Final sentential forms

Alexander Meduna
Faculty of Information Technology
Brno University of Technology
Czech Republic
meduna@fit.vutbr.cz

Tomáš Kožár
Faculty of Information Technology
Brno University of Technology
Czech Republic
xkozar02@stud.fit.vutbr.cz

Let G be a context-free grammar, and let F be a *final language* over an alphabet W. A *final sentential form* is any sentential form of G that belongs to F. The strings resulting from the elimination of all nonterminals from W in a final sentential forms are in the *language of G finalized by F* if and only if they contains only terminals.

The language of any context-free grammar finalized by a regular language is context-free. On other hand, it is demonstrated that L is a recursively enumerable language if and only if there exists a propagating context-free grammar G such that L equals the language of G finalized by $\{w\#w^R \mid w \in \{0,1\}^*\}$, where w^R is the reversal of w.

1 Introduction

The present paper introduces and studies *final sentential forms* of context-free grammars. These forms represent the sentential forms in which the sequences of prescribed symbols, possibly including nonterminals, belong to given *final languages*. If all the other symbols are terminals, these final forms are changed to the sentences of the generated languages by simply eliminating all nonterminals in them. Next, we sketch both a practical inspiration and a theoretical reason for introducing this new way of context-free language generation.

I. Indisputably, parsing represents a crucially important application area of ordinary context-free grammars (see Chapters 3 through 5 in [4]) as well as their modified versions, such as regulated grammars (see Section 20.3 in [6]). During the parsing process, the correctness of the source program syntax is often verified before all nonterminals are eliminated; nevertheless, most classically constructed parsers go on eliminating these nonterminals by using erasing rules until only terminals are derived. As a result, the entire parsing process is slowed down uselessly during this closing phase (for a simple, but straightforward illustration of this computational situation, see, for instance, Case Study 14/35 in [4] or Example 4.35 in [1]. Clearly, as the newly introduced way of language generation frees us from a necessity of this closing elimination of all nonterminals, the parsers that make use of it work faster.

II. From a theoretical viewpoint, in the present paper, we achieve a new representation for recursively enumerable languages based upon context-free languages. Admittedly, the theory of formal languages is overflown with many representations for recursively enumerable languages based upon operations over some context-free languages or their spacial cases (see Section 4.1.3 in [7]). Nonetheless, we believe this new representation is of some interest when compared with the previously demonstrated representations. Indeed, each of the already existing representations is demonstrated, in essence, by a proof that has the following general format. (i) First, given any recursively enumerable language L, it represents L by a suitable language model G, such as a phrase structure grammar in a normal form. (ii) Then, from G, it derives both operations and context-free

Submitted to:
NCMA 2023

languages involved in the representation in question. (iii) Finally, it shows that the representation made in this way from G holds true. What is important from our standpoint is that in a proof like this, the specific form of all the operations as well as the languages involved in the representation always depend on G, which generates L. As opposed to this, the new representation achieved in the present paper is much less dependent on L or any of its language models . More precisely, we demonstrate the existence of a unique constant language C defined as $C = \{w\#w^R \mid w \in \{0,1\}^*\}$ and express any recursively enumerable language L by using C and a minimal linear language without any operation. Consequently, C always remains unchanged and, therefore, independent of L or its models. Considering this independency as well as the absence of any operations in the new representation, we believe this representation might be of some interest to formal language theory.

To give a more detailed insight into this study, we first informally recall the notion of an ordinary context-free grammar and its language (this paper assumes a familiarity with formal language theory). A context-free grammar G is based upon a grammatical alphabet V of symbols and a finite set of rules. V is divided into two disjoint subalphabets—the alphabet of terminals T and the alphabet of nonterminals N. Each rule has the form $A \rightarrow x$, where A is a nonterminal and x is a string over V. Starting from a special start nonterminal, G repeatedly rewrites strings according to its rules, and in this way, it generated its sentential forms. Sentential forms that consists only of terminal symbols are called sentences, and the set of all sentences represents the language generated by G.

