Controlled Hybrid CD Grammar Systems

Maurice H. ter Beek(✉)

CNR–ISTI, Pisa, Italy
`maurice.terbeek@isti.cnr.it`

Abstract. We study the generative power of hybrid CD grammar systems whose derivations are controlled by a graph, a hypothesis language or a generalized sequential machine. We relate them to the families of languages generated by matrix grammars with appearance checking. We thus characterise language families that lie in between that of the families of context-free and context-sensitive languages, which are of particular interest to linguists. In particular, we show that controlled hybrid CD grammar systems can generate the non-context-free features multiple agreements, crossed agreements and (re)duplication that occur in many natural languages.

Keywords: Grammar systems · controlled derivation · non-context-free

1 Introduction

Erzsébet Csuhaj-Varjú's impressive legacy includes the introduction and development of (cooperating/distributed) grammar systems to formalize the relations between the blackboard model of (cooperative) problem solving (by multi-agent systems) known from Artificial Intelligence and formal languages [19,21]. The cooperating agents are modelled as grammars that edit the blackboard by distributed rewriting of a sentential form taking turns. Their problem-solving strategy is regulated by derivation modes and the solutions are represented by terminal words. Since their introduction, grammar systems have been studied intensively, including well-motivated extensions like *controlled* and *hybrid* CD grammar systems, *teams* in CD grammar systems and combinations thereof [4,5,23,24,34,37,41,42,44,45].

In this paper, we report on the generative power of several kinds of controlled hybrid CD grammar systems, in particular their relations with the family of languages generated by matrix grammars with appearance checking. The reason is that the generative power of this family is known to lie in between that of the family of languages generated by context-free grammars and that of the one generated by context-sensitive grammars, which has been a sweet spot of particular interest to linguists for many decades. Since 1959, Chomsky has inspired

M. D. Jiménez López and G. Vaszil (Eds.): Erzsébet Csuhaj-Varjú Festschrift,
LNCS 15840, pp. 3–25, 2025.
https://doi.org/10.1007/978-3-031-97274-4_1

many linguists and theoretical computer scientists to wonder about the position of natural languages in what is now known as the Chomsky hierarchy [18]:

> "*the main problem of immediate relevance to the theory of language is that of determining where in the hierarchy of devices the grammars of natural languages lie.*"

While the exact position remains to be settled, linguists have narrowed it down since they apparently agree that many natural languages contain non-context-free constructions [51] while at the same time the generative power of context-sensitive grammars is too computationally expensive to parse [33]. The three basic features that occur in natural languages like English [35], Dutch [36], Swiss German [17], Mali's lingua franca Bambara [22] and Chinese [46] are so-called *multiple agreements*, *crossed agreements* and *(re)duplication* [2,27]. Languages with these three features can be generated by (controlled) (hybrid) CD grammar systems, as we will show in Examples 1, 2 (and 4) and 3, respectively.

Outline. In Sect. 2, we define some basic notions from formal language theory, such as Chomsky grammars and matrix grammars, followed by the definition of CD grammar systems in Sect. 3. We then extend the latter to hybrid CD grammar systems in Sect. 4, to controlled CD grammar systems in Sect. 5 and to various types of controlled hybrid CD grammar systems in Sect. 6. Section 7 presents the results obtained for the newly defined families of languages and positions them in the literature, followed by concluding remarks in Sect. 8.

2 Preliminaries

In this section, some prerequisites necessary for understanding the sequel are defined. For details and unexplained notions, the reader is referred to [48–50] for formal languages in general and in particular for L systems [38,47], regulated rewriting [25,27] and grammar systems [20,26].

The set of all non-empty strings over an *alphabet* V is denoted by V^+; V^* if the *empty string* λ is included. The *length* of a string x is denoted by $|x|$; $|\lambda| = 0$. Set *inclusion* is denoted by $\subseteq$, *proper inclusion* by $\subset$ and set *difference* by $\setminus$.

A *Chomsky grammar* of *type* i ($i \in \{0,1,2,3\}$) is a construct $G = (N, T, P, S)$, where N is the set of *nonterminals*, T is the set of *terminals*, $N \cap T = \varnothing$, $S \in N$ is the *axiom* and P is a finite set of *productions* of the form i, defined as follows:

(0) $\alpha \to \beta$, for $\alpha, \beta \in (N \cup T)^*$ and $|\alpha|_N \geq 1$,
(1) $w_1 A w_2 \to w_1 x w_2$, for $w_1, w_2 \in (N \cup T)^*$, $x \in (N \cup T)^+$ and $A \in N$, and $S \to \lambda$ is allowed iff S does not occur in the right-hand side of any production,
(2) $A \to x$, for $A \in N$ and $x \in (N \cup T)^*$, and
(3) $A \to xB$ or $A \to x$, for $A, B \in N$ and $x \in T^*$.

Moreover, two other well-known types of grammars G are used in the sequel, defined with a finite set of productions P of the form j, with $j \in \{4, 5\}$, as follows:

(4) $A \rightarrow x_1Bx_2$ or $A \rightarrow x$, for $A, B \in N$ and $x, x_1, x_2 \in T^*$, and
(5) $A \rightarrow x_1Bx_2$ or $A \rightarrow x$, for $A, B \in N$ and $x, x_1, x_2 \in T^*$, and productions of the form $S \rightarrow x_1A_1x_2A_2 \dots x_nA_nx_{n+1}$ are allowed for $A_i \in N$, $x_i \in T^*$ and $1 \leq i \leq n$ iff S does not appear in the right-hand side of any production.

A string x *directly derives* a string y in G, denoted by $x \Longrightarrow_G y$, iff $x = w_1\alpha w_2$, $y = w_1\beta w_2$ and $\alpha \rightarrow \beta \in P$, for $w_1, w_2 \in (N \cup T)^*$. This is also called a one-step derivation in G; consequently a k-step derivation (for $k \geq 0$) in G, denoted by $\Longrightarrow_G^k$, is defined for $x, y \in (N \cup T)^*$ as $x \Longrightarrow_G^k y$ iff there are words $x_0, x_1, \dots, x_k$ such that $x = x_0$, $y = x_k$ and $x_i \Longrightarrow_G x_{i+1}$, with $0 \leq i \leq k-1$. If G is clear from the context, it is omitted, writing only $\Longrightarrow$ and $\Longrightarrow^k$, respectively. This applies to all definitions of derivation steps in the sequel. The transitive (and reflexive) closure of the one-step derivation is denoted by $\Longrightarrow^+$ ($\Longrightarrow^*$).

The language generated by G is denoted by $L(G)$ and it is defined by:

$$L(G) = \{\, w \in T^* \mid S \Longrightarrow_G^* w \,\}.$$

A word $w \in (N \cup T)^*$ is called a *sentential form* (*terminal word* if $w \in T^*$) of G iff $S \Longrightarrow_G^* w$, so the language generated by G consists of all terminal words of G.

A language is said to be of type i iff it is generated by a Chomsky grammar of type i, with $i \in \{0, 1, 2, 3\}$. Type 0 grammars are also called *phrase structure grammars* and the family of type 0 languages is called *recursively enumerable*, denoted by RE.

Type 1 grammars and language are also called *context-sensitive* and their family of languages is denoted by CS. A *non-contracting* (or length-increasing, monotonous) grammar is a Chomsky grammar (N, T, P, S) such that the productions in P are of the form $\alpha \rightarrow \beta$, for $\alpha, \beta \in (N \cup T)^+$ and $|\alpha| \leq |\beta|$, and $S \rightarrow \lambda$ is allowed iff S does not occur in the right-hand side of any production. Context-sensitive grammars and non-contracting grammars generate the same family of languages CS.

Type 2 grammars and languages are also called *context-free* and their family of languages is denoted by CF.

In the literature, type 3 grammars and languages are called *right-linear*. When in condition (3) above the requirement $x \in T^*$ is replaced by $x \in T$, the definition of a *simply regular* grammar is obtained. Right-linear and simply regular grammars generate the same family of languages. In the sequel, grammars in the sense of condition (3) will be considered and are called *regular*; their family of languages is denoted by REG.

Furthermore, grammars and languages of type 4 (5) are called *linear* (*metalinear*) and their family of languages is denoted by LIN ($MLIN$).

Finally, a grammar is called λ-*free* if it does not contain any production $\alpha \rightarrow \lambda$, or if the only λ-production it contains is $S \rightarrow \lambda$, with S the axiom not appearing in the right-hand side of any production of the grammar.

A *generalized sequential machine* (gsm) is a construct $g = (K, I, O, s_0, \delta, H)$, where K is a finite non-empty set of *states*, I is the non-empty *input* alphabet,

O is the non-empty *output* alphabet, $s_0 \in K$ is the *initial state*, $H \subseteq K$ is the set of *final states* and δ is a finite set of productions of the form:

$$s_i v \to w s_j, \text{ for } s_i, s_j \in K, \; v \in I \text{ and } w \in O^*.$$

Let $\Rightarrow^*$ denote the transitive and reflexive closure of $\to$. For a word $v \in I^+$, let

$$g(v) = \{w \mid s_0 v \Rightarrow^* w s_z \text{ for some } s_z \in H\}.$$

Then we define a *gsm mapping*, for a gsm g and a language L over I, as follows:

$$g(L) = \{z \mid z \in g(v) \text{ for some } v \in L\}.$$

Without providing its definition, in the sequel we will also refer to the family of languages generated by ET0L systems ($ET0L$) without using it in any construction in the proofs. Definitions of this language family can be found in, e.g., [47,48]. The family of languages that is used in the proofs is defined next.

