Computational Completeness Resulting from Scattered Context Grammars Working Under Various Derivation Modes

Alexander Meduna and Ondřej Soukup

Brno University of Technology, Faculty of Information Technology Centre of Excellence, Božetěchova 1/2, 612 66 Brno, Czech Republic
meduna@fit.vutbr.cz, isoukup@fit.vutbr.cz

Abstract. This paper introduces and studies a whole variety of derivation modes in scattered context grammars. These grammars are conceptualized just like classical scattered context grammars except that during the applications of their rules, after erasing n nonterminals, they can insert new substrings possibly at different positions than the original occurrence of the erased nonterminal.

The paper concentrates its attention on investigating the generative power of scattered context grammars working under these derivation modes. It demonstrates that all of them are computationally complete– that is, they characterize the family of recursively enumerable languages.

Keywords: scattered context grammars; alternative derivation modes; generative power; computational completeness.

1 Introduction

The present section informally sketches scattered context grammars working under various new derivation modes and explains the reason why they are introduced. This section also describes how the paper is organized.

At present, processing information in a discontinuous way represents a common computational phenomenon. Indeed, consider a process p that deals with information i. Typically, during a single computational step, p (1) reads n pieces of information, x_1 through x_n, in i, (2) erases them, (3) generate n new pieces of information, y_1 through y_n, and (4) inserts them into i possibly at different positions than the original occurrence of x_1 through x_n, which was erased. To explore computation like this systematically and rigorously, computer science obviously needs formal models that reflect it in an adequate way.

Traditionally, formal language theory has always provided computer science with language-defining models to explore various information processors mathematically, so it should do so for the purpose sketched above as well. However, the classical versions of these models, such as grammars, work on words so they

erase and insert subwords at the same position, hence they can hardly serve as appropriate models of this kind. Therefore, a proper formalization of processors that work in the way described above needs an adaptation of some classical well-known grammars so they reflect the above-described computation more adequately. At the same time, any adaptation of this kind should conceptually maintain the original structure of these models as much as possible so computer science can quite naturally base its investigation upon these newly adapted grammatical models by analogy with the standard approach based upon their classical versions. Simply put, while keeping their structural conceptualization unchanged, these grammatical models should work on words in newly introduced ways, which more properly reflect the above-mentioned modern computation.

The present paper discusses this topic in terms of scattered context grammars, which definitely represent important language-generating grammatical models of computation. Indeed, the paper introduces a whole variety of derivation modes in scattered context grammars so they reflect the above-sketched computation in a more adequate way than the standard derivation mode.

Recall that the notion of a scattered context grammar G represents a language-generating rewriting system based upon an alphabet of symbols and a finite set of rules. The alphabet of symbols is divided into two disjoint subalphabets—the alphabet of terminal symbols and the alphabet of nonterminal symbols. In G, a rule r is of the form

$$(A_1, A_2, \ldots, A_n) \rightarrow (x_1, x_2, \ldots, x_n),$$

for some positive integer n. On the left-hand side of r, the As are nonterminals. On the right-hand side, the xs are strings. G can apply r to any string u of the form

$$u = u_0 A_1 u_1 \ldots u_{n-1} A_n u_n$$

where us are any strings. Notice that A_1 through A_n are scattered throughout u, but they occur in the order prescribed by the left-hand side of r. In essence, G applies r to u so

(1) it deletes A_1, A_2, ..., A_n in u, after which
(2) it inserts x_1, x_2, ..., x_n into the string resulting from the deletion (1).

By this application, G makes a derivation step from u to a string v of the form

$$v = v_0 x_1 v_1 \ldots v_{n-1} x_n v_n$$

Notice that x_1, x_2, ..., x_n are inserted in the order prescribed by the right-hand side of r. However, they are inserted in a scattered way—that is, in between the inserted xs, some substrings vs occur.

This paper partially introduces the results of larger study, which is currently being in progress and will be hopefully published soon. In this study, 9 derivation modes of scattered context grammars are defined, however, due to shortage of space, only a few selected modes are presented here. For consistence, their

numbering is preserved. The chosen modes are mutually dual or complementary to the others.