In this paper, we shorted the generating process sketched above by introducing a final language F over a subalphabet $W \subseteq V$. A *final sentential form* of G is any of the sentential forms in which the sequence of symbols from W belong to F. If in this form, all the symbols from $V - W$ are terminals, the string resulting from eliminating all nonterminals from $N \cap W$ results into a sentence of the generated language $L(G,F)$ finalized by F.

Next, we illustrate the newly introduced concept of final sentential forms by a simple example in linguistic morphology, which studies word formation, such as inflection and compounding, in natural languages.

Example 1. Consider an alphabet Σ of consonants and vowels. Suppose that a morphological study deals with a language L consisting of all possible words over Σ together with their consonant-vowel binary schemes in which every consonant and every vowel are represented by 1 and 0, respectively. Mathematically, $L = \{w\#\sigma(w) \mid w \in \Sigma^+\}$, where σ is the homomorphism from Σ^* to $\{0,1\}^*$ defined as $\sigma(x) = 1$ and $\sigma(y) = 0$ for every consonant x in Σ and every vowel y in Σ, respectively. For instance, considering Σ as the English alphabet, $the\#110 \in L$ while $the\#100 \notin L$. Define the context-free grammar G with following rules.

- $S \rightarrow A\#B, B \rightarrow 0YB, B \rightarrow 0Y, B \rightarrow 1XB, B \rightarrow 1X$,
- $A \rightarrow aAY, A \rightarrow aY$ for all vowels a in Σ,
- $A \rightarrow bAX, A \rightarrow bX$ for all consonants b in Σ,

where the uppercases are nonterminals with S being the start nonterminal, and the other symbols are terminals. Set $W = \{X,Y,\#\}$ and $F = \{w\#w^R \mid w \in \{X,Y\}^*\}$. For instance, take this step-by-step derivation

$$\begin{aligned} S &\Rightarrow A\#B \Rightarrow tAX\#B \Rightarrow thAXX\#B \Rightarrow theYXX\#B \\ &\Rightarrow theYXX\#1XB \Rightarrow theYXX\#1X1XB \Rightarrow theYXX\#1X1X0Y \end{aligned}$$

In $theYXX\#1X1X0Y, YXX\#XXY \in F$, and apart from $X, Y, \# \in W$, $theYXX\#1X1X0Y$ contains only terminals. The removal of all Xs and Ys in $theYXX\#1X1X0Y$ results into $the\#110$, which thus belongs to $L(G,F)$. Clearly, $L(G,F) = L$.

As its main result, the present paper demonstrated that L is a recursively enumerable language if and only if $L = L(G, \{w\#w^R \,|\, w \in \{0,1\}^*\})$, where G is a context-free grammar; observe that in this equivalence, the final language $\{w\#w^R \,|\, w \in \{0,1\}^*\}$ remains constant independently of L. On the other hand, the paper also proves that any $L(G,F)$ is context-free if G is a context-free grammar and F is regular.

The rest of the paper is organized as follows. First, Section 2 gives all the necessary terminology and defines the new notions, informally sketched in this introduction. Then, Section 3 establishes the above-mentioned results and points out an open problem related to the present study.

2 Preliminaries and Definitions

This paper assumes that the reader is familiar with the language theory (see [5]).

For a set, Q, $card(Q)$ denotes the cardinality of Q. For an alphabet, V, 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 w^R denotes the length of w and the reversal of w, respectively. For every $i \in \{0, 1, ..., |w|\}$, $suffix(w,i)$ denotes the suffix of w of length i; analogously, $prefix(w,i)$ denotes the prefix of w of length i. Let W be an alphabet and ω be a homomorphism from V^* to W^* (see [5] for the definition of homomorphism); ω is a *weak identity* if $\omega(a) \in \{a, \varepsilon\}$ for all $a \in V$.