A *matrix grammar (with appearance checking, ac for short)* [1] 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 $|\alpha_i|_N \geq 1$, with $1 \leq i \leq n$, and F 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 $F = \varnothing$, the matrix grammar is called *without appearance checking* and F is omitted from the construct. 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 appearance checking with λ-free context-free productions of type 2 is denoted by MAT_{ac} (MAT^{λ}_{ac} when not restricted to λ-free productions); MAT for matrix grammars without appearance checking. The following hierarchy, which includes the so-called Chomsky hierarchy for Chomsky grammars, is well known:

$$REG \subset LIN \subset MLIN \subset CF \subset MAT \subset MAT_{ac} \subset CS \subset MAT^{\lambda}_{ac} = RE.$$

3 CD Grammar Systems

Cooperating grammar systems were introduced in [40], the motivation coming from the theory of two-level grammars. Ten years later, they received renewed attention when a link with the notion of the blackboard model of problem solving from the theory of AI was established in [21], as described in the Introduction. A more general form was introduced in [19] and is presented next.

Definition 1. *A* cooperating distributed grammar system*, CD grammar system for short, is a construct* $\mathit{\Gamma} = (N, T, S, P_1, P_2, \ldots, P_n)$*, where* N *is the set of nonterminals,* T *is the set of terminals,* $S \in N$ *is the axiom and each* P_i*, with* $1 \leq i \leq n$*, is a finite set of productions over* $N \cup T$*, called a* component *of* $\mathit{\Gamma}$*.*

CD grammar systems rewrite differently than the grammars presented so far.

Definition 2. *Let* $\mathit{\Gamma} = (N, T, S, P_1, P_2, \ldots, P_n)$ *be a CD grammar system. Then* $\mathit{\Gamma}$ *can rewrite according to one of the following five* modes of derivation*. For all modes, let* $x, y, z \in (N \cup T)^*$*,* $k \geq 1$*,* $1 \leq i \leq n$ *and* $\Longrightarrow^{\ell}_{P_i}$ *is used for the* ℓ*-step derivation* $\Longrightarrow^{\ell}$ *as defined for a Chomsky grammar* (N, T, S, P_i)*.*

$\leq k$ *This mode corresponds to* at most k *direct derivation steps in succession by some component* P_i *in the CD grammar system:*

$$x \Longrightarrow^{\leq k}_{\mathit{\Gamma}} y \text{ iff there exists } P_i \text{ such that } x \Longrightarrow^{0}_{P_i} y \text{ or } x \Longrightarrow^{k'}_{P_i} y \text{ for some } k' \leq k.$$

$=k$ *This mode corresponds to* exactly k *direct derivation steps in succession by some component* P_i *in the CD grammar system:*

$$x \Longrightarrow^{=k}_{\mathit{\Gamma}} y \text{ iff there exists } P_i \text{ such that } x \Longrightarrow^{k}_{P_i} y.$$

$\geq k$ *This mode corresponds to* at least k *direct derivation steps in succession by some component* P_i *in the CD grammar system:*

$$x \Longrightarrow^{\geq k}_{\mathit{\Gamma}} y \text{ iff there exists } P_i \text{ such that } x \Longrightarrow^{k'}_{P_i} y \text{ for some } k' \geq k.$$

$*$ *This mode corresponds to* an arbitrary number *of direct derivation steps in succession by some component* P_i *in the CD grammar system:*

$$x \Longrightarrow^{*}_{\mathit{\Gamma}} y \text{ iff there exists } P_i \text{ such that } x \Longrightarrow^{0}_{P_i} y \text{ or } x \Longrightarrow^{k}_{P_i} y \text{ for some } k.$$

t *This mode corresponds to* maximal *derivations by some component* P_i *in the CD grammar system (i.e., the component must rewrite the sentential form as long as it is able to):*

$$x \Longrightarrow^{t}_{\mathit{\Gamma}} y \text{ iff there exists } P_i \text{ such that } x \Longrightarrow^{*}_{P_i} y \text{ and no } z \text{ such that } y \Longrightarrow_{P_i} z.$$

The language generated by a CD grammar system depends on the mode of derivation according to which it rewrites.

Definition 3. *Let* $\mathit{\Gamma} = (N, T, S, P_1, P_2, \ldots, P_n)$ *be a CD grammar system. The language generated by* $\mathit{\Gamma}$ *in derivation mode* f*, for* $f \in \{\, {\leq}k, {=}k, {\geq}k \mid k \geq 1 \,\} \cup \{*, t\}$*, is denoted by:*

$$L_f(\mathit{\Gamma}) = \{z \in T^* \mid S \Longrightarrow^{f}_{\mathit{\Gamma}} w_1 \Longrightarrow^{f}_{\mathit{\Gamma}} \cdots \Longrightarrow^{f}_{\mathit{\Gamma}} w_m = z,\ m \geq 1\}.$$

The family of languages that is generated by CD grammar systems $(N, T, S, P_1, P_2, \ldots, P_m)$ with only λ-free context-free productions of type 2 in each P_j, working in derivation mode f, and $1 \leq j \leq m \leq n$, is denoted by $CD_n(f)$. Denote $CD(f) = \bigcup_{n \geq 1} CD_n(f)$.

This definition is illustrated by the following example.

Example 1. Consider the CD grammar system

$$\Gamma_1 = (\{S, A, B, A', B'\}, \{a, b, c\}, S, P_1, P_2, P_3),$$

where

$$\begin{aligned} P_1 &= \{S \to AB, A' \to A, B' \to B\}, \\ P_2 &= \{A \to a^4 A' b^4, B \to c^4 B'\} \text{ and} \\ P_3 &= \{A \to ab, B \to c\}. \end{aligned}$$

Suppose that this CD grammar system works in the maximal derivation mode t. Clearly, the first component to be applied is P_1, resulting in AB. Then either component P_3 is used, resulting in the terminal word abc, or P_2 is used. After using P_2, the sentential form is $a^4A'b^4c^4B'$ and only P_1 can be used, resulting in $a^4Ab^4c^4B$. Using P_2 and P_1 iteratively, strings of the form $a^iAb^ic^iB$ are generated, where $i \textit{ mod } 4 \equiv 0$. This string can be rewritten into a terminal one by using P_3 and hence the language generated by Γ_1 operating in mode t is:

$$L_t(\Gamma_1) = \{\, a^n b^n c^n \mid n \textit{ mod } 4 \equiv 1,\ n \geq 1 \,\}.$$

This language is not a context-free language, thus demonstrating that CD grammar systems with context-free components are able to produce languages not in the context-free language class.

In fact, concerning the generative power of CD grammar systems, we know from [19,20] that for $f \in \{\,{=}k, {\geq}k \mid k \geq 2\,\}$, $k \geq 1$ and $k', k'' \geq 2$, holds that:

$$\begin{aligned} &CF = CD({=}1) = CD({\geq}1) = CD(*) = CD({\leq}k) \subset (CD({=}k') \cap CD({\geq}k'')), \\ &CF = CD_1(f) \subset CD_2(f) \subseteq CD_3(f) \subseteq \ldots \subseteq CD(f) \subseteq MAT \text{ and} \\ &CF = CD_1(t) = CD_2(t) \subset CD_3(t) = CD(t) = ET0L. \end{aligned}$$

4 Hybrid CD Grammar Systems

The idea of considering the agents of a multi-agent system to have different capabilities, formally modelled by allowing different modes of derivation to be associated with different components, was introduced into the model of CD grammar systems in [42]. This has resulted in hybrid CD grammar systems [42,43], a.k.a. *externally* hybrid CD grammar systems to distinguish them from *internally* hybrid CD grammar systems [29–31]. In the latter, not only different components may work according to different (classical) modes of derivation (e.g., in the ${\leq}k$-mode a component has to perform at most k steps) but components may moreover work according to arbitrary Boolean combinations of classical modes of derivation (e.g., in the $(t \wedge {\leq}k)$-mode a component has to perform as many derivation steps as possible, but at most k steps).

Definition 4. *A* hybrid cooperating distributed (CD) grammar system *is a construct* $\Gamma = (N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n))$*, where* N *is the set of nonterminals,* T *is the set of terminals,* $S \in N$ *is the axiom,* $P_1, P_2, \ldots, P_n$ *are the components as for usual CD grammar systems and* $f_i \in \{\, \leq k, = k, \geq k \mid k \geq 1 \,\} \cup \{*, t\}$ *is the mode of derivation associated with the component* P_i*, with* $1 \leq i \leq n$*. The language generated by* Γ *is:*

$$\begin{aligned} L(\Gamma) =\{\, z \in T^* \mid S \Longrightarrow^{f_{i_1}}_{P_{i_1}} w_{i_1} \Longrightarrow^{f_{i_2}}_{P_{i_2}} \cdots \Longrightarrow^{f_{i_m}}_{P_{i_m}} w_{i_m} = z, \\ 1 \leq i_j \leq n,\ 1 \leq j \leq m \,\}. \end{aligned}$$

The family of languages that is generated by hybrid CD grammar systems $(N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_m, f_m))$ with only λ-free context-free productions of type 2 in each P_j and $1 \leq j \leq m \leq n$, is denoted by HCD_n. Denote $HCD = \bigcup_{n \geq 1} HCD_n$.