(1) Mode 1 requires that $u_i = v_i$ for all $i = 0, \ldots, n$ in the above described derivation step.
(3) Mode 3 obtains v from u so it changes u by performing (3a) through (3c), described next:
  (a) $A_1, A_2, \ldots, A_n$ are deleted;
  (b) x_1 and x_n are inserted into u_0 and u_n, respectively;
  (c) x_2 through x_{n-1} are inserted in between the newly inserted x_1 and x_n.
(5) In mode 5, v is obtained from u by (5a) through (5e), given next:
  (a) $A_1, A_2, \ldots, A_n$ are deleted;
  (b) a central u_i is nondeterministically chosen, for some $0 \leq i \leq n$;
  (c) x_1 and x_n are inserted into u_0 and u_n, respectively;
  (d) x_j is inserted between u_{j-2} and u_{j-1}, for all $1 < j \leq i$;
  (e) x_k is inserted between u_k and u_{k+1}, for all $i + 1 \leq k < n$.
(7) Mode 7 obtains v from u performing the steps stated below:
  (a) $A_1, A_2, \ldots, A_n$ are deleted;
  (b) a central u_i is nondeterministically chosen, for some $0 \leq i \leq n$;
  (c) x_j is inserted between u_{j-2} and u_{j-1}, for all $1 < j \leq i$;
  (d) x_k is inserted between u_k and u_{k+1}, for all $i + 1 \leq k < n$.

This paper is organized as follows. Section 2 gives all the necessary notation and terminology to follow the rest of the paper. Then, Section 3 formally introduces all the new derivation modes in scattered context grammars. After that, Section 4 demonstrates that scattered context grammars working under any of the newly introduced derivation modes are computationally complete–that is, they characterize the family of recursively enumerable languages.

2 Preliminaries

We assume that the reader is familiar with formal language theory (see [1, 2]). For a set W, card(W) denotes its cardinality. Let V be an alphabet (finite nonempty set). V^* is the set of all strings over V. Algebraically, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$. Algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. For $w \in V^*$, $|w|$ and reversal(w) denote the length of w and the reversal of w, respectively. For $L \subseteq V^*$, reversal$(L) = \{\text{reversal}(w) \mid w \in L\}$. The alphabet of w, denoted by alph(w), is the set of symbols appearing in w. For $v \in \Sigma$ and $w \in \Sigma^*$, occur(v, w) equals the number of occurrences of v in w.

Let ϱ be a relation over V^*. The transitive and transitive and reflexive closure of ϱ are denoted ϱ^+ and ϱ^*, respectively. Unless explicitly stated otherwise, we write $x \ \varrho \ y$ instead $(x, y) \in \varrho$.

The families of regular languages, context-free languages, and recursively enumerable languages are denoted by **REG**, **CF**, and **RE**, respectively. Recall that scattered context grammars characterize **RE** (see [3]).

3 Definitions

In this section, we define scattered context grammars and the following new derivation modes in scattered context grammars. Then, we illustrate them by examples.

Definition 1. A *scattered context grammar* (an *SCG* for short) is a quadruple

$$G = (V, T, P, S)$$

where

- V is an alphabet;
- $T \subset V$;
- set $N = V - T$;
- $P \subseteq \bigcup_{m=1}^{\infty} \left(N_1 \times N_2 \times \cdots \times N_m \times V_1^* \times V_2^* \times \cdots \times V_m^* \right)$ is finite, where each $N_j = N$, $V_j = V$, $1 \leq j \leq m$;
- $S \in N$.

