow, Cooperating grammar systems (definitions, basic results, open problems). In [18] (1995), 40 - 52.
9. J. Dassow and Gh. Păun, *Regulated Rewriting in Formal Language Theory*, Springer-Verlag, 1989.
10. J. Dassow and Gh. Păun, Cooperating distributed grammar systems with regular components. *Computers and AI* (1992).
11. R. Freund and Gh. Păun, A variant of team cooperation in grammar systems. *J. UCS* 1, 2 (1995), 105 - 130.

12. O. H. Ibarra, Simple matrix languages. *Inform. Control* 17 (1970), 359 - 394.

13. L. Kari, A. Mateescu, Gh. Păun and A. Salomaa, Teams in cooperating grammar systems, *J. Exper. Th. AI* 7 (1995), 347 - 359.

14. O. Mayer, Some restricted devices for context-free languages. *Inform. Control* 20 (1972), 69 - 92.

15. V. Mitrana, Hybrid cooperating distributed grammar systems. *Computers and AI* 2 (1993), 83 - 88.

16. Gh. Păun, On eliminating the λ-rules from simple matrix grammars. *Fundamenta Informaticae* 4 (1981), 185 - 195.

17. Gh. Păun, Grammar systems: a grammatical approach to distribution and cooperation. In *Automata, Languages and Programming; 22nd International Colloquium, ICALP'95, Szeged, Hungary, Lecture Notes in Computer Science* 944 (1995), 429 - 443.

18. Gh. Păun, ed., *Artificial Life: Grammatical Models*, Black Sea Univ. Press, Bucharest, Romania, 1995.

19. Gh. Păun and G. Rozenberg, Prescribed teams of grammars. *Acta Informatica* 31 (1994), 525 - 537.

20. D. J. Rosenkrantz, Programmed grammars and classes of formal languages. *J. ACM* 16, 1 (1969), 107 - 131.

21. G. Rozenberg and A. Salomaa, *The Mathematical Theory of L Systems*, Academic Press, New York, 1980.

22. A. Salomaa, *Formal Languages*, Academic Press, New York, 1973.