Comparison of Two Context-Free Rewriting Systems with Simple Context-Checking Mechanisms

Tomáš Masopust
Institute of Mathematics of the Czech Academy of Sciences
Žižkova 22, 616 62 Brno, Czech Republic
masopust@ipm.cz

Abstract

This paper solves an open problem concerning the generative power of nonerasing context-free rewriting systems using a simple mechanism for checking for context dependencies, in the literature known as semi-conditional grammars of degree $(1,1)$. In these grammars, two nonterminal symbols are attached to each context-free production, and such a production is applicable if one of the two attached symbols occurs in the current sentential form, while the other does not. Specifically, this paper demonstrates that the family of languages generated by semi-conditional grammars of degree $(1,1)$ coincides with the family of random context languages. In addition, it shows that the normal form proved by Mayer for random context grammars with erasing productions holds for random context grammars without erasing productions, too. It also discusses two possible definitions of the relation of the direct derivation step used in the literature.

1 Introduction

It is well known that context-free grammars play an important role in form language theory from both practical and theoretical point of view. However, some kinds of context dependencies are required in many practical applications, such as the analysis of programming and natural languages, which, therefore, cannot be handled by context-free grammars. For that reason, some more powerful rewriting mechanisms that generate convenient proper subfamilies of the family of context sensitive languages and that make use of advantages of the simple form of context-free productions are of interest.

This paper discusses two such rewriting mechanisms based on context-free productions. Specifically, it discusses random context grammars and their special and more simple variant, semi-conditional grammars of degree $(1,1)$. In comparison with context-free grammars where erasing productions can be eliminated without affecting the generative power, erasing productions play a significant role in random context grammars and semi-conditional grammars of degree $(1,1)$. Specifically, with them both these rewriting mechanisms characterize the family of recursively enumerable languages (see [4] and [13], respectively), while without them they are less powerful then context sensitive grammars (see [4] and [14], respectively). As the erasing cases of random context grammars and semi-conditional grammars of degree $(1,1)$ have been studied carefully, this paper concentrates its attention on the nonerasing variants of these grammars.

A *random context grammar*, introduced by van der Walt [16] in 1970, is a context-free grammar the productions of which are applicable to a sentential form only if some of the nonterminal symbols occur in the sentential form, while some others do not. Specifically, two finite sets of nonterminal symbols—a *permitting* and a *forbidding* set—are attached to each production, and such a production is applicable to a sentential form if all permitting symbols occur in that sentential form, while no forbidding symbol does. It is well known (see [1, 4]) that the family of languages generated by random context grammars is properly included in the family of context sensitive languages, and, in addition, that the elimination of either all permitting or all forbidding sets makes them less powerful (see [1, 5, 17]).

In 1985, Păun [14] introduced *semi-conditional grammars* as a variant of random context grammars, where permitting and forbidding sets are replaced with permitting and forbidding strings. According to the length of these strings, semi-conditional grammars of degree (i,j), for $i,j \geq 0$, are defined. It is proved in [14] that for any $i,j \geq 0$, the family of languages generated by semi-conditional grammars of degree (i,j) contains the family of context-free languages and, in addition, is included in the family of context sensitive languages. Furthermore, semi-conditional grammars of degree (i,j), where $1 \leq i,j \leq 2$, $i \neq j$, are powerful enough to characterize the family of context sensitive languages. On the other hand, however, the precise generative power of semi-conditional grammars of degree $(1,1)$ was left open.

This paper solves this problem so that it demonstrates that semi-conditional grammars of degree $(1,1)$ characterize the family of random context languages. As a consequence, it presents a normal form for random context grammars without erasing productions similar to the normal form for random context grammars with erasing productions proved by Mayer in [13], who left the question of whether this normal form also holds for random context grammars without erasing productions open. Two possible definitions of the relation of the direct derivation step used in the literature are also discussed.

A semi-conditional grammar G is called *simple* if for each production, either its permitting or its forbidding set is empty. It is proved in [7] that for every semi-conditional grammar G, there is an equivalent simple semi-conditional grammar G' of the same degree such that G' is without erasing productions if and only if G is. If, in addition, G is of degree $(1,1)$, terminal symbols are not contained in either permitting or forbidding sets, and the set of productions can be decomposed into two disjoint sets according to the permitting and forbidding symbols, we have so-called *conditional context-free rewriting systems* introduced in [12]. It is known that these rewriting systems (with or without erasing productions) are as powerful as semi-conditional grammars of degree $(1,1)$ (with or without erasing productions, respectively), see [7, 12]. Thus, this paper proves that they are as powerful as random context grammars. The reader is also referred to [6] for the discussion of some additional restrictions placed on these systems.

Finally, as far as the descriptional complexity of semi-conditional grammars is concerned, the reader is referred to [10, 11, 18] for the latest results; an overview of these results is also presented in [7]. Note also that the descriptional complexity of semi-conditional grammars without erasing productions, the descriptional complexity of semi-conditional grammars of degree $(1,1)$, and the descriptional complexity of conditional context-free rewriting systems are open.

2 Preliminaries and Definitions

This paper assumes that the reader is familiar with formal language theory (see [15]). For a set A, $|A|$ denotes the cardinality of A. For an alphabet (finite nonempty set) V, V^* represents the free monoid generated by V where the unit is denoted by λ. Set $V^+ = V^* - \{\lambda\}$. For a string $w \in V^*$, let $|w|$ denote the length of w and $alph(w)$ denote the set of all symbols occurring in w. For a symbol $a \in V$, let $|w|_a$ be the number of occurrences of a in w. Let **CF**, **CS**, **REC**, and **RE** denote the families of context-free, context-sensitive, recursive, and recursively enumerable languages, respectively.