Also this definition is illustrated by an example.

Example 2. Consider the hybrid CD grammar system

$$\begin{aligned} \Gamma_2 =(\{S, A, B, C, D, A', B', C', D'\}, \{a, b, c, d\}, S, \\ (P_1, =1), (P_2, =2), (P_3, =2), (P_4, t), (P_5, =4)), \end{aligned}$$

where

$$\begin{aligned} P_1 &= \{S \to ABCD\}, \\ P_2 &= \{A \to aA', C \to cC'\}, \\ P_3 &= \{B \to bB', D \to dD'\}, \\ P_4 &= \{A' \to A, B' \to B, C' \to C, D' \to D\} \text{ and} \\ P_5 &= \{A \to a, B \to b, C \to c, D \to d\}. \end{aligned}$$

Obviously, every successful derivation starts with the application of P_1, thus generating $ABCD$. There are three possibilities to continue. When P_5 is applied, the terminal word $abcd$ is generated since P_5 operates in mode $=4$. Otherwise, either P_2 or P_3 has to be used in mode $=2$, resulting in $aA'BcC'D$ or $AbB'CdD'$, respectively. Then, in case P_2 (P_3) was used in the last step, the derivation can proceed with P_3 (P_2) or P_4. Using P_2 as well as P_3, the only possibility to continue is by using P_4. Hence, in all cases (basically only two different ones) there comes a time when P_4 is used. This P_4 works in mode t, which makes it possible to remove the primes from only A and C or from only B and D or from all four of them, in all cases leaving no primed nonterminals and thus fulfil the stop condition of mode t. Consequently, this process can be iterated until eventually P_5 is used to replace A, B, C and D and thus fulfilling the stop condition for mode $=4$. This explanation shows that the generated language is:

$$L(\Gamma_2) = \{a^n b^m c^n d^m \mid m, n \geq 1\}.$$

Like the language generated in Example 1, this language is non-context-free.

Concerning the generative power of hybrid CD grammar systems, we know from the definitions and [42,43] that for $f \in \{\leq k, = k, \geq k \mid k \geq 1\} \cup \{*, t\}$, $n' \geq 1$ and $n \geq 4$, holds that:

$$\begin{aligned} \mathit{CF} = \mathit{CD}_1(f) = \mathit{HCD}_1 \subset \mathit{HCD}_2 \subseteq \mathit{HCD}_3 \subseteq \mathit{HCD}_4 = \mathit{HCD}_n = \mathit{HCD} \subseteq \mathit{MAT}_{ac}, \\ \mathit{ET0L} = \mathit{CD}_3(t) \subseteq \mathit{HCD}_3, \\ \mathit{CD}(f) \subseteq \mathit{HCD}_3, \\ \mathit{ET0L} \subset \mathit{HCD}_4 \text{ and} \\ \mathit{CD}_{n'}(f) \subseteq \mathit{HCD}_{n'}. \end{aligned}$$

Furthermore, in [42] it was shown that for each hybrid CD grammar system, an equivalent hybrid CD grammar system can be constructed that contains three components working in the t-mode and one in the $=k$-mode, for some $k \geq 1$.

For the generative devices mentioned so far, only the notation for λ-free context-free productions of type 2 was given. Here and in the sequel, however, when productions are of type X, for $X \in \{\mathit{REG}, \mathit{LIN}, \mathit{MLIN}, \mathit{CF}, \mathit{CS}, \mathit{RE}\}$, a subscript X may be added to the notation; moreover, a superscript λ is added when not restricted to λ-free productions.

Concerning the generative power of hybrid CD grammar systems with sub-context-free productions, we know from [3,5,20,42] that for $f \in \{\leq k, = k, \geq k \mid k \geq 1\} \cup \{*, t\}$ holds that:

$$\begin{aligned} \mathit{REG} &= \mathit{CD}_{\mathit{REG}}(f) = \mathit{HCD}_{\mathit{REG}}, \\ \mathit{LIN} &= \mathit{CD}_{\mathit{LIN}}(f) = \mathit{HCD}_{\mathit{LIN}} \text{ and} \\ \mathit{MLIN} &= \mathit{CD}_{\mathit{MLIN}}(f) = \mathit{HCD}_{\mathit{MLIN}}. \end{aligned}$$

5 Controlled CD Grammar Systems

Apart from hybrid CD grammar systems, numerous other types of CD grammar systems have been defined and investigated in the literature, among which grammar systems controlled by a directed graph, in which derivations are guided by paths in the graph (cf., e.g., [19,24,28,34,40]), or by a Petri net (cf. [15]), which allow (vector controlled) concurrent rewriting, grammar systems whose sentential forms are put into a gsm before another derivation step can take place (cf., e.g., [41,44]), and grammar systems whose sentential forms are compared to a (regular) hypothesis language, accepting only those sentential forms satisfying the format of the hypothesis language (cf., e.g., [23,44]).

We denote the above mentioned families of languages that are generated by CD grammar systems with graph control (GC), with a hypothesis language (HL) and with a gsm (GSM), with only λ-free productions of type X, for $X \in \{\mathit{REG}, \mathit{LIN}, \mathit{MLIN}, \mathit{CF}, \mathit{CS}, \mathit{RE}\}$, and working in derivation mode f, for $f \in \{\leq k, = k, \geq k \mid k \geq 1\} \cup \{*, t\}$, by $\mathit{YCD}_X(f)$, where $Y \in \{\mathit{GC}, \mathit{HL}, \mathit{GSM}\}$. Typically, subscript X is omitted if $x = \mathit{CF}$.

To put the results in this section in a proper perspective, we restate some results from [19,23,41] in the following lemma.

Lemma 1. *For* $k, k' \geq 1$ *and* $f \in \{\, \leq k, = k, \geq k \mid k \geq 1 \,\} \cup \{*, t\}$ *holds that:*

1. $GCCD_{REG}(=k) = GCCD(\leq k) = GCCD(=k) = MAT$,
   $GCCD(\geq k) = GCCD(\geq k') \subseteq MAT$ *and*
   $CF = GCCD(*) \subset GCCD_{REG}(t) = GCCD(t) = ET0L$,
2. $HLCD(f) = CS$ *and*
3. $GSMCD(f) = CS$.

It is not difficult to see that every (hybrid) CD grammar system is also a controlled (hybrid) CD grammar system.

Lemma 2. *For* $X \in \{REG, LIN, MLIN, CF, CS, RE\}$, $Y \in \{GC, HL, GSM\}$ *and* $f \in \{\, \leq k, = k, \geq k \mid k \geq 1 \,\} \cup \{*, t\}$ *holds that:*

1. $CD_X(f) \subseteq YCD_X(f)$ *and*
2. $HCD_X \subseteq YHCD_X$.

Proof. The inclusions follow when the controlling devices impose no restriction. Thus for $Y = GC$, consider the complete graph C_n with n nodes. For $Y = HL$, consider the regular language $R = (N \cup T)^* \setminus T^*$. For $Y = GSM$, finally, consider the gsm that does not translate but only copies the string it receives as input, i.e., $g = (\{s_0\}, N \cup T, N \cup T, s_0, \{(s_0 x, x s_0) \mid x \in (N \cup T)\}, \{s_0\})$. □

6 Controlled Hybrid CD Grammar Systems

In this section, we combine the ideas from the previous sections and investigate the generative power of controlled hybrid CD grammar systems. These results, most of them originally reported in the MSc thesis [3], have not been published before. Lemma 4 is new.

6.1 Hybrid CD Grammar Systems with Static Control

First the case of a graph as *static* control mechanism is considered. The control is called static, since the current state of the problem is not taken into consideration. The notion of derivations controlled by a directed graph has already been presented in [19,28,40,52] for Chomsky grammars, cooperating grammar systems, CD grammar systems and CD grammar systems with singleton components (or, equivalently, graph-controlled grammars with derivation modes) working in classical modes or internally hybrid modes (i.e., arbitrary Boolean combinations of classical modes, cf. Sect. 4). Results on graph controlled CD grammar systems with appearance checking and characterizations of graphs associated to specific language classes can be found in [24,34].