A *context-free grammar* (CFG for short) is quadruple $G = (V,T,P,S)$, where V is an alphabet, $T \subseteq V$, $P \subseteq (V-T) \times V^*$ is finite, and $S \in V-T$. Set $N = V-T$. V,T,N,P, and S are referred to as the total alphabet, the terminal alphabet, the nonterminal alphabet, the set of rules, and the start symbol of G, respectively. Instead of $(A,x) \in P$, we write $A \rightarrow x \in P$ throughout. For brevity, we often denote $A \rightarrow x$ by a unique label p as $p : A \rightarrow x$, and we briefly use p instead of $A \rightarrow x$ under this denotation. For every $p : A \rightarrow x \in P$, the *left-hand side of* p is defined as $lhs(p) = X$. G is *propagating* if $A \rightarrow x \in P$ implies $x \in V^+$. G is *linear* if no more than one nonterminal appears on the right-hand side of any production in P. Furthermore, a linear grammar G is *minimal* (see page 76 in [8]) if $N = \{S\}$ and $S \rightarrow \# \in P, \# \in T$, is the only production with no nonterminal on the right side, whereas it is assumed that # does not occur in any other production. In this paper, a minimal linear grammar G is called a *palindromial grammar* if $card(P) \geq 2$, and every rule of the form $S \rightarrow xSy$, where $x,y \in T^*$, satisfies $x = y, x, y \in T$. For instance, $H = (\{S,0,1,\#\}, \{0,1,\#\}, \{S \rightarrow 0S0, S \rightarrow 1S1, S \rightarrow \#\}, S)$ is a palindromial grammar.

For every $u,v \in V^*$ and $p : A \rightarrow x \in P$, write $uAv \Rightarrow uxv\,[p]$ or, simply, $uAv \Rightarrow uxv$; $\Rightarrow$ is called the *direct derivation* relation over V^*. For $n \geq 0, \Rightarrow^n$ denotes the n-th power of $\Rightarrow$. Furthermore, $\Rightarrow^+$ and $\Rightarrow^*$ denote the transitive closure and the transitive-reflexive closure of $\Rightarrow$, respectively. Let $\phi(G) = \{w \in V^* \,|\, S \Rightarrow^* w\}$ denotes the set of all *sentential forms* of G. The language of G is denoted by $L(G)$ and defined as $L(G) = T^* \cap \phi(G)$. For example, $L(H) = \{w\#w^R \,|\, w \in \{0,1\}^*\}$, where H is defined as above.

Let $G = (V,T,P,S)$ be a CFG and $W \subseteq V$. Define the weak identity ${}_W\omega$ from V^* to W^* as ${}_W\omega(X) = X$ for all $X \in W$, and ${}_W\omega(X) = \varepsilon$ for all $X \in V-W$. Let $F \subseteq W^*$. Set

$$\phi(G,F) = \{x \,|\, x \in \phi(G), {}_W\omega(x) \in F\}$$
$$L(G,F) = \{{}_T\omega(y) \,|\, y \in \phi(G,F), {}_{(N-W)}\omega(y) = \varepsilon\}.$$

$\phi(G,F)$ and $L(G,F)$ are referred to as the set of *sentential forms of G finalized by F* and the *language of G finalized by F* respectively. Members of $\phi(G,F)$ are called *final sentential forms*. $\mathbf{REG}, \mathbf{PAL}, \mathbf{LIN}, \mathbf{CF}$ and $\mathbf{RE}$ denote the families of regular, palindromial, linear, context-free, and recursively enumerable languages, respectively. Observe that

$$\mathbf{REG} \cap \mathbf{PAL} = \emptyset \text{ and } \mathbf{REG} \cup \mathbf{PAL} \subset \mathbf{LIN}.$$

Set

$$\mathbf{CF_{PAL}} = \{L(G,F) \,|\, G \text{ is a CFG }, F \in \mathbf{PAL}\}$$
$$\mathbf{CF_{REG}} = \{L(G,F) \,|\, G \text{ is a CFG }, F \in \mathbf{REG}\}$$

Example 2. Set $I = \{i(x) \,|\, x \in \{0,1\}^+\}$, where $i(x)$ denotes the integer represented by x in the standard way; for instance, $i(011) = 3$. Consider

$$L = \{u\#v \,|\, u,v \in \{0,1\}^+, i(u) > i(v) \text{ and } |u| = |v|\}.$$

Next, we define a CFG G and $F \in \mathbf{PAL}$ such that $L = L(G,F)$. Let $G = (V,T,P,S)$ be a context-free grammar. Set $V = \{S,X,\overline{X},Y,\overline{Y},A,B,C,D,0,1,\#\}, T = \{0,1,\#\}$, and add following rules to P