A *random context grammar* (see [16]) is a quadruple $G = (N,T,P,S)$, where N is the alphabet of nonterminals, T is the alphabet of terminals such that $N \cap T = \emptyset$, $S \in N$ is the start symbol, and P is a finite set of productions of the form $(A \to x, Per, For)$, where $A \to x$ is a context-free production, $A \in N$, $x \in V^+$ $(V = N \cup T)$, and $Per, For \subseteq N$. If for each production $(A \to x, Per, For) \in P$, $Per = \emptyset$, then G is said to be a *forbidding grammar*. Analogously, if for each production $(A \to x, Per, For) \in P$, $For = \emptyset$, then G is said to be a *permitting grammar*.

For two strings $u, v \in V^*$ and a production $(A \to x, Per, For) \in P$, the relation $uAv \Rightarrow uxv$ holds provided that

$$Per \subseteq alph(uv) \qquad \text{and} \qquad alph(uv) \cap For = \emptyset. \tag{1}$$

The language generated by G is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$, where $\Rightarrow^*$ is the reflexive and transitive closure of the relation $\Rightarrow$. A *random context language* is a language generated by a random context grammar. The families of languages generated by random context grammars, permitting grammars, and forbidding grammars are denoted by **RC**, **P**, and **F**, respectively. As usual, if there is no confusion, forbidding sets are omitted from the permitting productions; i.e., $(A \to x, Per)$ is written instead of $(A \to x, Per, \emptyset)$. Analogously in case of forbidding grammars.

A *semi-conditional grammar of degree* (i, j), for $i, j \geq 0$, is a quadruple $G = (N,T,P,S)$, where N is the alphabet of nonterminals, T is the alphabet of terminals such that $N \cap T = \emptyset$, $S \in N$ is the start symbol, and P is a finite set of productions of the form $(A \to x, Per, For)$, where $A \to x$ is a context-free production, $V = N \cup T$,

1. $Per \subseteq \bigcup_{k=1}^{i} V^k$,
2. $For \subseteq \bigcup_{k=1}^{j} V^k$,
3. $|Per|, |For| \leq 1$,

and the rewritten symbol is considered in the relation of the direct derivation step (cf. the definition (1), where the rewritten symbol is not considered). Specifically, for two strings $u, v \in V^*$ and a production $(A \to x, Per, For) \in P$, the relation $uAv \Rightarrow uxv$ holds provided that

$$Per \subseteq alph(uAv) \qquad \text{and} \qquad alph(uAv) \cap For = \emptyset. \tag{2}$$

The language generated by G is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$, where $\Rightarrow^*$ is the reflexive and transitive closure of the relation $\Rightarrow$. A *semi-conditional language of degree* (i, j) is a language generated by a semi-conditional grammar of degree (i, j). The family of languages generated by semi-conditional grammars of degree (i, j) is denoted by $\mathbf{SC}(i, j)$. As usual and for the simplicity, curly brackets are omitted from the notation and $\emptyset$ is replaced with 0; i.e., for instance, $(A \to x, p, 0)$ is written instead of $(A \to x, \{p\}, \emptyset)$.

To prove the main results of this paper, we use the notion of cooperating distributed grammar systems, which are rewriting devices composed of several components represented by grammars cooperating according to a given protocol. In this paper, the considered protocol is so-called terminal derivation mode (or t-mode, for short) that makes the component work until it can.

A *cooperating distributed* (CD) *grammar system* (see [2] for more information) is a construct $\Gamma = (N, T, P_1, P_2, \dots, P_n, S)$, for some $n \geq 1$, where N is the alphabet of nonterminals, T is the alphabet of terminals such that $N \cap T = \emptyset$, $S \in N$ is the start symbol, and $P_1, P_2, \dots P_n$ are finite sets of productions.

By *components* we understand the sets P_i and by *g-components* we understand the grammars $G_i = (N, T, P_i, S)$, for all $i = 1, 2, \dots, n$. By a CD grammar system we understand a grammar system where all g-components are context-free grammars.

A *permitting CD grammar system* (see [3]) is a CD grammar system where all g-components are permitting grammars.

For two strings $u, v \in V^*$ $(V = N \cup T)$ and a number $1 \leq k \leq n$, let the relation $u \Rightarrow_k v$ denote a derivation step made by the g-component G_k, and let $u \Rightarrow_k^t v$ be a derivation such that $u \Rightarrow_k^+ v$ and there is no $w \in V^*$ for which $v \Rightarrow_k w$, where $\Rightarrow_k^+$ denotes the transitive closure of the relation $\Rightarrow_k$. The language generated by a CD grammar system Γ working in the terminal mode (t-mode) is defined as

$$\begin{aligned} L(\Gamma) = \{w \in T^* \quad : \quad & \text{there exists } \ell \geq 1 \text{ such that } \alpha_i \Rightarrow_{k_i}^t \alpha_{i+1}, \\ & 1 \leq k_i \leq n, \text{ for each } i = 1, \dots, \ell - 1, \alpha_1 = S, \text{ and } \alpha_\ell = w\}. \end{aligned}$$

Let **CD(P)** denote the family of languages generated by permitting CD grammar systems working in the t-mode. It is proved in [3] that $\mathbf{CD(P)} = \mathbf{RC}$. (The reader is referred to [3] and [8] for more details on CD grammar systems with permitting and forbidding components, respectively.) Finally, note that the generative power of CD grammar systems, where g-components are permitting grammars using the definition (2) of the direct derivation step, is an open problem.

3 Results

Recall that it is known that $\mathbf{CF} \subset \mathbf{SC}(1,1)$ and $\mathbf{RC} \subset \mathbf{CS}$ (see, for instance, [14] and [4], respectively). For an example of a semi-conditional grammar of degree $(1,1)$ generating the set of all prime numbers, the reader is referred to [7].