Definition 5. *A* hybrid CD grammar system with graph control *is a construct*

$$\Gamma = (N, T, S, (P_1, f_1), (P_2, f_2), \dots, (P_n, P_n), U),$$

where $(N, T, S, (P_1, f_1), (P_2, f_2), \dots, (P_n, P_n))$ *is a hybrid CD grammar system and* $U = (V, E)$ *is a directed graph with set of nodes* V *and set of edges* E*, the* n *nodes of which are labelled by* P_i*, with* $1 \leq i \leq n$*.*

For $f_i \in \{\, \leq k, = k, \geq k \mid k \geq 1\,\} \cup \{*, t\}$ *and* $1 \leq i \leq n$*, the language generated by* Γ *and controlled by* U *is:*

$$\begin{aligned} L^U(\Gamma) = \{\, z \in T^* \mid S \Longrightarrow_{P_{i_1}}^{f_{i_1}} w_{i_1} \Longrightarrow_{P_{i_2}}^{f_{i_2}} \cdots \Longrightarrow_{P_{i_m}}^{f_{i_m}} w_{i_m} = z, \\ (P_{i_k},\ P_{i_{k+1}}) \in E,\ 1 \leq i_j \leq n,\ 1 \leq j \leq m,\ 1 \leq k \leq m-1 \,\}. \end{aligned}$$

Note that the graph has exactly one node for each component labelled by the component.

The family of languages that is generated by hybrid CD grammar systems with graph control $(N, T, S, (P_1, f_1), (P_2, f_2), \dots, (P_m, f_m), U)$ with only λ-free context-free productions of type 2 in each P_j and $1 \leq j \leq m \leq n$, is denoted by $GCHCD_n$. Denote $GCHCD = \bigcup_{n \geq 1} GCHCD_n$.

The next example illustrates hybrid CD grammar systems with graph control.

Example 3. Consider the hybrid CD grammar system with graph control

$$\Gamma_3 = (\{S, A, B\}, \{a, b\}, S, (P_1, t), (P_2, = 1), (P_3, t), (P_4, t), (P_5, t), U),$$

where

$$\begin{aligned} &P_1 = \{S \rightarrow ABS, S \rightarrow AB\}, \\ &P_2 = \{A \rightarrow a\}, \\ &P_3 = \{B \rightarrow bB'\}, \\ &P_4 = \{B \rightarrow b\}, \\ &P_5 = \{B' \rightarrow B\} \text{ and} \end{aligned}$$

U is the following graph:

$$\begin{tikzcd} P_1 \arrow[r] & P_2 \arrow[r] \arrow[d] & P_4 \\ P_5 \arrow[ur] & P_3 \arrow[l] & \end{tikzcd}$$

Using P_1 for i times results in the sentential form $(AB)^i$, with $i \geq 1$. Then every time P_2 changes one A into an a, the components P_3 and P_5 change all B's into bB. This is repeated until there is only one A remaining in the sentential form, after which another path in the graph is taken to replace also this last A by a and consequently replacing all B's by b, thus obtaining a terminal string. No terminal string can be obtained if the last A's replacement by a is followed by replacing all B's by bB' since this would eventually require another application of P_2 before yielding a terminal string, which is not possible due to the absence of A's. It is thus clear that

$$L^U(\Gamma_3) = \{(ab^n)^n \mid n \geq 1\}.$$

Hence already only one non-metalinear production and a simple graph suffice to generate a non-ET0L language.

The proof of the next lemma makes use of the *domain* of a component P of a grammar system with set N of nonterminals, which is defined as $dom(P) = \{ A \in N \mid A \to x \in P \}$.

Lemma 3. *It holds that* $GCHCD \subseteq MAT_{ac}$ *and* $GCHCD^{\lambda} \subseteq MAT_{ac}^{\lambda}$.

Proof. Consider a graph controlled hybrid CD grammar system

$$\Gamma = (N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n), U).$$

To simulate this hybrid CD grammar system with graph control, construct the following matrix grammar with appearance checking

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

where

$$\begin{aligned}
N' &= N \cup \{S'\} \cup \{ [P_i, f_i], [P_i, \geq k]' \mid (P_i, f_i), (P_i, \geq k) \in \Gamma,\ 1 \leq i \leq n \},\\
T' &= T \cup \{z\},\\
M' &= \{ (S' \to S[P_i, f_i]) \mid f_i \in \{ \leq k, =k, \geq k \mid k \geq 1 \} \cup \{*, t\},\ 1 \leq i \leq n \} \cup\\
&\quad \{ ([P_i, \leq k] \to [P_j, f_j],\ A_1 \to x_1, A_2 \to x_2, \ldots, A_h \to x_h) \mid\\
&\qquad A_g \to x_g \in P_i,\ f_j \in \{ \leq k, =k, \geq k \mid k \geq 1 \} \cup \{*, t\},\ 1 \leq g \leq h \leq k,\\
&\qquad (P_i, P_j) \in E,\ 1 \leq i, j \leq n \} \cup\\
&\quad \{ ([P_i, \leq k] \to [P_j, f_j]) \mid f_j \in \{ \leq k, =k, \geq k \mid k \geq 1 \} \cup \{*, t\},\\
&\qquad 1 \leq i, j \leq n \} \cup\\
&\quad \{ ([P_i, =k] \to [P_j, f_j],\ A_1 \to x_1, A_2 \to x_2, \ldots, A_k \to x_k) \mid\\
&\qquad A_g \to x_g \in P_i,\ f_j \in \{ \leq k, =k, \geq k \mid k \geq 1 \} \cup \{*, t\},\ 1 \leq g \leq k,\\
&\qquad (P_i, P_j) \in E,\ 1 \leq i, j \leq n \} \cup\\
&\quad \{ ([P_i, \geq k] \to [P, \geq k]',\ A_1 \to x_1, A_2 \to x_2, \ldots, A_k \to x_k) \mid\\
&\qquad A_g \to x_g \in P_i,\ 1 \leq g \leq k,\ 1 \leq i \leq n \} \cup\\
&\quad \{ ([P_i, \geq k]' \to [P_i, \geq k]',\ A \to x) \mid A \to x \in P_i,\ 1 \leq i \leq n \} \cup\\
&\quad \{ ([P_i, \geq k]' \to [P_j, f_j]) \mid f_j \in \{ \leq k, =k, \geq k \mid k \geq 1 \} \cup \{*, t\},\\
&\qquad (P_i, P_j) \in E,\ 1 \leq i, j \leq n \} \cup\\
&\quad \{ ([P_i, *] \to [P_i, *],\ A \to x) \mid A \to x \in P_i,\ 1 \leq i \leq n \} \cup\\
&\quad \{ ([P_i, *] \to [P_j, f_j]) \mid f_j \in \{ \leq k, =k, \geq k \mid k \geq 1 \} \cup \{*, t\},\\
&\qquad (P_i, P_j) \in E,\ 1 \leq i, j \leq n \} \cup\\
&\quad \{ ([P_i, t] \to [P_i, t],\ A \to x) \mid A \to x \in P_i,\ 1 \leq i \leq n \} \cup\\
&\quad \{ ([P_i, t] \to [P_j, f_j],\ A_{i_1} \to F, A_{i_2} \to F, \ldots, A_{i_{s_i}} \to F) \mid\\
&\qquad dom(P_i) = \{A_{i_1}, A_{i_2}, \ldots, A_{i_{s_i}}\},\ f_j \in \{ \leq k, =k, \geq k \mid k \geq 1 \} \cup \{*, t\},\\
&\qquad (P_i, P_j) \in E,\ 1 \leq i, j \leq n \} \cup\\
&\quad \{ ([P_i, f_i] \to z) \mid f_i \in \{ \leq k, =k, \geq k \mid k \geq 1 \} \cup \{*, t\},\ 1 \leq i \leq n \} \text{ and}
\end{aligned}$$

in F' are all the productions $A \to F$ appearing in M'.

The simulation starts with applying a production $S' \to S[P_i, f_i]$, where S is the original axiom and $[P_i, f_i]$ a marker, indicating which component is being simulated and what its mode of derivation is. From S, the language of the graph controlled hybrid CD grammar system will be generated and the marker will control this generation.

When the mode is $\leq k$, indeed less than k times a production from the corresponding component is used before handing over control to another component. Furthermore, this other component has to be connected to the current component by an edge in the controlling graph. This test is done throughout the whole construction, before handing over control to another component. In the case of mode $= k$, exactly k productions are used. For mode $\geq k$, first exactly k productions are used, after which the primed version of $[P_i, \geq k]$ is used to hand over control to another component only after another zero or more rewriting steps.

If the mode is $*$, an arbitrary number of productions is used before handing over control. Finally, in mode t the same construction is used to rewrite an arbitrary number of times. Moreover, the productions in the set F' guarantee that in this mode control can only be handed over to another component when no more production of the particular component can be used. In the case of mode $\leq k$ and $*$ the control can also directly be given to another component, corresponding to the case when the less than k or arbitrary number of rewriting steps is in fact zero.

Eventually, the marker is replaced by z thus yielding $L(G') = L(\Gamma)\{z\}$. This z can be removed by a morphism and thus, since it is known (cf., e.g. [25]) that the family MAT_{ac} is closed under restricted morphisms, $L(\Gamma) \in MAT_{ac}$ and the first statement of the lemma is proved.

The second statement of the lemma can be proved by a similar construction, even simplified since the marker can be replaced by λ instead of z, making the use of a morphism unnecessary. □

Lemma 4. *It holds that* $MAT_{ac} \subseteq GCHCD$ *and* $MAT_{ac}^{\lambda} \subseteq GCHCD^{\lambda}$.