V, T and N are called the *total alphabet*, the *terminal alphabet* and the *nonterminal alphabet*, respectively. P is called the set of *productions*. Instead of

$$(A_1, A_2, \ldots, A_n, x_1, x_2, \ldots, x_n) \in P$$

where $A_i \in N$, $x_i \in V^*$, for $1 \leq i \leq n$, for some $n \geq 1$, we write

$$(A_1, A_2, \ldots, A_n) \to (x_1, x_2, \ldots, x_n)$$

S is the *start symbol*. □

Definition 2. Let $G = (V, T, P, S)$ be an SCG, and let ϱ be a relation over V^*. Set

$$L(G, \varrho) = \{x \mid x \in T^*, S \ \varrho^* \ x\}$$

$L(G, \varrho)$ is said to be the *language that G generates by ϱ*. Set

$$SC(\varrho) = \{L(G, \varrho) \mid G \text{ is an SCG}\}$$

$SC(\varrho)$ is said to be *the language family that SCGs generate by ϱ*. □

Definition 3. Let $G = (V, T, P, S)$ be an SCG. Next, we define the following direct derivation relations ${}_1\Rightarrow$ through ${}_9\Rightarrow$ over V^*.

First, let $(A) \to (x) \in P$ and $u = w_1 A w_2 \in V^*$. Then,

$$w_1 A w_2 \ {}_i\Rightarrow \ w_1 x w_2, \ i \in \{1, 2, \ldots, 9\}$$

Second, let $(A_1, A_2, \ldots, A_n) \to (x_1, x_2, \ldots, x_n) \in P$ and $u = u_0 A_1 u_1 \ldots A_n u_n$, z, z', u_i, $v_i, w_i \in V^*$, for all $0 \leq i \leq n$ and $1 \leq j \leq n-1$, for some $n \geq 2$, and $u_0 u_1 \ldots u_n = v_0 v_1 \ldots v_n$. Then,

(1) $u_0A_1u_1A_2u_2 \dots A_nu_n \ {}_1\Rightarrow u_0x_1u_1x_2v_2 \dots x_nu_n$;
(3) $u_0A_1u_1A_2u_2 \dots A_nu_n \ {}_3\Rightarrow v_0x_1v_1x_2v_2 \dots x_nv_n$, where $u_0 = v_0z$, $u_n = z'v_n$;
(5) $u_0u_1A_1u_2A_2 \dots u_{i-1}A_iu_iA_{i+1}u_{i+1} \dots A_nu_nz \ {}_5\Rightarrow$
$u_0x_1u_1x_2u_2 \dots x_iu_{i-1}u_iu_{i+1}x_{i+1} \dots u_nx_nz$;
(7) $u_0A_1u_2A_2 \dots u_{i-1}A_iu_iA_{i+1}u_{i+1} \dots A_nu_n \ {}_7\Rightarrow$
$u_0x_2u_2 \dots x_iu_{i-1}u_iu_{i+1}x_{i+1} \dots u_n$;

□

To illustrate the above-introduced notation, let $G = (V, T, P, S)$ be an SCG; then, $L(G, {}_5\Rightarrow) = \{x \mid x \in T^*, S \ {}_5\Rightarrow^* x\}$ and $SC({}_5\Rightarrow) = \{L(G, {}_5\Rightarrow) \mid G \text{ is an SCG}\}$. To give another example, $SC({}_1\Rightarrow)$ denotes the family of all scattered context languages.

4 Generative Power

In this section, for each defined derivation mode we investigate the generative power of SCGs using this mode.

Lemma 1. *Let $L \subseteq \Sigma^*$ be any recursively enumerable language. Then, L can be represented as $L = h(L_1 \cap L_2)$, where $h : T^* \to \Sigma^*$ is a morphism and L_1 and L_2 are two context-free languages.*

For a proof, see [4].

4.1 Mode 1

We prove that SCGs with mode 1 derivations characterize the family of recursively enumerable languages.

Theorem 1. [3] $SC({}_1\Rightarrow) = \mathbf{RE}$.

Since $SC({}_1\Rightarrow) \subseteq \mathbf{RE}$ follows directly from the Church-Turing thesis, we only have to prove the opposite inclusion.