3.1 Comparison of the two definitions

Theorem 1. $\mathbf{SC}(1,1) \subseteq \mathbf{RC}$.

Proof. Let $L \in \mathbf{SC}(1,1)$, then there is a semi-conditional grammar $G = (N,T,P,S)$ of degree $(1,1)$ such that $L(G) = L$. Construct the random context grammar $G' = (N',T,P',S)$ with $N' = N \cup \{a' : a \in T\}$ and P' constructed as follows:

1. set $P' = \{(A \to h(x), h(Per), h(For)) : (A \to x, Per, For) \in P\}$, where h is a homomorphism defined as $h(X) = X$, for $X \in N$, and $h(a) = a'$, for $a \in T$;
2. remove each production $(A \to x, Per, For)$ with $A \in For$ from P';
3. replace each production $(A \to x, Per, For)$ with $(A \to x, Per - \{A\}, For)$ in P';
4. for each $a \in T$, add $(a' \to a, \emptyset, N)$ to P'.

Thus, $(A \to h(x), h(Per) - \{A\}, h(For)) \in P'$ if and only if $(A \to x, Per, For) \in P$ and $A \notin For$. In addition,

- $(A \to x, Per, For) \in P$ is applicable to uAv in G if and only if
- $Per \subseteq alph(uAv)$ and $For \cap alph(uAv) = \emptyset$, which is if and only if
- $Per - \{A\} \subseteq alph(uv)$, $For \cap alph(uv) = \emptyset$, and $A \notin For$.
- This is if and only if $(A \to h(x), h(Per) - \{A\}, h(For))$ is applicable to $h(uAv)$ in G'.

As $h(Per \cup For) \subseteq N'$, G' is a random context grammar generating L. □

More generally, the previous proof gives a method how to transform any random context grammar using the definition (2) of the direct derivation step to an equivalent random context grammar using the definition (1). The converse transformation is proved so that each production $(A \to x, Per, For)$ is replaced with two productions $(A \to A', \emptyset, \{X' : X \in N\})$ and $(A' \to x, Per, For)$. Thus, both definitions of the relation of the direct derivation step are equivalent for random context grammars.

This paper also proves the analogous result for semi-conditional grammars of degree $(1,1)$. Let $\mathbf{SC}'(1,1)$ denote the family of languages generated by semi-conditional grammars of degree $(1,1)$ using the definition (1), then we have the following result.

Corollary 2. $\mathbf{SC}(1,1) \subseteq \mathbf{SC}'(1,1)$**.**

Proof. Modify the construction of $G' = (N',T,P',S)$ from the previous proof so that $N' = N$ and P' is constructed from P using only clauses 2 and 3. □

Theorem 3. $\mathbf{SC}(1,1) = \mathbf{SC}'(1,1)$.

Proof. By Corollary 2, it remains to show $\mathbf{SC}'(1,1) \subseteq \mathbf{SC}(1,1)$. Let $G = (N,T,P,S)$ be a semi-conditional of degree $(1,1)$ using the definition (1) such that $L(G) = L$. Construct the semi-conditional grammar $G' = (N',T,P',S')$ of degree $(1,1)$ using the definition (2), where S_1 is a new start symbol, $N' = N \cup \{S_1\} \cup \{[A] : A \in N \cup T\} \cup \{A' : A \in N\} \cup \{[pA],[p_1A],[p_2A] : p = (A \to \alpha,u,v) \in P\}$, and initialize

$$P' = \{(S_1 \to [S],0,0)\} \cup \{([a] \to a,0,0) : a \in T\}\,.$$

Then, for each production $p = (A \to \alpha,u,v) \in P$, the following productions are added to P'.

1. $([A] \to [x]\beta,u,v)$ for $\alpha = x\beta$, $x \in V$,

   and for each $B \in N \cup T$, add

2. $([B] \to [pB],0,0)$,
3. $(A \to A',[pB],A')$,
4. $([pB] \to [p_1B],A',0)$,
5. $([p_1B] \to [p_2B],u,v)$ for $v \neq B$,
6. $([p_1B] \to [p_2B],0,v)$ for $u = B$ and $v \neq B$,
7. $(A' \to \alpha,[p_2B],0)$,
8. $([p_2B] \to [B],0,A')$.

It is not hard to see that $L(G') = L(G)$. □

3.2 Generative power

Recall that the following holds: $\mathbf{CF} \subset \mathbf{SC}(1,1) \subseteq \mathbf{RC} \subset \mathbf{CS}$. In the rest of this section, we prove the other inclusion, i.e., we prove that $\mathbf{SC}(1,1) = \mathbf{RC}$. To do this, we first prove two auxiliary lemmas.

Lemma 4. *For each random context grammar G, there is an equivalent random context grammar G' such that $(A \to x,Per,For)$ is a production of G' implies that $A \notin For$.*

Proof. Let $G = (N,T,P,S)$ be a random context grammar. Construct the random context grammar $G' = (N \cup N',T,P',S)$, where $N' = \{A' : A \in N\}$ is such that $N \cap N' = \emptyset$, and $P' = \{(A \to A',\emptyset,N'),(A' \to x,Per,For) : (A \to x,Per,For) \in P\}$. Then, it is not hard to see that G and G' generate the same language and G' satisfies the required property. □

The following lemma proves that every random context language is generated by a CD grammar system with permitting components working in the t-mode, where each permitting set is of cardinality no more than one.