- $S \to X\#\overline{X}$
- $X \to 1AX, X \to 0BX, X \to 1CY, X \to 1C$
- $\overline{X} \to 1\overline{X}A, \overline{X} \to 0\overline{X}B, \overline{X} \to 0\overline{Y}C, \overline{X} \to 0C$
- $Y \to \alpha DY, Y \to \alpha D, \overline{Y} \to \alpha\overline{Y}D, \overline{Y} \to \alpha D$ for all $\alpha \in \{0,1\}$.

Set $W = \{A,B,C,D,\#\}$ and $F = \{w\#w^R \,|\, w \in \{A,B,C,D\}^+ \text{ and } n \geq 1\}$. Observe that $F = L(H)$, where $H = (\{S,A,B,C,D,\#\}, \{A,B,C,D,\#\}, \{S \to ASA, S \to BSB, S \to CSC, S \to DSD, S \to \#\}, S)$ is a palindromial grammar. Therefore, $F \in \mathbf{PAL}$. For instance, take this step-by-step derivation

$$\begin{aligned} S &\Rightarrow X\#\overline{X} \Rightarrow 1AX\#\overline{X} \Rightarrow 1A0BX\#\overline{X} \Rightarrow 1A0B1CY\#\overline{X} \Rightarrow 1A0B1C0D\#\overline{X} \\ &\Rightarrow 1A0B1C0D\#1\overline{X}A \Rightarrow 1A0B1C0D\#10\overline{X}BA \Rightarrow 1A0B1C0D\#100\overline{X}CBA \\ &\Rightarrow 1A0B1C0D\#1001\overline{Y}DCBA \Rightarrow 1A0B1C0D\#1001DCBA \end{aligned}$$

in G. Notice that ${}_W\omega(1A0B1C0D\#1001DCBA) \in F$, and ${}_T\omega(1A0B1C0D\#1001DCBA) \in L(G,F)$.As obvious, $L = L(G,F)$.

A *queue grammar* (see [2]) is a sextuple, $Q = (V,T,U,D,s,P)$, where V and U are alphabets satisfying $V \cap U = s$, $T \subseteq V$, $D \subseteq U$, $s \in (V-T)(U-D)$, and $P \subseteq (V \times (U-D)) \times (V^* \times U)$ is a finite relation such that for for every $a \in V$, there exists an element $(a,b,z,c) \in P$. If $u,v \in V^*U$ such that $u = arb; v = rzc; a \in V; r,z \in V^*; b,c \in U$; and $(a,b,z,c) \in P$, then $u \Rightarrow v\ [(a,b,z,c)]$ in G or, simply, $u \Rightarrow v$. In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The language of Q, $L(Q)$, is defined as $L(Q) = \{w \in T^* \,|\, s \Rightarrow^* wf \text{ where } f \in D\}$. A *left-extended queue grammar* is a sextuple, $Q = (V,T,U,D,s,P)$, where V,T,U,D, and s have the same meaning as in a queue grammar. $P \subseteq (V \times (U-D)) \times (V^* \times U)$ is a finite relation (as opposed to an ordinary queue grammar, this definition does not require that for every $a \in V$, there exists an element $(a,b,z,c) \in P$). Furthermore, assume that $\# \notin V \cup U$. If $u,v \in V^*\{\#\}V^*U$ so that $u = w\#arb; v = wa\#rzc; a \in V; r,z,w \in V^*; b,c \in U$;

and $(a,b,z,c) \in P$, then $u \Rightarrow v[(a,b,z,c)]$ in G or, simply $u \Rightarrow v$. In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The language of Q, $L(Q)$, is defined as $L(Q) = \{v \in T^* \mid \#s \Rightarrow^* w\#vf$ for some $w \in V^*$ and $f \in D\}$. Less formally, during every step of a derivation, a left-extended queue grammar shifts the rewritten symbol over #; in this way, it records the derivation history, which plays a crucial role in the proof of Lemma 5 in the next section.