Proof. Construction. Recall Lemma 1. By the closure properties of context-free languages, there are context-free grammars G_1 and G_2 that generate L_1 and reversal(L_2), respectively. More precisely, let $G_i = (V_i, T, P_i, S_i)$ for $i = 1, 2$. Let $T = \{a_1, \dots, a_n\}$ and 0, 1, \$, $S \notin (V_1 \cup V_2 \cup \Sigma)$ be the new symbols. Without any loss of generality, assume that $V_1 \cap V_2 = \emptyset$. Define the new morphisms

(1) $c : a_i \mapsto 10^i1$;
(2) $C_1 : V_1 \cup T \to V_1 \cup \Sigma \cup \{0,1\}^*$, $\begin{cases} A \mapsto A, & A \in V_1, \\ a \mapsto f(a), & a \in T; \end{cases}$
(3) $C_2 : V_2 \cup T \to V_2 \cup \{0,1\}^*$, $\begin{cases} A \mapsto A, & A \in V_2, \\ a \mapsto c(a), & a \in T; \end{cases}$
(4) $f : a_i \mapsto h(a_i)c(a_i)$;
(5) $t : \Sigma \cup \{0,1,\$\} \to \Sigma$, $\begin{cases} a \mapsto a, & a \in \Sigma, \\ A \mapsto \varepsilon, & A \notin \Sigma; \end{cases}$
(6) $t' : \Sigma \cup \{0,1,\$\} \to \{0,1\}$, $\begin{cases} a \mapsto a, & a \in \{0,1\}, \\ A \mapsto \varepsilon, & A \notin \{0,1\}. \end{cases}$

Finally, let $G = (V, \Sigma, P, S)$ be SCG, with $V = V_1 \cup V_2 \cup \{S, 0, 1, \$\}$ and P containing the rules

(1) $(S) \to (\$S_1 1111 S_2 \$)$;
(2) $(A) \to (C_i(w))$, for all $A \to w \in P_i$, where $i = 1, 2$;
(3) $(\$, a, a, \$) \to (\varepsilon, \$, \$, \varepsilon)$, for $a = 0, 1$;
(4) $(\$) \to (\varepsilon)$.

Claim 1. $L(G, {}_1\Rightarrow) = L$.

Proof. Basic idea. First the starting rule from (1) is applied. The starting nonterminals S_1 and S_2 are inserted into the current sentential form. Then, by using the rules from (2) G simulates derivations in both G_1 and G_2 and generates the sentential form $w = \$w_1 1111 w_2 \$$.

Suppose $S_1 \Rightarrow^* w$, where $\text{alph}(w) \cap (V_1 \cup V_2) = \emptyset$. If $t'(w_1) = \text{reversal}(w_2)$, then $t(w_1) = h(v)$, where $v \in L_1 \cap L_2$ and $h(v) \in L$. In other words, w represents a successful derivation of both G_1 and G_2, where the both grammars have generated the same sentence v, therefore G must generate the sentence $h(v)$.

The rules from (3) serve to check, whether the simulated grammars have generated the identical words. Binary codings of the generated words are erased while checking the equality. Each time the leftmost and the rightmost symbols are erased, otherwise some symbol is skipped. If the codings do not match, some 0 or 1 cannot be erased and no terminal string can be generated.

Finally, the symbols \$ are erased with the rule from (4), and if G_1, G_2, respectively, generated the same sentence and both codings were successfully erased, then the G has generated the terminal sentence $h(v)$. □

For a rigorous proof, see [3]. Since L is an arbitrary recursively enumerable language, by Claim 1 the proof of Theorem 1 is completed. □

4.2 Mode 3

In this section, we prove the family of languages generated by SCGs with mode 3 derivations coincides with the family of recursively enumerable languages.

Theorem 2. $SC({}_3\Rightarrow) = \mathbf{RE}$.

Since $SC({}_3\Rightarrow) \subseteq \mathbf{RE}$ follows directly from the Church-Turing thesis, we only have to prove the opposite inclusion.

Proof. Let $G = (V, \Sigma, P, S)$ be the SCG constructed in the proof of Theorem 1. Next, we modify G to a new SCG G' such that $L(G, {}_1\Rightarrow) = L(G', {}_1\Rightarrow)$. Finally, we prove $L(G', {}_3\Rightarrow) = L(G', {}_1\Rightarrow)$.