Lemma 5. *Every random context language is generated by a permitting CD grammar system where each permitting set is either empty or a one element set.*

Proof. Let L be a random context language, and let $G = (N, T, P, S)$ be a random context grammar generating L that satisfies the property of Lemma 4. Let the productions of P be labeled by numbers from 1 to $n = |P|$. Then, for each labeled production $i.(A \to x, Per, For) \in P$ with $Per = \{X_1, X_2, \dots, X_k\}$, for some $k \geq 0$, create a new component P_i containing the following productions:

1. $([A,i] \to [A,i,1], \emptyset)$,
2. $([A,i,j] \to [A,i,j+1], \{[X_j, i]\})$, for $1 \leq j \leq k$,
3. $([A,i,k+1] \to \langle h_i(x) \rangle, \emptyset)$, where h_i is a homomorphism defined as $h_i(X) = [X,i]$, for $X \in N$, and $h_i(a) = a$, for $a \in T$,
4. $(\langle h_i(x) \rangle \to \langle h_i(x) \rangle, \{\langle h_i(x) \rangle\})$,
5. $([X,i] \to [X,i], \emptyset)$, for $X \in For$,
6. $([X,i] \to [X,i]', \{\langle h_i(x) \rangle\})$, for $X \in N - For$,
7. $([A,i,j] \to [A,i,j], \emptyset)$, for $1 \leq j \leq k$,

and a new component $\overline{P_i}$ containing the following productions:

8. $([X,i] \to [X,j], \emptyset)$, for $X \in N$, $1 \leq j \leq n$,
9. $([X,k] \to [X,\ell], \{[Y,m]\})$, for $X, Y \in N$, $1 \leq k, \ell, m \leq n$, $k \neq m$.

Finally, add the component

$$P_0 = \{(S' \to [S,i], \emptyset), ([A,i]' \to [A,i], \emptyset), (\langle h_i(x) \rangle \to h_i(x), \emptyset) : A \in N, 1 \leq i \leq n\}.$$

Let $\Gamma = (N', T, P_0, P_1, \overline{P_1}, \dots, P_n, \overline{P_n}, S')$ be a permitting CD grammar system, where

$$\begin{aligned} N' &= \{S'\} \cup N \times \{1, 2, \dots, n\} \\ &\cup \{[A,i,j] : i.(A \to x, Per, For) \in P, 1 \leq j \leq |Per| + 1\} \\ &\cup \{\langle h_i(x) \rangle : (A \to x, Per, For) \in P, 1 \leq i \leq n\}. \end{aligned}$$

To prove that $L(G) \subseteq L(\Gamma)$, consider a derivation step of a successful derivation of G. Assume that a production $(A \to x, Per, For) \in P$ labeled by i is applied in this derivation step, i.e., $uAv \Rightarrow uxv$, $Per \subseteq alph(uv)$, and $For \cap alph(uv) = \emptyset$. We prove that

$$h_i(uAv) \Rightarrow_i^t h_i(u)' \langle h_i(x) \rangle h_i(v)' \Rightarrow_0^t h_i(uxv)$$

in Γ, where $h_i(z)'$ denotes $h_i(z)$ with all nonterminal symbols primed. Furthermore, if the next production applied in G is labeled by j, we prove that the derivation of Γ proceeds either by productions from P_i, for $i = j$, or, otherwise, by productions from $\overline{P_i}$, i.e., $h_i(uxv) \Rightarrow_{\overline{i}}^t h_j(uxv)$.

Clearly, by productions from P_i,

$$h_i(u)[A,i]h_i(v) \Rightarrow h_i(u)[A,i,1]h_i(v) \Rightarrow h_i(u)[A,i,2]h_i(v) \Rightarrow^* h_i(u)\langle h_i(x)\rangle h_i(v)$$

because all symbols from *Per* occur in uv. Then, all other nonterminals can be primed since there are no symbols from *For* in uv, i.e., $h_i(u)\langle h_i(x)\rangle h_i(v) \Rightarrow^* h_i(u)'\langle h_i(x)\rangle h_i(v)'$. Now, notice that only one symbol $\langle h_i(x)\rangle$ is presented in $h_i(u)'\langle h_i(x)\rangle h_i(v)'$, and, therefore, this component of Γ is blocked; i.e., the whole derivation by productions from P_i is $h_i(uAv) \Rightarrow_i^t h_i(u)'\langle h_i(x)\rangle h_i(v)'$. Then, by productions from P_0, the derivation proceeds as $h_i(u)'\langle h_i(x)\rangle h_i(v)' \Rightarrow_0^t h_i(uxv)$. Finally, for $j = i$, productions from P_i are applied again. Otherwise, if $j \neq i$, productions from $\bar{P}_i$ are applied and the derivation is $h_i(uxv) \Rightarrow_{\bar{i}}^t h_j(uxv)$. In either case, the proof proceeds by induction.

To prove the other inclusion, $L(\Gamma) \subseteq L(G)$, consider a successful derivation of Γ. Such a derivation is of the form $S' \Rightarrow_0^t \alpha_1 \Rightarrow^t \alpha_2 \Rightarrow^t \ldots \Rightarrow^t \alpha_k$, where $\alpha_k \in T^*$, for some $k \geq 1$. Assume that $\alpha_m \Rightarrow_i^t \alpha_{m+1}$ by productions from P_i, for some $i \in \{0, j, \bar{j}\}$, where $1 \leq j \leq n$ and $1 \leq m < k$, and that $\alpha_m = h_i(u_0Au_1Au_2 \ldots Au_r)$, for some $r \geq 0$, where $A \notin alph(u_0u_1 \ldots u_r)$, $r = 0$ implies that there is no $[A,i]$ in α_m, and $h_0 \in \{h_i : 1 \leq i \leq n\}$.

Then, with respect to i:

A. If $i = \bar{j}$, then $\alpha_{m+1} = h_\ell(u_0Au_1Au_2 \ldots Au_r)$, for some $\ell \neq i$. In addition, the only applicable productions are productions from P_ℓ and $\bar{P}_\ell$. Therefore, the derivation proceeds as in **A** or **B**.