Proof. Consider a matrix grammar with appearance checking

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

Denote $M = \{m_1, m_2, \ldots, m_m\}$. Moreover, assume that any matrix m_i, with $1 \leq i \leq m$, contains at most one production in F; it is known (cf., e.g., [25]) that this is a normal form for matrix grammars.

To simulate this matrix grammar with appearance checking, construct the following graph controlled hybrid CD grammar system

$$\Gamma' = (N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n), U),$$

such that $(P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n)$ contains the below components, for every matrix $m_i : (\alpha_{i_1} \to \beta_{i_1}, \alpha_{i_2} \to \beta_{i_2}, \ldots, \alpha_{i_{\ell_i}} \to \beta_{i_{\ell_i}}) \in M$, with $1 \leq i \leq m$, and U is as below.

If m_i contains no production which is in F, then for all $1 \leq j \leq \ell_i$, we add a component

$$(P_{i_j}, {=}1), \text{ where } P_{i_j} = \{\alpha_{i_j} \to \beta_{i_j}\}.$$

Else, let $\alpha_{i_t} \to \beta_{i_t}$, for some $1 \leq t \leq \ell_i$, be the production of m_i which is in F. Then we add a component

$$(P'_{i_t}, t), \text{ where } P'_{i_t} = \{\alpha_{i_t} \to \alpha_{i_t}\}.$$

Moreover, $U = (V, E)$, where U and V are as follows, for every matrix $m_i : (\alpha_{i_1} \to \beta_{i_1}, \alpha_{i_2} \to \beta_{i_2}, \ldots, \alpha_{i_{\ell_i}} \to \beta_{i_{\ell_i}}) \in M$, with $1 \leq i \leq m$:

$$\begin{aligned}
V &= \{\, P_{i_j} \mid 1 \leq j \leq \ell_i,\ 1 \leq i \leq m \,\} \cup \{\, P'_{i_t} \mid 1 \leq t \leq \ell_i,\ 1 \leq i \leq m \,\} \text{ and} \\
E &= \{\, (P_{i_j}, P_{i_{j+1}}) \mid 1 \leq j \leq \ell_i,\ 1 \leq i \leq m \,\} \cup \{\, (P_{i_{\ell_i}}, P_{k_1}) \mid 1 \leq i, k \leq m \,\} \cup \\
&\qquad \{\, (P_{i_{t-1}}, P'_{i_t}),\ (P'_{i_t}, P_{i_{t+1}}) \mid 1 \leq t \leq \ell_i,\ 1 \leq i \leq m \,\}.
\end{aligned}$$

For every production in matrices of M, a component P_{i_j} is constructed operating in mode $=1$. Additionally, for every production in F, an 'appearance checking' component P'_{i_t} is constructed operating in mode t. The strict ordering of productions in matrices of M is preserved by the graph. After all productions of a matrix have been used, a new matrix can be started by beginning with its first production. This is all imposed by the graph. Since the productions are put in different components following each other, it is clear that a production can rewrite nonterminals introduced by a production from the same matrix which precedes it in the ordering.

The appearance checking works as follows. First assume that α_{i_t} occurs in the sentential form. Then the production $\alpha_{i_t} \to \beta_{i_t}$ of component P_{i_t} can be applied and matrix simulation can continue with component $P_{i_{t+1}}$. However, P'_{i_t} works in the t-mode and since α_{i_t} is present, this component will continue working without ever obtaining a terminal string.

Next assume that α_{i_t} does not occur in the sentential form. Then the derivation cannot continue by applying component P_{i_t}, since it works in mode $=1$, which is impossible. However, component P'_{i_t} works in the t-mode, meaning that it can be applied zero times if the left-hand side of its production does not occur in the sentential form, and matrix simulation can continue with component $P_{i_{t+1}}$. This shows that the appearance checking case is simulated correctly.

It is thus clear that $L(\Gamma') = L(G)$ and that both statements of the lemma are now proved. $\square$

To put these results in perspective, the following results now hold, for $f \in \{\, {\leq}k, {=}k, {\geq}k \mid k \geq 1 \,\} \cup \{*, t\}$ and $k \geq 1$:

$$\begin{tikzcd}
CS \arrow[r, phantom, "="] \arrow[d, phantom, "\cup"] & HLCD(f) \arrow[r, phantom, "="] & GSMCD(f) & \\
MAT_{ac} \arrow[r, phantom, "="] \arrow[d, phantom, "\cup"] & GCHCD \arrow[r, phantom, "\supseteq"] & HCD \arrow[dr, phantom, sloped, "\supset"] & \\
MAT \arrow[r, phantom, "="] \arrow[d, phantom, "\cup"] & GCCD_{[REG]}(=k) & & ET0L = GCCD_{[REG]}(t) = CD(t) \arrow[dl, phantom, sloped, "\supset"] \\
CF \arrow[r, phantom, "="] & GCCD(*) \arrow[r, phantom, "="] & CD(*) &
\end{tikzcd}$$

Families which are not connected are not necessarily incomparable. The reported statements can be found in Sects. 2–5 and in Lemma 3 and 4, or are obvious.

Note that hybridity strictly increases the generative power of graph controlled CD grammar systems, whereas it remains an open problem whether control by a graph strictly increases the generative power of hybrid CD grammar systems. A solution to this problem could shed light on the relation between hybrid CD grammar systems and matrix grammars, or perhaps even solve this open problem first stated in [43].

6.2 Hybrid CD Grammar Systems with Dynamic Control

Next the case of a hypothesis (or target) language or a gsm as *dynamic* control mechanism is considered. This kind of control is called dynamic, since the current state of the problem is taken into consideration by assuming a hypothesis (target) language with which the sentential forms are compared during derivation, or a gsm that translates sentential forms during derivation.

We first define hybrid CD grammar systems with a (regular) hypothesis language. The notion of a hypothesis language (or the slightly different concept of regular restriction) to compare sentential forms with during derivation has already been introduced in [23,32,44] for context-free grammars, CD grammar systems and colonies, which is a subclass of CD grammar systems with components generating finite languages (cf. [39] for a result on context-free grammars with a weak regular restriction).

Definition 6. *A* hybrid CD grammar system with a (regular) hypothesis language *is a construct*

$$\Gamma = (N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n), R),$$

where $(N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n))$ *is a hybrid CD grammar system and* R *is a regular language in* $(N \cup T)^* \setminus T^*$*.*

A derivation consists of accepted derivation steps, where a derivation step $x \Longrightarrow_{P_i}^{f_i} y$*, for* $f_i \in \{\leq k, = k, \geq k \mid k \geq 1\} \cup \{*, t\}$ *and* $1 \leq i \leq n$*, is accepted iff* $y \in R$ *or* $y \in T^*$*. The language generated by* Γ *with hypothesis language* R *is:*

$$\begin{aligned} L^R(\Gamma) = \{\, z \in T^* \mid S \Longrightarrow_{P_{i_1}}^{f_{i_1}} w_{i_1} \Longrightarrow_{P_{i_2}}^{f_{i_2}} \cdots \Longrightarrow_{P_{i_m}}^{f_{i_m}} w_{i_m} = z, \\ w_{i_k} \in R,\ 1 \leq i_j \leq n,\ 1 \leq j \leq m,\ 1 \leq k \leq m-1 \,\}. \end{aligned}$$

Note that no hypothesis is made about the final terminal string. The regularity of the hypothesis language is motivated by the fact that the test for the condition $y \in R$ can be done in linear time.

The family of languages that is generated by hybrid CD grammar systems with a hypothesis language $(N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_m, f_m), R)$ with only λ-free context-free productions of type 2 in each P_j and $1 \leq j \leq m \leq n$, is denoted by $HLHCD_n$. Denote $HLHCD = \bigcup_{n \geq 1} HLHCD_n$.

We now define hybrid CD grammar systems with a gsm. The notion of a gsm that translates sentential forms during derivation has already been introduced in [41,44] for CD grammar systems and colonies.

Definition 7. *A* hybrid CD grammar system with a gsm *is a construct*

$$\Gamma = (N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n), g),$$

where $(N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n))$ *is a hybrid CD grammar system and*

$$g = (K, N \cup T, N \cup T, s_0, \delta, H)$$

is a gsm.

For $f_i \in \{ \leq k, {=}k, \geq k \mid k \geq 1 \} \cup \{*, t\}$ *and* $1 \leq i \leq n$*, the language generated by* Γ *with gsm* g *is:*

$$L^g(\Gamma) = \{ z \in T^* \mid S \Longrightarrow^{f_{i_1}}_{P_{i_1}} w_{i_1} \Longrightarrow g(w_{i_1}) \Longrightarrow^{f_{i_2}}_{P_{i_2}} w_{i_2} \Longrightarrow g(w_{i_2}) \Longrightarrow^{f_{i_3}}_{P_{i_3}} \cdots$$
$$\cdots \Longrightarrow^{f_{i_m}}_{P_{i_m}} w_{i_m} = z,\ 1 \leq i_j \leq n,\ 1 \leq j \leq m \}$$

The family of languages that is generated by hybrid CD grammar systems with a gsm $(N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_m, f_m), g)$ with only λ-free context-free productions of type 2 in each P_j and $1 \leq j \leq m \leq n$, is denoted by $GSMHCD_n$. Denote $GSMHCD = \bigcup_{n \geq 1} GSMHCD_n$.