Construction. Let $G' = \{V, \Sigma, P', S\}$ be SCG with P' containing

(1) $(S) \to (S_1 11\$\$11 S_2)$;
(2) $(A) \to (C_i(w))$ for $A \to w \in P_i$, where $i = 1, 2$;
(3) $(a, \$, \$, a) \to (\$, \varepsilon, \varepsilon, \$)$, for $a = 0, 1$;
(4) $(\$) \to (\varepsilon)$.

We establish the proof of Theorem 2 by the following two claims.

Claim 2. $L(G', {}_1\Rightarrow) = L(G, {}_1\Rightarrow)$.

Proof. G' is closely related to G, only the rules from (1) and (3) are slightly modified. As a result the correspondence of the sentences generated by the simulated G_1, G_2, respectively, is not checked in the direction from the outermost to the central symbols but from the central to the outermost symbols. Again, if the current two symbols do not match, they can not be erased both and the derivation blocks. □

Claim 3. $L(G', {}_3\Rightarrow) = L(G', {}_1\Rightarrow)$.

Proof. Without any loss of generality, we can suppose the rules from (1) and (2) are used only before the first usage of the rule from (3). The context-free rules work unchanged with mode 3 derivations. Then, for every derivation

$$S \; {}_1\Rightarrow^* w = w_1 11\$\$11 w_2$$

generated only by the rules from (1) and (2), where $\text{alph}(w) \cap (V_1 \cup V_2) = \emptyset$, there is the identical derivation

$$S \; {}_3\Rightarrow^* w$$

and vice versa. Since

$$w \; {}_1\Rightarrow^* w', w' \in \Sigma^*$$

if and only in $t'(w_1) = \text{reversal}(w_2)$, we can complete the proof of the previous claim by the following one.

Claim 4. Let the sentential form w be generated only by the rules from (1) and (2). Without any loss of generality, suppose $\text{alph}(w) \cap (V_1 \cup V_2) = \emptyset$. Consider

$$S \; {}_3\Rightarrow^* w = w_1 11\$\$11 w_2$$

Then, $w \; {}_3\Rightarrow^* w'$, where $w' \in \Sigma^*$, if and only if $t'(w_1) = \text{reversal}(w_2)$.

For better readability, in the next proof we omit all symbols of w_1 from Σ— we consider only nonterminal symbols, which are to be erased.

Basic idea. The rules from (3) are the only with 0s and 1s on their left hand sides. These symbols are simultaneously erasing to the left and to the right of \$s checking the equality in reverse. While proceeding from the center to the edges, when there is any symbol skipped, which is remaining between \$s, there is no way, how to erase it, and no terminal string can be generated.

Consider the mode 3 derivations. Even when the symbols are erasing one after another from the center to the left and right, the derivation mode can potentially shift left one \$ to the left and right one \$ to the right skipping some symbols. Also in this case the symbols between \$s can not be erased anymore.

Proof. If. Recall

$$w = 10^{m_1}110^{m_2}1 \dots 10^{m_o}111\$\$1110^{m_o}1 \dots 10^{m_2}110^{m_1}1$$

Suppose the check works properly not skipping any symbol. Then

$$w \ {}_3\Rightarrow^* \ w' = \$\$$$

and twice applying the rule from (4) the derivation finishes. □

Proof. Only if. If $w_1 \neq \text{reversal}(w_2)$, though the check works properly,

$$w \ {}_1\Rightarrow^* \ w' = w'_1 x\$\$x' w'_2$$

and $x, x' \in \{0,1\}$, $x \neq x'$. Continuing the check with application of the rules from (3) will definitely skip x or x'. Consequently, no terminal string can be generated.