B. If $i = j$, let $(A \to x, Per, For) \in P$ be the production labeled by i. Then, $u_0u_1 \ldots u_r \in h_i((V - (For \cup \{A\}))^*)$, which follows from the fact that the derivation is successful because if there appeared a symbol $X \in For$ in the sentential form, the derivation would keep replacing $[X,i]$ with $[X,i]$ for ever, see production 5. It also implies that $r \geq 1$; otherwise, there is no applicable production in P_i, but each component is required to make at least one derivation step. Therefore, according to the productions of P_i,

$$\alpha_{m+1} = h_i(u_0)'A_1h_i(u_1)'A_2h_i(u_2)' \ldots A_rh_i(u_r)', \tag{3}$$

where $A_1, A_2, \ldots, A_r \in \{\langle h_i(x)\rangle, [A,i]', [A,i,j] : 1 \leq j \leq n\}$, and $m+1 < k$. However, the derivation is successful only if there is no more than one occurrence of $\langle h_i(x)\rangle$ and no occurrence of a symbol of the form $[A,i,j]$ in α_{m+1}; otherwise, $\langle h_i(x)\rangle$ or $[A,i,j]$ are replaced with themselves for ever, see productions 4 and 7. This and production 6 imply that $A_1, A_2, \ldots, A_r \in \{\langle h_i(x)\rangle, [A,i]'\}$ and $|\alpha_{m+1}|_{\langle h_i(x)\rangle} = 1$. Finally, notice that only the productions of P_0 are applicable.

Thus, we can assume that $\alpha_m = h_i(v_0Av_1)$ and $\alpha_{m+1} = h_i(v_0)'\langle h_i(x)\rangle h_i(v_1)'$, for some $v_0v_1 \in (V - For)^*$. By productions constructed in 2 and 5, we have verified that $Per \subseteq alph(v_0v_1)$ and $For \cap alph(v_0v_1) = \emptyset$. Then,

$$v_0Av_1 \Rightarrow v_0xv_1$$

in G by the production $(A \to x, Per, For)$.

C. If $i = 0$, then, as shown above, there is an applicable production in P_0 only if α_m is of the form achieved in (3) above, i.e., $\alpha_m = h_i(u)'\langle h_i(x)\rangle h_i(v)'$, for some $x, uv \in V^*$, and $\alpha_{m+1} = h_i(uxv)$.

The proof now proceeds by induction.

As $\alpha_1 = [S, i]$, for some $1 \leq i \leq n$, the inclusion is proved. □

Using the previous lemma, we can prove that any random context language is generated by a semi-conditional grammar of degree $(1,1)$.

Theorem 6. $\mathbf{RC} \subseteq \mathbf{SC}(1,1)$.

Proof. Let L be a random context language, and let $\Gamma = (N, T, P_1, P_2, \ldots, P_n, S)$, for some $n \geq 1$, be a permitting CD grammar system working in t-mode generating L constructed as in Lemma 5. Let $V = N \cup T$. Construct the semi-conditional grammar of degree $(1,1)$ as follows. For each $(A \to x, Per) \in P_i$, recall that $|Per| \leq 1$, add

1. $(S' \to [S, i], 0, 0)$
   where $1 \leq i \leq n$;
2. $(A \to [x, Per], [X, i], 0)$
   where $X \in V$;
3. $([x, Per] \to x, Per, 0)$;
4. $([x, Per] \to x, [Z, i], 0)$
   where $Per = \{Z\}$;
5. $([A, i] \to [x_1, i]x_2 \ldots x_z, Per, 0)$
   where $x = x_1x_2 \ldots x_z$, for some $z \geq 1$, $x_i \in V$, $i = 1, \ldots, z$;
6. $([X, i] \to [X, Q_i], 0, 0)$
   where $X \in V$ and $Q_i = \{[x, Per] : (A \to x, Per) \in P_i\}$;
7. $([X, Q] \to [X, (Q - \{q\}) \cup \{q'\}], 0, q)$
   where $X \in V$, $Q \subseteq Q_i \cup Q_i'$, $Q_i' = \{x' : x \in Q_i\}$, and $q \in Q \cap Q_i$;
8. $([X, Q_i'] \to [X, P_i], 0, 0)$
   where $X \in V$;
9. $([X, P] \to [X, (P - \{p_j\}) \cup \{p_j'\}], 0, A_j)$
   where $P \subseteq P_i \cup P_i'$, $P_i' = \{x' : x \in P_i\}$, p_j is the label of $(A_j \to x_j, Per_j) \in P \cap P_i$, and $X \in V - \{A_j\}$;
10. $([X, P] \to [X, (P - \{p_j\}) \cup \{p_j'\}], A_j, Y)$ and
    $([A_j, P] \to [A_j, (P - \{p_j\}) \cup \{p_j'\}], 0, Y)$
    where $P \subseteq P_i \cup P_i'$, p_j is the label of $(A_j \to x_j, Per_j) \in P \cap P_i$, $Y \in Per_j$, and $X \in V - \{Y\}$;
11. $([X, P_i'] \to [X, j], 0, 0)$
    where $X \in V$ and $j \in \{1, 2, \ldots, n\}$.