The next example illustrates hybrid CD grammar systems with a gsm.

Example 4. Consider the hybrid CD grammar system with a gsm

$$\Gamma_4 = (\{S, A, B, C, D\}, \{a, b, c, d\}, S, (P_1, {=}1), (P_2, {=}2), (P_3, {=}2), (P_4, {=}4), g),$$

where

$$\begin{aligned}
P_1 &= \{S \to abcd\},\\
P_2 &= \{A \to aa, C \to cc\},\\
P_3 &= \{B \to bb, D \to dd\},\\
P_4 &= \{A \to a, B \to b, C \to c, D \to d\} \text{ and}\\
g &= (\{s_a, s_b, s_c, s_d, s_z\}, I, O, s_a, \delta, \{s_z\}),
\end{aligned}$$

with $I = O = \{S, A, B, C, D, a, b, c, d\}$ and

$$\begin{aligned}
\delta = \{&s_a a \to A s_b, s_a A \to A s_b, s_b a \to a s_b, s_b b \to B s_c, s_b B \to B s_c,\\
&s_c b \to b s_c, s_c c \to C s_d, s_c C \to C s_d, s_d c \to c s_d, s_d d \to D s_z,\\
&s_d D \to D s_z, s_z d \to d s_z\}.
\end{aligned}$$

Initially, only P_1 can be used, resulting in the sentential form $abcd$. This component can now never be used again. The gsm translates this string into $ABCD$. The gsm always translates the first a, b, c and d that it meets into A, B, C and D, respectively, if this particular nonterminal is not yet present in the sentential form, meanwhile skipping specific intermediate terminals and nonterminals. From $ABCD$, the derivation can be continued by using either P_2, P_3 or P_4.

Component P_4 results in the terminal string $abcd$. Using P_2 (P_3) leads to $aaBccD$ ($AbbCdd$), which the gsm thus translates into $AaBCcD$ ($ABbCDd$). Repeating this process, strings of the form $Aa^ib^jCc^id^j$ or $a^iBb^jc^iDd^j$ are translated into strings of the form $Aa^iBb^jCc^iDd^j$, with $i \geq 1$ and $j \geq 0$ ($i \geq 0$ and $j \geq 1$). Finally, component P_4 can then be used (only after translation since it requires the presence of A, B, C and D in the sentential form) to obtain terminal strings of the form $a^ib^jc^id^j$, with $i \geq 1$ and $j \geq 1$. It is thus clear that

$$L^g(\Gamma_4) = \{\, a^nb^mc^nd^m \mid m, n \geq 1 \,\}.$$

Note that the gsm in the example is in fact a Mealy machine since it is deterministic and every production in the set δ has only one output letter. Hence already regular productions and a restricted gsm suffice to generate a non-context-free language.

The following lemma leads to a result (cf. Theorem 1) that is not limited to hybrid CD grammar systems with control by a gsm, but that also holds for control by a hypothesis language.

Lemma 5. *It holds that* $GSMHCD \subseteq CS$.

Proof. Consider a hybrid CD grammar system with a gsm

$$\Gamma = (N, T, S, (P_1, f_1), (P_2, f_2), \ldots, (P_n, f_n), g).$$

Furthermore, let

$$g = (S, I, O, s_0, \delta, F), \text{ where } I = O = (N \cup T).$$

To simulate this hybrid CD grammar system with a gsm, construct the following Chomsky grammar of type 1.

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

where

$$\begin{aligned} N' = {} & N \cup \{S', T, T', s_0\} \cup \{\, [C_i, f_i, j],\ [C'_i, f_i, j],\ [C''_i, f_i, j] \mid (P_i, f_i) \in \Gamma, \\ & \quad f_i \in \{\, {\leq}k, {=}k, {\geq}k \mid k \geq 1 \,\},\ 1 \leq i \leq n,\ 0 \leq j \leq k \,\} \cup \\ & \{\, [C_i, g_i],\ [C'_i, g_i],\ [C''_i, g_i] \mid (P_i, g_i) \in \Gamma,\ g_i \in \{*, t\},\ 1 \leq i \leq n \,\} \cup \\ & \{\, C_{t_i}, C'_{t_i} \mid (P_i, t) \in \Gamma,\ 1 \leq i \leq n \,\}, \\ T' = {} & T \cup \{L, R\} \text{ and} \end{aligned}$$

$$
\begin{aligned}
P' = \{&S' \to \mathit{LTSR}\} \cup \\
&\{\, LT \to L[C_i, f_i, 0],\ LT \to L[C_i, g_i] \mid f_i \in \{\, {\le}k, {=}k, {\ge}k \mid k \ge 1\,\}, \\
&\quad g_i \in \{*, t\},\ 1 \le i \le n\,\} \cup \\
&\{\, [C_i, f_i, j]y \to y[C_i, f_i, j],\ [C'_i, f_i, j]y \to y[C'_i, f_i, j], \\
&\quad y[C''_i, f_i, j] \to [C''_i, f_i, j]y \mid y \in (N \cup T),\ f_i \in \{\, {\le}k, {=}k, {\ge}k \mid k \ge 1\,\}, \\
&\quad 0 \le j \le k,\ 1 \le i \le n\,\} \cup \\
&\{\, [C_i, g_i]y \to y[C_i, g_i],\ [C'_i, g_i]y \to y[C'_i, g_i],\ y[C''_i, g_i] \to [C''_i, g_i]y \mid \\
&\quad y \in (N \cup T),\ g_i \in \{*, t\},\ 1 \le i \le n\,\} \cup \\
&\{\, [C_i, f_i, j]A \to x[C'_i, f_i, j] \mid A \to x \in P_i,\ f_i \in \{\, {\le}k, {=}k, {\ge}k \mid k \ge 1\,\}, \\
&\quad 0 \le j \le k,\ 1 \le i \le n\,\} \cup \\
&\{\, [C_i, g_i]A \to x[C'_i, g_i] \mid A \to x \in P_i,\ g_i \in \{*, t\},\ 1 \le i \le n\,\} \cup \\
&\{\, [C_i, f_i, j]R \to [C''_i, f_i, j]R \mid f_i \in \{\, {\le}k, {=}k, {\ge}k \mid k \ge 1\,\},\ 0 \le j \le k, \\
&\quad 1 \le i \le n\,\} \cup \\
&\{\, [C_i, g_i]R \to [C''_i, g_i]R \mid g_i \in \{*, t\},\ 1 \le i \le n\,\} \cup \\
&\{\, L[C''_i, f_i, j] \to L[C_i, f_i, j+1] \mid f_i \in \{\, {\le}k, {=}k, {\ge}k \mid k \ge 1\,\}, \\
&\quad 0 \le j \le k-1,\ 1 \le i \le n\,\} \cup \\
&\{\, L[C''_i, {\le}k, j] \to Ls_0 \mid 0 \le j \le k,\ 1 \le i \le n\,\} \cup \\
&\{\, L[C''_i, {=}k, k] \to Ls_0 \mid 1 \le i \le n\,\} \cup \\
&\{\, L[C''_i, {\ge}k, k] \to L[C_i, {\ge}k, k],\ L[C''_i, {\ge}k, k] \to Ls_0 \mid 1 \le i \le n\,\} \cup \\
&\{\, L[C''_i, *] \to L[C_i, *],\ L[C''_i, *] \to Ls_0 \mid 1 \le i \le n\,\} \cup \\
&\{\, L[C_i, *] \to Ls_0 \mid 1 \le i \le n\,\} \cup \\
&\{\, L[C_i, t] \to LC_{t_i},\ L[C''_i, t] \to LC_{t_i} \mid 1 \le i \le n\,\} \cup \\
&\{\, C_{t_i}y \to yC_{t_i} \mid y \to x \notin P_i,\ y \in (N \cup T)^+,\ x \in (N \cup T)^*,\ 1 \le i \le n\,\} \cup \\
&\{\, C_{t_i}R \to C'_{t_i}R,\ yC'_{t_i} \to C'_{t_i}y,\ LC'_{t_i} \to Ls_0 \mid y \in (N \cup T),\ 1 \le i \le n\,\} \cup \\
&\{\, s_1x \to ys_2 \mid (s_1x, ys_2) \in \delta\,\} \cup \{\, sR \to TR \mid s \in F\,\} \cup \\
&\{\, yT \to Ty \mid y \in O\,\} \cup \{LT \to LT'\} \cup \\
&\{\, T'y \to yT' \mid y \in T\} \cup \{T'R \to R\,\}.
\end{aligned}
$$

Note that G contains non-contracting productions; it is known (cf., e.g., [48,50]) that these can be transformed into context-sensitive productions.