We showed, that G' can generate the terminal string from the sentence form w, if only if $t'(w_1) = \text{reversal}(w_2)$, and the claim holds. □

Since $S \ {}_1\Rightarrow^* \ w$, $w \in \Sigma^*$, if and only if $S \ {}_3\Rightarrow^* \ w$, Claim 3 holds. □

We proved $L(G, {}_1\Rightarrow) = L$, $L(G', {}_1\Rightarrow) = L(G, {}_1\Rightarrow)$ and $L(G', {}_3\Rightarrow) = L(G', {}_1\Rightarrow)$, therefore $L(G', {}_3\Rightarrow) = L$ holds. Since L is an arbitrary recursively enumerable language, the proof of Theorem 2 is completed. □

4.3 Mode 5

This section investigates mode 5 derivations. It proves the family of languages SCGs with mode 5 derivations generates corresponds to the family of recursively enumerable languages.

Theorem 3. $SC({}_5\Rightarrow) = \mathbf{RE}$.

Since $SC({}_5\Rightarrow) \subseteq \mathbf{RE}$ follows directly from the Church-Turing thesis, we only have to prove the opposite inclusion.

Proof. Let $G = (V, \Sigma, P, S)$ be the SCG constructed in the proof of Theorem 1. Next, we modify G to a new SCG G' so $L(G, {}_1\Rightarrow) = L(G', {}_5\Rightarrow)$.

Construction. Introduce four new symbols—D,E,F and $\circ$. Set $N = \{D,E,F,\circ\}$. Let $G' = (V', \Sigma, P', S)$ be SCG, with $V' = V \cup N$ and P' containing the rules

(1) $(S) \to (\$S_1 1111 S_2 \$ \circ E \circ F)$;
(2) $(A) \to (C_i(w))$ for $A \to w \in P_i$, where $i = 1, 2$;
(3) $(F) \to (FF)$;
(4) $(\$, a, a, \$, E, F) \to (\varepsilon, \varepsilon, \$, \$, \varepsilon, D)$, for $a = 0, 1$;
(5) $(\circ, D, \circ) \to (\varepsilon, \circ E \circ, \varepsilon)$;
(6) $(\$) \to (\varepsilon)$, $(E) \to (\varepsilon)$, $(\circ) \to (\varepsilon)$.

Claim 5. $L(G, {}_1\Rightarrow) = L(G', {}_5\Rightarrow)$.

Proof. Context-free rules are not influenced by the derivation mode. The rule from (3) must generate precisely as many Fs as the number of applications of the rule from (4). Context-sensitive rules of G' correspond to context-sensitive rules of G, except the special rule from (5). We show, the construction of G' forces context-sensitive rules to work exactly in the same way as the rules of G do.

Every application of the rule from (4) must be followed by the application of the rule from (5), to rewrite D back to E, which requires the symbol D between two $\circ$s. It ensures the previous usage of context-sensitive rule selected the center to the right of the rightmost affected nonterminal and all right hand side strings changed their positions with the more left ones. The leftmost right hand side string is then shifted randomly to the left, but it is always ε. The derivation mode has no influence on the rule from (5).

From the construction of G', it works exactly in the same way as G does. □

$L(G, {}_1\Rightarrow) = L(G', {}_5\Rightarrow)$ and $L(G, {}_1\Rightarrow) = L$, therefore $L(G', {}_5\Rightarrow) = L$. Since L is an arbitrary recursively enumerable language, the proof of Theorem 3 is completed. □

4.4 Mode 7

This section investigates mode 7 derivations and proves SCGs with this mode derivations are Turing-complete.

Theorem 4. $SC({}_7\Rightarrow) = \mathbf{RE}$.

Since $SC({}_7\Rightarrow) \subseteq \mathbf{RE}$ follows directly from the Church-Turing thesis, we only have to prove the opposite inclusion.

Proof. Let $G = (V, \Sigma, P, S)$ be the SCG constructed in the proof of Theorem 1. Next, we modify G to a new SCG G' so $L(G, {}_1\Rightarrow) = L(G', {}_7\Rightarrow)$.