12. $([x, P_i'] \rightarrow x, 0, 0)$
   where $x \in T$;

Let $G = (N', T, P', S')$ be the semi-conditional grammar of degree $(1,1)$ defined above, i.e., P' is defined as described above and

$$\begin{aligned} N' = N \cup \{S'\} \quad & \cup \quad \{[X,i] : X \in V, i \in \{1,2,\ldots,n\}\} \\ & \cup \quad \{[X,Q] : X \in V, Q \in \{Q_1, Q_2, \ldots, Q_n\}, Q_i \text{ are defined as above}\} \\ & \cup \quad \{[X,P] : X \in V, P \in \{P_1, P_2, \ldots, P_n\}\} \\ & \cup \quad \{[x, Per] : [x, Per] \in \bigcup_{i=1}^{n} Q_i\}. \end{aligned}$$

Informally, G simulates Γ so that it remembers the simulated component P_i of Γ in the first nonterminal, which is of the form $[X,i]$, for some $X \in V$. More specifically, productions 2 to 5 simulate the derivation steps of the ith component of Γ. Production 6 starts the verification process during which none of productions 2, 4, and 5 are applicable: productions constructed in 7 verify that there is no symbol of the form $[x, Per]$ in the sentential form; if so, production 3 is not applicable, and production 8 starts to verify whether there is no applicable production in P_i of Γ (see productions constructed in 9 and 10); if so, production 11 changes the simulated component, or production 12 finishes the derivation.

Formally, to prove that $L(\Gamma) \subseteq L(G)$, consider a successful derivation of Γ. Such a derivation is of the form $S \Rightarrow^t \alpha_1 \Rightarrow^t \alpha_2 \Rightarrow^t \ldots \Rightarrow^t \alpha_k$, where $\alpha_k \in T^*$, for some $k \geq 1$. Assume that $\alpha_m \Rightarrow_i^t \alpha_{m+1}$ by productions from P_i, for some $1 \leq i \leq n$ and $1 \leq m < k$. Let $\alpha_m = z_1 z_2 \ldots z_\ell$ and $\alpha_{m+1} = y_1 y_2 \ldots y_{\ell'}$, where $z_s, y_t \in V$ for all $s = 1, 2, \ldots, \ell$ and $t = 1, 2, \ldots, \ell'$. As the derivation of G starts by the application of a production constructed in 1, i.e., the sentential form is of the form $[S,i]$, for some $1 \leq i \leq n$, assume that $[z_1, i] z_2 \ldots z_\ell$ is the current sentential form of G. Then, if the rewritten symbol is the first symbol of the current sentential form of Γ, production 5 is applied in G, and if the rewritten symbol is not the first symbol of the sentential form of Γ, production 2 is applied in G followed by an application of production 3 or 4, where the choice depends on the permitting set. In either case, sentential forms of Γ and G modified as described above coincide except for the first symbol. However, if $x \in V$ is the first symbol of the sentential form of Γ, then $[x,i]$ is the first symbol of the sentential form of G, for some $1 \leq i \leq n$. Therefore, by the corresponding derivation replacing the same symbols at the same positions as in Γ, we have that $[z_1, i] z_2 \ldots z_\ell \Rightarrow^* [y_1, i] y_2 \ldots y_{\ell'}$ in G. There is no production applicable to α_{m+1} in Γ. Thus, production 6 is applied followed by a sequence of productions constructed in 7 verifying that there is no symbol of the form $[x, Per]$ in the sentential form. As there is no such symbol, production 8 is applied. As no productions from P_i are applicable in Γ, which means that either there is not the left-hand side of the production in the sentential form, or there is the left-hand side of the production but there is not a symbol from its permitting set in the sentential form, productions constructed in 9 and 10, followed by production 11, are applicable, i.e.,

$$\begin{aligned} [y_1, i] y_2 \ldots y_{\ell'} & \Rightarrow [y_1, Q_i] y_2 \ldots y_{\ell'} \Rightarrow^* [y_1, Q_i'] y_2 \ldots y_{\ell'} \\ & \Rightarrow [y_1, P_i] y_2 \ldots y_{\ell'} \Rightarrow^* [y_1, P_i'] y_2 \ldots y_{\ell'} \\ & \Rightarrow [y_1, j] y_2 \ldots y_{\ell'}, \end{aligned}$$

where j is such that $\alpha_{m+1} \Rightarrow_j^t \alpha_{m+2}$. The proof then proceeds by induction. If $m+1 = k$, then production 12 is applied instead of production 11.

To prove the other inclusion, $L(G) \subseteq L(\Gamma)$, consider a successful derivation of G. Such a derivation starts $S' \Rightarrow [S, j]$, for some $1 \leq j \leq n$. Consider a more general sentential form $[X,i]w$, for some $X \in V$, $1 \leq i \leq n$, and $w \in (N' \cup T)^*$. To simplify the proof, denote each nonterminal symbol $[x, Per]$ by the nonterminal that has generated it. It means, if, for instance, $(A \to [x, Per], [X,i], 0)$ was applied, write $[x, Per]_A$. Assume that $S \Rightarrow^* X f(w)$ in Γ, where f is a homomorphism defined as $f([x, Per]_A) = A$, and $f(X) = X$ otherwise. Then, there are the following possibilities how to proceed the derivation:

1. If production 2 is applied in the successful derivation, i.e., $S' \Rightarrow^* [X,i]uAv \Rightarrow [X,i]u[x, Per]_A v$. Then, by the assumption,
$$S \Rightarrow^* X f(u)Af(v) = X f(u[x, Per]_A v)$$
in Γ.

2. Assume that production 3 or 4 is applied in the successful derivation, replacing the nonterminal $[x, Per]_A$. Then, there had to be a preceding application of a production constructed in 2 in the derivation, i.e.,
$$S' \Rightarrow^* [Y,i]uAv \Rightarrow [Y,i]u[x, Per]_A v \Rightarrow^* [X,i]u'[x, Per]_A v' \Rightarrow [X,i]u'xv',$$
where i is unchanged in the first nonterminals of the shown part of the derivation as proved in 4 below. By the assumption and the production $(A \to x, Per) \in P_i$,
$$S \Rightarrow^* X f(u')Af(v') \Rightarrow X f(u')xf(v')$$
because $Per \subseteq alph(Xu'v') \cap N \subseteq alph(X f(u'v'))$.