The simulation starts by introducing the sentential form LTSR, where S is the original axiom, T is a marker and L and R are terminal symbols indicating the left and right end, respectively, of the sentential form. From S, the language of the hybrid CD grammar system with a gsm will be generated and the marker is non-deterministically replaced by a control symbol $[C_i, f_i, j]$ or $[C_i, g_i]$ indicating the use of a component P_i working in mode $f_i \in \{\, \le k, = k, \ge k \mid k \ge 1\,\}$ or $g_i \in \{*, t\}$, respectively. In the case of a mode $f_i \in \{\, \le k, = k, \ge k \mid k \ge 1\,\}$, a counter j, with $0 \le j \le k$, is used; in the case of mode $*$ or t this is not necessary.

The simulation continues by moving the control symbol to the right (skipping terminals and nonterminals) until a nonterminal is replaced by a production from the component P_i, consequently priming the control symbol. Then this primed version of the control symbol is moved completely to the right of the sentential from, where it becomes double primed and is moved completely to the left again. When it is completely on the left of the sentential form, some different cases need to be considered.

In the case of mode $\leq k$, $=k$ or $\geq k$ (for a $k \geq 1$) the counter is increased by one and the process is repeated. When exactly k productions of the component P_i are used (when j reaches the value k) and the mode is $=k$, the control symbol is replaced by s_0. In the case of mode $\leq k$, this can happen for every value of j between zero and k. For mode $\geq k$ this production can be used when j is equal to k, but also another process guided by the initial control symbol with counter k can be started. This allows the use of a production more than k times, indeed corresponding to mode $\geq k$, before replacing the control symbol by s_0.

In the case of mode $*$, the control symbol can be replaced by s_0 after using P_i an arbitrary number of times indeed. The same holds for mode t, except that before introducing s_0 a test is done to check if there is indeed no production left from P_i that can be used on the current sentential form. For this test, a test symbol C_{t_i} is introduced, indicating for which component (P_i) this test is done. This test symbol is moved from left to right over the sentential form, allowed to skip any terminals but only those nonterminals for which there is no production in component P_i being tested. When it reaches the right end, it is replaced by its primed version which is then moved completely to the left to be replaced by s_0.

In any mode, the result is the introduction of s_0 to indicate the end of the work of the component. This s_0 is the start symbol of the gsm. Next, the usage of the gsm is simulated on the sentential form remaining after using a component. It does its work as usual and when it reaches the right side of the sentential form in a final state, this final state symbol is replaced by the marker T again. This T is moved to the left, skipping only symbols from the output alphabet of the gsm, where it is replaced by T'. Finally, this T' is moved completely to the right skipping only terminals before it disappears.

From this detailed explanation it is clear that $L(G') = \{L\}L(\Gamma)\{R\}$ and, since it is known (cf., e.g., [50]) that the family CS is closed under cancellation of first and last letter, $L(G') \in CS$, and the lemma is thus proved. □

This lemma leads to the final result of this paper, presented in below theorem.

Theorem 1. *For* $f \in \{\leq k, =k, \geq k \mid k \geq 1\} \cup \{*, t\}$ *holds that:*

$$CS = HLCD(f) = HLHCD = GSMHCD.$$

Proof. The first equality can be found in Lemma 1(2). Furthermore, the inclusion $HLCD(f) \subseteq HLHCD$ is obvious. It is also clear that a gsm can check whether a given input string is in a regular language; it can thus play the role of a hypothesis language and hence $HLHCD \subseteq GSMHCD$ holds. Finally, Lemma 5 finishes the proof of this theorem. □

7 Results

To put the results of this paper in a proper perspective, the following diagram now holds, for $f \in \{\leq k, = k, \geq k \mid k \geq 1\} \cup \{*, t\}$ and $k \geq 1$, where families which are not connected are not necessarily incomparable and $[REG]$ means that the result holds with or without the restriction to regular productions of type 3:

$$
\begin{array}{rclclclclcl}
RE & = & MAT^{\lambda}_{ac} & & & & & & & & \\
\cup & & & & & & & & & & \\
CS & = & HLCD(f) & = & GSMCD(f) & = & HLHCD & = & GSMHCD & & \\
\cup & & & & & & & & & & \\
MAT_{ac} & = & GCHCD & \supseteq & HCD & & & & & & \\
\cup & & & & & \rotatebox{-45}{$\supseteq$} & & & & & \\
MAT & = & GCCD_{[REG]}(=k) & & & & ET0L & = & CD(t) & = & GCCD_{[REG]}(t) \\
\cup & & & & & & & & \cup & & \\
CF & = & CD(=1) & = & CD(\geq 1) & = & CD(\leq k) & = & CD(*) & = & GCCD(*) \\
\cup & & & & & & & & & & \\
MLIN & = & CD_{MLIN}(f) & = & HCD_{MLIN} & & & & & & \\
\cup & & & & & & & & & & \\
LIN & = & CD_{LIN}(f) & = & HCD_{LIN} & & & & & & \\
\cup & & & & & & & & & & \\
REG & = & CD_{REG}(f) & = & HCD_{REG} & & & & & &
\end{array}
$$

8 Conclusion

We have characterised the generative power of several types of hybrid CD grammar systems with controlled derivations, by relating them to the families of languages generated by matrix grammars with appearance checking and by context-sensitive grammars. In particular, due to the new Lemma 4, we now know that hybridity strictly increases the generative power of graph controlled CD grammar systems. However, it remains an open problem whether control by a graph strictly increases the generative power of hybrid CD grammar systems.

Moreover, we have shown that such formal languages can generate the non-context-free constructs known as multiple agreements, crossed agreements and (re)duplication, which linguists have identified as features of natural languages. In particular, graph controlled hybrid CD grammar systems can generate these non-context-free features. This is important, since their generative power is shown to be strictly less than that of context-sensitive grammars, which are known to be too computationally expensive to parse.

Acknowledgement. This paper is dedicated to Erzsébet Csuhaj-Varjú (Erzsi for friends like me) who has taught me everything I always wanted to know about CD grammar systems exactly 30 years ago when Grzegorz Rozenberg and Jetty Kleijn put me in contact with her. She was an excellent supervisor of my Master's thesis on teams in grammar systems [3], which has resulted in this paper and in three earlier papers

[4–6]. The concept of teamwork has been an important part of my research ever since. We continued to collaborate on CD grammar systems also while I was working on my PhD thesis on team automata [7], which have become a successful area of research [13,14]. This collaboration resulted in teams of pushdown automata [11,12] as well as in a series of papers in which we teamed up to study notions of competence in CD grammar systems [8–10]. For a special journal issue celebrating an earlier birthday of Erzsi, Jetty and I transferred the team automata concept of synchronised collaboration to teams of grammars [16]. For this Festschrift, I decided to return to the roots and report on specific types of CD grammar systems. Who knows what's next ...

I would also like to thank the anonymous reviewers for their useful comments and suggestions. In particular, the proof that the statement of Lemma 4 holds for matrix grammars with appearance checking, and not just for matrix grammars without appearance checking as proved in [3], is due to one of the reviewers.

Disclosure of Interests. The author has no competing interests to declare that are relevant to the content of this article.

References

1. Abraham, S.: Some questions of phrase-structure grammars, I. Comput. Linguist. **4**, 61–70 (1965). https://doi.org/10.1515/ling.1967.5.31.5
2. Becerra-Bonache, L., Bel-Enguix, G., Jiménez-López, M.D., Martín-Vide, C.: Mathematical foundations: formal grammars and languages. In: Mitkov, R. (ed.) The Oxford Handbook of Computational Linguistics, Chap. 9, pp. 207–229. Oxford University Press (2003). https://doi.org/10.1093/oxfordhb/9780199573691.013.021
3. ter Beek, M.H.: Teams in grammar systems. Master's thesis, Leiden University, September 1996. https://theses.liacs.nl/716
4. ter Beek, M.H.: Teams in grammar systems: hybridity and weak rewriting. Acta Cybern. **12**(4), 425–444 (1996). https://cyber.bibl.u-szeged.hu/index.php/actcybern/article/view/3473
5. ter Beek, M.H.: Teams in grammar systems: sub-context-free cases. In: Păun, G., Salomaa, A. (eds.) New Trends in Formal Languages. LNCS, vol. 1218, pp. 197–216. Springer, Heidelberg (1997). https://doi.org/10.1007/3-540-62844-4_13
6. ter Beek, M.H.: Simple eco-grammar systems with prescribed teams. In: Păun, G., Salomaa, A. (eds.) Grammatical Models of Multi-Agent Systems. Topics in Computer Mathematics, vol. 8, pp. 113–135. Gordon and Breach, London (1999)
7. ter Beek, M.H.: Team automata: a formal approach to the modeling of collaboration between system components. Ph.D. thesis, Leiden University (2003). https://doi.org/https://hdl.handle.net/1887/29570
8. ter Beek, M.H., Csuhaj-Varjú, E., Holzer, M., Vaszil, G.: On competence in CD grammar systems. In: Calude, C.S., Calude, E., Dinneen, M.J. (eds.) DLT 2004. LNCS, vol. 3340, pp. 76–88. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30550-7_7
9. ter Beek, M.H., Csuhaj-Varjú, E., Holzer, M., Vaszil, G.: On competence in CD grammar systems with parallel rewriting. Int. J. Found. Comput. Sci. **18**(6), 1425–1439 (2007). https://doi.org/10.1142/S0129054107005467
10. ter Beek, M.H., Csuhaj-Varjú, E., Holzer, M., Vaszil, G.: Cooperating distributed grammar systems: components with nonincreasing competence. In: Kelemen, J., Kelemenová, A. (eds.) Computation, Cooperation, and Life. LNCS, vol. 6610, pp. 70–89. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20000-7_7