A *deterministic finite automaton* (DFA for short) is quintuple $M = (Q, \Sigma, R, s, F)$, where Q is a finite *set of states*, Σ is an *alphabet of input symbols*, $Q \cap \Sigma = \emptyset, s \in Q$ is a special state called the *start state*, and $F \subseteq Q$ is a *set of final states* in M. R is a total function from $Q \times \Sigma$ to Q. Instead of $R(q,a) = p$, we write $qa \to p$, where $q, p \in Q$ and $a \in \Sigma \cup \{\varepsilon\}$;$R$ is referred to as the *set of rules* in M. For any $x \in \Sigma^*$ and $qa \to p \in R$, we write $qax \Rightarrow px$. The *language of* $M, L(M)$, is defined as $L(M) = \{w \mid w \in \Sigma^*, sw \Rightarrow^* f, f \in F\}$, where $\Rightarrow^*$ denotes the reflexive-transitive closure of $\Rightarrow$. Recall that DFAs characterize **REG** (see page 29 in [5]).

3 Results

In this section, we show that every language generated by a context-free grammar finalized by a regular language is context-free (see Theorem 2). On the other hand, we prove that every recursively enumerable language can be generated by a propagating context-free grammar finalized by a unique palindromial language of this form—$\{w\#w^R \mid w \in \{0,1\}^*\}$ (see Theorem 9).

Lemma 1. Let $G = (V, T, P, S)$ be any CFG and $F \in$ **REG**. Then, $L(G, F) \in$ **CF**.

Proof. Let $G = (V, T, P, S)$ be any CFG and $F \in$ **REG**. Let $F = L(M)$, where $M = (Q, W, R, q_s, Q_F)$ is a deterministic finite automaton.

Construction. Introduce $U = \{\langle paq \rangle \mid p, q \in Q, a \in V\} \cup \{\langle q_s S Q_F \rangle\}$. From G and M, construct a new CFG H such that $L(H) = L(G, F)$ in the following way. Set

$$H = (\overline{V}, T, \overline{P}, \langle q_s S Q_F \rangle)$$

The components of H are constructed as follows. Set $\overline{V} = V \cup U$. Construct $\overline{P}$ as follows:

(0) Add $\langle q_s S Q_F \rangle \to \langle q_s S q_f \rangle$ for all $q_f \in Q_F$.

(1) Let $A \to y_0 X_1 y_1 X_2 ... X_n y_n \in P$, where $A \in V - T, y_i \in (V - W)^*$ and $X_j \in V, 0 \leq i \leq n, 1 \leq j \leq n$, for some $n \geq 1$;
then, add $\langle q_1 A q_{n+1} \rangle \to y_0 \langle q_1 X_1 q_2 \rangle y_1 \langle q_2 X_2 q_3 \rangle ... \langle q_n X_n q_{n+1} \rangle y_n$ to $\overline{P}$, for all $q_1, q_2, ..., q_{n+1} \in Q$.

(2) Let $A \to \alpha \in P$, where $A \in V - (T \cup W), \alpha \in (V - W)^*$;
then, add $A \to \alpha$ to $\overline{P}$.

(3) Let $\langle paq \rangle \in U$, where $a \in W \cap T, pa \to q \in R$;
then, add $\langle paq \rangle \to a$ to $\overline{P}$.

(4) Let $\langle pBq \rangle \in U$, where $pB \to q \in R, B \in W \cap (V - T)$;
then, add $\langle pBq \rangle \to \varepsilon$ to $\overline{P}$.

To prove $L(G, F) = L(H)$, we first prove $L(H) \subseteq L(G, F)$; then, we establish $L(G, F) \subseteq L(H)$. To demonstrate $L(H) \subseteq L(G, F)$, we first make three observations—(i) through (iii)—concerning every derivation of the form $\langle q_s S q_f \rangle \Rightarrow^* y$ with $y \in T^*$.