3. If production 5 is applied in the successful derivation, $[X,i]w \Rightarrow [x_1,i]x_2 \ldots x_k w$, then
$$S \Rightarrow^* X f(w) \Rightarrow x_1 x_2 \ldots x_k f(w)$$
by the production $(X \to x_1 x_2 \ldots x_k, Per) \in P_i$.

4. Finally, assume that production 6 is applied in the successful derivation. Then, only productions constructed in 7 and 3 are applicable, followed by an application of production 8, i.e.,
$$[X,i]\overline{w} \Rightarrow [X,Q_i]\overline{w} \Rightarrow^* [X,Q_i']w \Rightarrow [X,P_i]w.$$
However, each of the productions constructed in 7 primes a symbol $[y, Per] \in Q_i$ only if there is no nonterminal symbol $[y, Per]$ in the current sentential form. Therefore, after this part of the derivation, it is verified that $w \in V^*$, which implies that any application of a production constructed in 2 is followed by an application of a production constructed in 3 or 4 before production 8 is applied. By the assumption and the argument analogous to the argument in 2 above,
$$S \Rightarrow^* X f(\overline{w}) \Rightarrow^* Xw.$$

Then, only productions constructed in 9 and 10 are applicable, i.e.,

$$[X, P_i]w \Rightarrow^* [X, P_i']w\,.$$

More specifically, if production 9 is applied, then A_j does not occur in the sentential form Xw, which implies that the production $p_j.(A_j \to x_j, Per_j) \in P_i$ is not applicable in Γ. On the other hand, if production 10 is applied, then A_j occurs in the current sentential form, but some $Y \in Per_j$ does not. Again, the production $p_j.(A_j \to x_j, Per_j) \in P_i$ is not applicable in Γ. As all productions of P_i are checked by this part of the derivation, it is verified that there is no production in P_i applicable by Γ. Then, production 11 is applied, which Γ simulates by changing the component.

If production 12 is applied, then no production constructed in 2 is applicable, which implies that $Xw \in T^*$, and the derivation is successfully finished.

As, in all cases, the sentential form is of the form $[Y, j]w'$, for some $Y \in V$, $1 \leq j \leq n$, and $w' \in (N' \cup T)^*$. The proof proceeds by induction. □

Thus, we have proved that the family of random context languages and the family of semi-conditional languages of degree $(1,1)$ coincide.

Corollary 7. $\mathbf{RC} = \mathbf{SC}(1,1)$.

3.3 Normal forms of random context grammars

This section discusses the normal forms of random context grammars. Specifically, it proves that the normal form proved by Mayer in [13] for random context grammars with erasing productions holds for random context grammars in general. It means that it holds for random context grammars without erasing productions, too.

Definition 1. A random context grammar $G = (N, T, P, S)$ is called *production-limited* if every production from P is of one of the following three forms:

1. $(A \to BC, Per, For)$
2. $(A \to B, Per, For)$
3. $(A \to a, \emptyset, \emptyset)$

where $A, B, C \in N$, $a \in T$, and $Per, For \subseteq N$.

Definition 2. A random context grammar $G = (N, T, P, S)$ is called *limited* if it is production-limited and, in addition, each $Per, For \subseteq N$ is either empty or a one element set.

Mayer [13, Theorem 6] proved that if erasing productions are allowed, then each recursively enumerable language can be generated by a limited random context grammar. In the nonerasing case, however, he only proved (see [13, Lemmas 7 and 8]) that every random context language can be generated by a production-limited random context grammar, and it was left open whether the same normal form also holds for random context grammars without erasing productions. The following corollary answers this question.

Corollary 8. *Every random context language can be generated by a limited random context grammar.*

Proof. Given a production-limited random context grammar, the sequence of applications of constructions of Lemma 5, Theorem 6, and Corollary 2, respectively, preserves the required form of productions. The resulting grammar is random context because there are no terminal symbols in permitting and forbidding sets. In addition, each of these sets is either empty or contains only one element. □

4 Conclusion

This section summarizes the results and open problems concerning random context grammars and semi-conditional grammars. In what follows, the superscript λ is added if erasing productions are allowed.

Theorem 9. *The following holds for grammars with erasing productions. The proofs can be found in [1, 7, 13, 14].*

1. $\mathbf{SC}^{\lambda}(\mathbf{0},\mathbf{0}) = \mathbf{CF}$.
2. $\mathbf{CF} \subset \mathbf{SC}^{\lambda}(\mathbf{0},\mathbf{1}) \subseteq \mathbf{F}^{\lambda} \subset \mathbf{REC}$.
3. $\mathbf{CF} \subset \mathbf{SC}^{\lambda}(\mathbf{1},\mathbf{0}) \subseteq \mathbf{P}^{\lambda} \subset \mathbf{REC}$.
4. $\mathbf{SC}^{\lambda}(\mathbf{1},\mathbf{1}) = \mathbf{RE}$.

Theorem 10. *The following holds for grammars without erasing productions. The proofs can be found in [1, 5, 14, 17]. The first part 5 is proved in this paper.*

1. $\mathbf{SC}(\mathbf{0},\mathbf{0}) = \mathbf{CF}$.
2. $\mathbf{CF} \subset \mathbf{SC}(\mathbf{0},\mathbf{1}) \subseteq \mathbf{F} \subset \mathbf{RC}$.
3. $\mathbf{CF} \subset \mathbf{SC}(\mathbf{1},\mathbf{0}) \subseteq \mathbf{P} \subset \mathbf{RC}$.
4. $\mathbf{SC}(\mathbf{2},\mathbf{1}) = \mathbf{SC}(\mathbf{1},\mathbf{2}) = \mathbf{CS}$.
5. $\mathbf{SC}(\mathbf{1},\mathbf{1}) = \mathbf{RC} \subset \mathbf{CS}$.