11. ter Beek, M.H., Csuhaj-Varjú, E., Mitrana, V.: Teams of pushdown automata. In: Broy, M., Zamulin, A.V. (eds.) PSI 2003. LNCS, vol. 2890, pp. 329–337. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-39866-0_32
12. ter Beek, M.H., Csuhaj-Varjú, E., Mitrana, V.: Teams of pushdown automata. Int. J. Comput. Math. **81**(2), 141–156 (2004). https://doi.org/10.1080/00207160310001650099
13. ter Beek, M.H., Hennicker, R., Proença, J.: Team automata: overview and roadmap. In: Castellani, I., Tiezzi, F. (eds.) COORDINATION 2024. LNCS, vol. 14676, pp. 161–198. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-62697-5_10
14. ter Beek, M.H., Hennicker, R., Proença, J.: Overview and roadmap of team automata. Log. Methods Comput. Sci. (2025)
15. ter Beek, M., Kleijn, J.: Petri net control for grammar systems. In: Brauer, W., Ehrig, H., Karhumäki, J., Salomaa, A. (eds.) Formal and Natural Computing. LNCS, vol. 2300, pp. 220–243. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45711-9_13
16. ter Beek, M.H., Kleijn, J.: On distributed cooperation and synchronised collaboration. J. Autom. Lang. Comb. **19**(1-4), 17–32 (2014). https://doi.org/10.25596/jalc-2014-017
17. Bresnan, J., Kaplan, R.M., Peters, S., Zaenen, A.: Cross-serial dependencies in Dutch. Linguist. Inq. **13**(4), 613–635 (1982). https://doi.org/10.1007/978-94-009-3401-6_11
18. Chomsky, N.: On certain formal properties of grammars. Inf. Control **2**(2), 137–167 (1959). https://doi.org/10.1016/S0019-9958(59)90362-6
19. Csuhaj-Varjú, E., Dassow, J.: On cooperating/distributed grammar systems. J. Inf. Process. Cybern. **26**(1–2), 49–63 (1990)
20. Csuhaj-Varjú, E., Dassow, J., Kelemen, J., Păun, G. (eds.): Grammar Systems: A Grammatical Approach to Distribution and Cooperation, Topics in Computer Mathematics, vol. 5. Gordon and Breach, London (1994)
21. Csuhaj-Varjú, E., Kelemen, J.: Cooperating grammar systems: a syntactical framework for the blackboard model of problem solving. In: Plander, I. (ed.) Proceedings of the 5th International Conference on Artificial Intelligence and Information-Control Systems of Robots (AIICSR 1989), North-Holland, pp. 121–127 (1989)
22. Culy, C.: The complexity of the vocabulary of Bambara. Linguist. Philos. **8**, 345–351 (1984). https://doi.org/10.1007/978-94-009-3401-6_14
23. Dassow, J.: Cooperating/distributed grammar systems with hypothesis languages. J. Exp. Theor. Artif. Intell. **3**(1), 11–16 (1991). https://doi.org/10.1080/09528139108915278
24. Dassow, J.: A remark on cooperating distributed grammar systems controlled by graphs. Wiss. Zeit. T.U. Magdeburg **35**, 4–6 (1991)
25. Dassow, J., Păun, G.: Regulated Rewriting in Formal Language Theory. EATCS Monographs on Theoretical Computer Science, vol. 18. Springer, Berlin (1989). https://www.springer.com/de/book/9783642749346
26. Dassow, J., Păun, G., Rozenberg, G.: Grammar systems. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, Volume 2: Linear Modeling: Background and Application, pp. 155–213. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-662-07675-0_4
27. Dassow, J., Păun, G., Salomaa, A.: Grammars with controlled derivations. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, Volume 2: Linear Modeling: Background and Application, pp. 101–154. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-662-07675-0_3

28. Fernau, H., Holzer, M.: Graph-controlled cooperating distributed grammar systems with singleton components. J. Autom. Lang. Comb. **7**(4), 487–503 (2002). https://doi.org/10.25596/JALC-2002-487
29. Fernau, H., Holzer, M., Freund, R.: Bounding resources in cooperating distributed grammar systems. In: Bozapalidis, S. (ed.) Proceedings of the 3rd International Conference on Developments in Language Theory (DLT 1997), pp. 261–272. Aristotle University of Thessaloniki (1997)
30. Fernau, H., Holzer, M., Freund, R.: Hybrid modes in cooperating distributed grammar systems: internal versus external hybridization. Theor. Comput. Sci. **259**(1–2), 405–426 (2001). https://doi.org/10.1016/S0304-3975(00)00022-0
31. Fernau, H., Holzer, M., Freund, R.: Hybrid modes in cooperating distributed grammar systems: combining the t-mode with the modes $\leq k$ and $= k$. Theor. Comput. Sci. **299**(1–3), 633–662 (2003). https://doi.org/10.1016/S0304-3975(02)00541-8
32. Friš, I.: Grammars with partial ordering of the rules. Inf. Control. **12**(5-6), 415–425 (1968). https://doi.org/10.1016/S0019-9958(68)90439-7. Errata in Inf. Control. **15**(5), 452–453 (1969). https://doi.org/10.1016/S0019-9958(69)90521-X
33. Gazdar, G., Pullum, G.K.: Computationally relevant properties of natural languages and their grammars. New Gener. Comput. **3**, 273–306 (1985). https://doi.org/10.1007/978-94-009-3401-6_17
34. Gheorghe, M., Păun, G.: Further remarks on cooperating/distributed grammar systems. Bull. Math. Soc. Sci. Math. Roumanie **34(82)**(3), 231–245 (1990). http://www.jstor.org/stable/43678342
35. Higginbotham, J.: English is not a context-free language. Linguist. Inq. **15**(2), 225–234 (1984). https://doi.org/10.1007/978-94-009-3401-6_13
36. Huijbregts, R.: The weak inadequacy of context-free phrase structure grammars. In: de Haan, G.J., Trommele, M., Zonneveld, W. (eds.) Van Periferie Naar Kern, pp. 81–99. Foris (1984)
37. Kari, L., Mateescu, A., Păun, G., Salomaa, A.: Teams in cooperating grammar systems. J. Exp. Theor. Artif. Intell. **7**(4), 347–359 (1995). https://doi.org/10.1080/09528139508953816
38. Kari, L., Rozenberg, G., Salomaa, A.: L systems. In: Rozenberg, G., Salomaa, A. (eds.) Handbook of Formal Languages, Volume 1: Word, Language, Grammar, pp. 253–328. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-642-59136-5_5
39. Král, J.: A note on grammars with regular restrictions. Kybernetika **9**(3), 159–161 (1973). http://www.kybernetika.cz/content/1973/3/159
40. Meersman, R., Rozenberg, G.: Cooperating grammar systems. In: Winkowski, J. (ed.) MFCS 1978. LNCS, vol. 64, pp. 364–373. Springer, Heidelberg (1978). https://doi.org/10.1007/3-540-08921-7_84
41. Mitrana, V.: Pairs grammar systems - transducers. Ann. Univ. Buc. Ser. Matem.-Inform. **39**, 73–81 (1990)
42. Mitrana, V.: Hybrid cooperating distributed grammar systems. Comput. Artif. Intell. **2**, 83–88 (1993)
43. Păun, G.: On the generative capacity of hybrid CD grammar systems. J. Inf. Process. Cybern. **30**(4), 231–244 (1994)
44. Păun, G.: On the generative capacity of colonies. Kybernetika **31**(1), 83–97 (1995). http://www.kybernetika.cz/content/1995/1/83
45. Păun, G., Rozenberg, G.: Prescribed teams of grammars. Acta Inform. **31**(6), 525–537 (1994). https://doi.org/10.1007/BF01213205
46. Radzinski, D.: Unbounded syntactic copying in Mandarin Chinese. Linguist. Philos. **13**, 113–127 (1990). https://doi.org/10.1007/BF00630518

47. Rozenberg, G., Salomaa, A.: The Mathematical Theory of L Systems. Academic Press, New York (1980)
48. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Volume 1: Word, Language, Grammar. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-642-59136-5
49. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Volume 2: Linear Modeling: Background and Application. Springer, Heidelberg (1997). https://doi.org/10.1007/978-3-662-07675-0
50. Salomaa, A.: Formal Languages. Academic Press, New York (1973)
51. Shieber, S.M.: Evidence against the context-freeness of natural language. Linguist. Philos. **8**(4), 333–343 (1985). https://doi.org/10.1007/978-94-009-3401-6_12
52. Wood, D.: Bicolored digraph grammar systems. RAIRO **R-1**, 45–52 (1973). http://www.numdam.org/item/M2AN_1973__7_1_45_0