(i) By using rules constructed in (1) and (2), H makes a derivation of the form

$$\langle q_s S q_f \rangle \Rightarrow^* x_0 \langle q_1 Z_1 q_2 \rangle x_1 \ldots \langle q_n Z_n q_{n+1} \rangle x_n$$

where $x_i \in (T - W)^*, 0 \leq i \leq n, \langle q_j Z_j q_{j+1} \rangle \in U, Z_j \in W, 1 \leq j \leq n, q_1 = q_s, q_{n+1} = q_f, q_1 \ldots q_{n+1} \in Q, q_f \in Q_F$.

(ii) If

$$\langle q_s S q_f \rangle \Rightarrow^* x_0 \langle q_1 Z_1 q_2 \rangle x_1 \ldots \langle q_n Z_n q_{n+1} \rangle x_n$$

in H, then

$$S \Rightarrow^* x_0 Z_1 x_1 \ldots Z_n x_n$$

in G, where all the symbols have the same meaning as in (i).

(iii) Let H make

$$x_0 \langle q_1 Z_1 q_2 \rangle x_1 \ldots \langle q_n Z_n q_{n+1} \rangle x_n \Rightarrow^* y$$

by using rules constructed in (3) and (4), where $y \in T^*$, and all the other symbols have the same meaning as in (i). Then, for all $1 \leq j \leq n, q_j Z_j \rightarrow q_{j+1} \in R, y = x_0 U_1 x_1 \ldots U_n x_n$, where $U_j = {}_T\omega(Z_j)$. As $q_j Z_j \rightarrow q_{j+1} \in R, 1 \leq j \leq n, q_1 = q_s$ and $q_{n+1} = q_f, q_f \in Q_F$, we have $Z_1 \ldots Z_n \in L(M)$.

Based on (i) through (iii), we are now ready to prove $L(H) \subseteq L(G, F)$. Let $y \in L(H)$. Thus $\langle q_s S Q_F \rangle \Rightarrow^* y, y \in T^*$ in H. As H is an ordinary CFG, we can always rearrange the applications of rules during $\langle q_s S Q_F \rangle \Rightarrow^* y$ in such a way that

$$\begin{array}{llll} \langle q_s S Q_F \rangle & \Rightarrow & \langle q_s S q_f \rangle & (\alpha) \\ & \Rightarrow^* & x_0 \langle q_1 Z_1 q_2 \rangle x_1 \ldots \langle q_m Z_m q_{m+1} \rangle x_m & (\beta) \\ & \Rightarrow^* & y & (\gamma) \end{array}$$

so that during (α), only a rule from (0) is used, during β only rules from (1) and (2) are used, and during (γ) only rules from (3) and (4) are used. Recall that $Z_1 Z_2 \ldots Z_n \in F$ (see (iii)). Consequently, ${}_W\omega(x_0 Z_1 x_1 \ldots Z_n x_n) \in F$. From (3), (4), (ii), and (iii), it follows that

$$S \Rightarrow^* x_0 Z_1 x_1 \ldots x_{n-1} Z_n x_n \text{ in } G$$

Thus, as $L(M) = F$, we have $y \in L(G, F)$, so $L(H) \subseteq L(G, F)$.

To prove $L(G, F) \subseteq L(H)$, take any $y \in L(G, F)$. Thus,

$$\begin{array}{l} S \Rightarrow^* x_0 Z_1 x_1 \ldots x_{n-1} Z_n x_n \text{ in G, and} \\ y = {}_T\omega(x_0 Z_1 x_1 \ldots x_{n-1} Z_n x_n) \text{ with } Z_1 \ldots Z_n \in F, \end{array}$$

where $x_i \in (T-W)^*, 0 \le i \le n, Z_j \in W, 1 \le j \le n$. As $Z_1 ... Z_n \in F$, we have $q_1 Z_1 \rightarrow q_2, ..., q_n Z_n \rightarrow q_{n+1} \in R, q_1 ... q_{n+1} \in Q, q_1 = q_s, q_{n+1} = q_f, q_f \in Q_F$. Consequently, from (0) through (4) of the Construction, we see that

$$\begin{aligned}\langle q_s S Q_f \rangle &\Rightarrow \langle q_s S q_f \rangle \\ &\Rightarrow^* x_0 Z_1 x_1 ... Z_n x_n \\ &\Rightarrow^* x_0 U_1 x_1 ... U_n x_n\end{aligned}$$

where $U_j = {}_T\omega(Z_j), 1 \le j \le n$. Hence, $y \in L(H)$, so $L(G,F) \subseteq L(H)$.