The generative power of semi-conditional grammars of degree $(0,i)$ and $(i,0)$ (with or without erasing productions), for $i \geq 2$, is not known. However, if more than one forbidding string is allowed to be attached to a production (i.e., there are sets of forbidding strings instead of only one string), it is known that such grammars (referred to as *generalized forbidding grammars*) are computationally complete. In addition, it is sufficient to have no more than four forbidding strings each of which is of length one or two to characterize the family of recursively enumerable languages (see [9, Corollary 6]). On the other hand, however, the question of what is the generative power of *generalized permitting grammars* (defined in the same manner) is an open problem.

Let $(A \to \alpha, u, v)$ be a production of a semi-conditional grammar. If $u = v = 0$, then it is said to be context-free; otherwise, it is said to be *conditional*. The latest descriptional complexity result showing that only a finite number of resources is needed by semi-conditional grammars to generate any recursively enumerable language is proved in [11].

Theorem 11 ([11])**.** *Every recursively enumerable language is generated by a semi-conditional grammar of degree* $(2,1)$ *with seven conditional productions and eight nonterminals.*

Finally, Example 4.1.1 in [4] shows that there is no bound on the number of nonterminals for random context grammars. (The proof works for semi-conditional grammars of degree $(1,1)$ where terminals are not allowed to appear as permitting or forbidding symbols, too.) More specifically, the example shows that any random context grammar generating the language

$$T_n = \bigcup_{i=1}^{n} \{a_i^j : j \geq 1\}$$

requires, in the nonerasing case, exactly $n+1$ nonterminals and, in the erasing case, at least $f(n)$ nonterminals, for some unbounded mapping $f : N \to N$.

In the case of semi-conditional grammars, terminal symbols are allowed to appear as both permitting and forbidding symbols. As $G = (\{S, A\}, \{a_1, a_2, \ldots, a_n\}, P, S)$, where

$$P = \{(S \to a_i A, 0, 0), (S \to a_i, 0, 0), (A \to a_i A, a_i, 0), (A \to a_i, a_i, 0) : 1 \leq i \leq n\},$$

is a semi-conditional grammar of degree $(1,0)$ generating T_n, the question of whether analogous descriptional complexity results can be achieved for semi-conditional grammars of degree $(1,1)$ is open.

References

[1] H. Bordihn and H. Fernau. Accepting grammars and systems. Technical Report 9/94, Universitat Karlsruhe, Fakultat fur Informatik, 1994.

[2] E. Csuhaj-Varjú, J. Dassow, J. Kelemen, and Gh. Păun. *Grammar Systems: A Grammatical Approach to Distribution and Cooperation*. Gordon and Breach Science Publishers, Topics in Computer Mathematics 5, Yverdon, 1994.

[3] E. Csuhaj-Varjú, T. Masopust, and Gy. Vaszil. Cooperating distributed grammar systems with permitting grammars as components. *Romanian Journal of Information Science and Technology*, 12(2):175–189, 2009.

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

[5] S. Ewert and A. P. J. van der Walt. A pumping lemma for random permitting context languages. *Theoretical Computer Science*, 270(1–2):959–967, 2002.

[6] T. Masopust. Simple restriction in context-free rewriting. Submitted manuscript.

[7] T. Masopust. A note on the generative power of some simple variants of context-free grammars regulated by context conditions. In A.H. Dediu, A.M. Ionescu, and C. Martín-Vide, editors, *LATA 2009 proceedings*, volume 5457 of *Lecture Notes in Computer Science*, pages 554–565. Springer-Verlag, 2009.

[8] T. Masopust. On the terminating derivation mode in cooperating distributed grammar systems with forbidding components. *Internation Journal of Foundations of Computer Science*, 20(2):331–340, 2009.

[9] T. Masopust and A. Meduna. Descriptional complexity of generalized forbidding grammars. In *Proceedings of 9th International Workshop on Descriptional Complexity of Formal Systems*, pages 170–177. High Tatras, Slovakia, 2007.

[10] T. Masopust and A. Meduna. Descriptional complexity of grammars regulated by context conditions. In *Pre-proceedings of 1st International Conference on Language and Automata Theory and Application (LATA 2007)*, pages 403–411, Tarragona, Spain, 2007.

[11] T. Masopust and A. Meduna. Descriptional complexity of semi-conditional grammars. *Information Processing Letters*, 104(1):29–31, 2007.

[12] T. Masopust and A. Meduna. On context-free rewriting with a simple restriction and its computational completeness. *RAIRO – Theoretical Informatics and Applications*, 43(2):365–378, 2009.

[13] O. Mayer. Some restrictive devices for context-free grammars. *Information and Control*, 20:69–92, 1972.

[14] Gh. Păun. A variant of random context grammars: Semi-conditional grammars. *Theoretical Computer Science*, 41:1–17, 1985.

[15] A. Salomaa. *Formal languages*. Academic Press, New York, 1973.

[16] A. P. J. van der Walt. Random context grammars. In *Proceedings of the Symposium on Formal Languages*, pages 163–165. 1970.

[17] A. P. J. van der Walt and S. Ewert. A shrinking lemma for random forbidding context languages. *Theoretical Computer Science*, 237(1-2):149–158, 2000.

[18] Gy. Vaszil. On the descriptional complexity of some rewriting mechanisms regulated by context conditions. *Theoretical Computer Science*, 330:361–373, 2005.