Construction. Introduce four new symbols—E,F,G and $|$. Set $N = \{E,F,G,|\}$. Let $G' = (V', \Sigma, P', S)$ be SCG, with $V' = V \cup N$ and P' containing the rules

(1) $(S) \to (FGS_1 11\$|\$11S_2)$;
(2) $(A) \to (C_i(w))$ for $A \to w \in P_i$, where $i = 1, 2$;
(3) $(F) \to (FF)$;
(4) $(a, \$, \$, a) \to (\varepsilon, E, E, \varepsilon)$, for $a = 0, 1$;
(5) $(F, G, E, |, E) \to (G, \$, |, \$, \varepsilon)$;
(6) $(\$) \to (\varepsilon)$, $(G) \to (\varepsilon)$, $(|) \to (\varepsilon)$.

Claim 6. $L(G, {}_1\Rightarrow) = L(G', {}_7\Rightarrow)$.

Proof. The behaviour of context-free rules remains unchanged under mode 7 derivations. Since the rules of G' simulating the derivations of G_1, G_2, respectively, are identical to the ones of G simulating both grammars, for every derivation of G

$$S \ {}_1\Rightarrow^* \ \$w_1 1111 w_2\$ = w$$

where w was generated only using the rules from (1) and(2) and $\text{alph}(w) \cap (V_1 \cup V_2) = \emptyset$, there is

$$S \ {}_7\Rightarrow^* \ FGw_1 11\$|\$11w_2 = w'$$

in G', generated by the corresponding rules from (1) and (2), and vice versa. Without any loss of generality, we can consider such a sentence form in every successful derivation. Additionally, in G

$$w \ {}_1\Rightarrow^* \ v, v \in \Sigma^*$$

if and only if $t'(w_1) = \text{reversal}(w_2)$. Note, then $v = t(w)$. Therefore, we have to prove

$$w' \ {}_4\Rightarrow^* \ v', v' \in \Sigma^*$$

if and only if $t'(w_1) = \text{reversal}(w_2)$. Then obviously $v' = v$ and we can complete the proof by the following claim.

Claim 7. In G', for

$$S \ {}_7\Rightarrow^* \ w = F^i Gw_1\$|\$w_2 E$$

where w was generated only using the rules from (1) through (3) and $\text{alph}(w) \cap (V_1 \cup V_2) = \emptyset$. Then

$$w \ {}_7\Rightarrow^* \ w'$$

where $w' \in \Sigma^*$, if and only if $t'(w_1) = \text{reversal}(w_2)$, for some $i \geq 1$.

The new rule from (3) may potentially arbitrarily multiply the number of Fs to the left. Then, Fs are erasing using the rule from (5). Thus, without any loss of generality, suppose i equals the number of the future usages of the rule from (5).

For better readability, in the next proof we omit all symbols of w_1 from Σ—we consider only nonterminal symbols, which are to be erased.

Proof. If. Suppose $w_1 = \text{reversal}(w_2)$, then $w \ {}_7\Rightarrow^* \ \varepsilon$. We prove this by the induction on the length of w_1, w_2, where $|w_1| = |w_2| = k$. Then, obviously $i = k$. By the construction of G', the least k equals 2, but we prove the claim for all $k \geq 0$.

Basis. Let $k = 0$. Then

$$w = G\$|\$$$

By the rules from (6)

$$G\$|\$ \ {}_7\Rightarrow^* \ \varepsilon$$

and the basis holds.

Induction Hypothesis. Suppose there exists $k \geq 0$ such that the claim holds for all m, where

$$w = F^m G w_1 \$ | \$ w_2, |w_1| = |w_2| = m, 0 \leq m \leq k$$

Induction Step. Consider G' generates w, where

$$w = F^{k+1} G w_1 \$ | \$ w_2, |w_1| = |w_2| = k+1$$

Since $w_1 = \text{reversal}(w_2)$ and $|w_1| = |w_2| = k+1$, $w_1 = w_1' a$, $w_2 = a w_2'$.

The symbols a can be erased by application of the rules from (4) and (5) under several conditions. First, when the rule from (4) is applied, the center for interchanging right hand side strings must be chosen between the two \$s, otherwise both Es appear on the same side of the symbol | and the rule from (5) is not applicable. Next, no 0 or 1 may be skipped, while proceeding in the direction from center to the edges. Finally, when the rule from (5) is applied, the center must be chosen to the left of F, otherwise G is erased and the future application of this rule is excluded.

$$F^{k+1} G w_1' a \$ | \$ a w_2' \ {}_7\!\Rightarrow\ F^{k+1} G w_1' D | D w_2' \ {}_7\!\Rightarrow\ F^k G w_1' \$ | \$ w_2' = w'$$

By induction hypothesis $w' {}_7\!\Rightarrow^* \varepsilon$, which completes the proof.

Only if. Suppose $w_1 \neq \text{reversal}(w_2)$, then, there is no w', where $w \ {}_7\!\Rightarrow^*\ w'$ and $w' = \varepsilon$.

Since $w_1 \neq \text{reversal}(w_2)$, $w_1 = uav$, $w_2 = va'u'$ and $a \neq a'$. Suppose both vs are correctly erased and no symbol is skipped producing the sentential form

$$F^i G u a \$ | \$ a' u'$$

Next the rule from (4) can be applied to erase innermost 0s or 1s. However, since $a \neq a'$, even if the center if chosen properly between the two \$s, there is 0 or 1 between inserted Es and thus unable to be erased, which completes the proof.

We showed, that G' can generate the terminal string from the sentence form w, if only if $t'(w_1) = \text{reversal}(w_2)$, and the claim holds. □

We proved $S \ {}_1\!\Rightarrow^*\ w$, $w \in \Sigma^*$, in G, if and only if $S \ {}_7\!\Rightarrow^*\ w$ in G', hence $L(G, {}_1\!\Rightarrow) = L(G', {}_7\!\Rightarrow)$ and the claim holds. □

Since $L(G, {}_1\!\Rightarrow) = L(G', {}_7\!\Rightarrow)$, $L(G, {}_1\!\Rightarrow) = L$ and L is an arbitrary recursively enumerable language, the proof of Theorem 4 is completed. □

5 Conclusion

The modern trend in information processing is parallel access to typically distributed data, however, traditionally, in formal languages theory automata and

grammars process information in continuous and often sequential way. Such models are not entirely suitable for the study of modern approaches in the data processing, where the data are frequently simultaneously read from and written to the different parts of the memory space, sometimes physically separated.

For modelling the parallel data processing the usage of the scattered context grammars that have been investigated already in a long series of studies, which brought a number of important results—especially their computational completeness—, seems to be appropriate. Though, even the scattered context grammars are not a perfect model of the modern data processing and the whole variety of suitable modifications can be established.

However, the aim of this study was not the attempt to modify the model itself, only the way it generates the terminal string. The main motivation was to break the usual approach of the rewriting and try to divide the process into deletion and insertion, which may not necessarily take place at the same part of the sentence form. The mutual relation of these two now separated actions is then defined by the constraints resulting from the definition of the used derivation mode. Despite the fact that the additional nondeterminism is brought into the computational process, it has been proven that the generative power of the model is not reduced and it is still as powerful as Turing machines.

Acknowledgments

This work was supported by the following grants: BUT FIT FIT-S-14-2299, MŠMT CZ1.1.00/02.0070, and TAČR TE01010415.

References

1. Salomaa, A.: Formal Languages. Academic Press, London (1973)
2. Rozenberg, G., Salomaa, A., eds.: Handbook of Formal Languages, Vol. 1: Word, Language, Grammar. Springer, New York (1997)
3. Fernau, H., Meduna, A.: A simultaneous reduction of several measures of descriptional complexity in scattered context grammars. Information Processing Letters **86**(5) (2003) 235–240
4. Harrison, M.A.: Introduction to Formal Language Theory. 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1978)
5. Meduna, A., Zemek, P.: Regulated Grammars and Their Transformations. Faculty of Information Technology, Brno University of Technology (2010)
6. Meduna, A., Techet, J.: Scattered Context Grammars and their Applications. WIT Press (2010) ISBN: 978-1-84564-426-0.