Thus, $L(G,F) = L(H)$. □

Theorem 2. $\mathbf{CF_{REG}} = \mathbf{CF}$.

Proof. Clearly, $\mathbf{CF} \subseteq \mathbf{CF_{REG}}$. From Lemma 1, $\mathbf{CF_{REG}} \subseteq \mathbf{CF}$. Thus, Theorem 2 holds true. □

Now, we prove that by using constant palindromial language $C = \{w\#w^R \,|\, w \in \{0,1\}^*\}$ to finalize propagating context-free grammar, we can represent any recursively enumerable language.

Lemma 3. Let $L \in \mathbf{RE}$. Then, there exists a left-extended queue grammar Q satisfying $L(Q) = L$.

Proof. See *Lemma 1* in [3]. □

Lemma 4. Let H be a left-extended queue grammar. Then, there exists a left-extended queue grammar, $Q = (V,T,U,D,s,R)$, such that $L(H) = L(Q)$ and every $(a,b,x,c) \in R$ satisfies $a \in V-T, b \in U-D, x \in ((V-T)^* \cup T^*)$ and $c \in U$.

Proof. See *Lemma 2* in [3]. □

Lemma 5. Let $Q = (V,T,U,D,s,R)$ be a left-extended queue grammar. Then, $L(Q) = L(G, \{w\#w^R \,|\, w \in \{0,1\}^*\})$, where G is a CFG.

Proof. Without any loss of generality, assume that Q satisfies the properties described in Lemma 4 and that $\{0,1\} \cap (V \cup U) = \emptyset$. For some positive integer, n, define an injection, ι, from Ψ^* to $(\{0,1\}^n - 1^n)$, where $\Psi = \{ab \,|\, (a,b,x,c) \in R, a \in V-T, b \in U-D, x \in (V-T)^* \cup T^*, c \in U\}$ so that ι is an injective homomorphism when its domain is extended to Ψ^*; after this extension, ι thus represents an injective homomorphism from Ψ^* to $(\{0,1\}^n - 1^n)^*$(a proof that such an injection necessarily exists is simple and left to the reader). Based on ι, define the substitution, ν from V to $(\{0,1\}^n - 1^n)$ as $\nu(a) = \{\iota(aq) \,|\, q \in U\}$ for every $a \in V$. Extend domain of ν to V^*. Furthermore, define the substitution, μ, from U to $(\{0,1\}^n - 1^n)$ as $\mu(q) = \{\iota(aq)^R \,|\, a \in V\}$ for every $q \in U$. Extend the domain of μ to U^*. Set $J = \{\langle p,i \rangle \,|\, p \in U-D \text{ and } i \in \{1,2\}\}$.

Construction. Next, we introduce a context-free grammar G so that $L(Q) = L(G, \{w\#w^R \,|\, w \in \{0,1\}^*\})$. Let $G = (\overline{V}, T, P, S)$, where $\overline{V} = J \cup \{0,1,\#\} \cup T$. Construct P in the following way. Initially, set $P = \emptyset$; then, perform the following steps 1 through 5.

1. if $(a,q,y,p) \in R$, where $a \in V-T, p,q \in U-D, y \in (V-T)^*$ and $aq = s$,
   then add $S \rightarrow u\langle p,1 \rangle v$ to P, for all $u \in \nu(y)$ and $v \in \mu(p)$;
2. if $(a,q,y,p) \in R$, where $a \in V-T, p,q \in U-D$ and $y \in (V-T)^*$,
   then add $\langle q,1 \rangle \rightarrow u\langle p,1 \rangle v$ to P, for all $u \in \nu(y)$ and $v \in \mu(p)$;
3. for every $q \in U-D$, add $\langle q,1 \rangle \rightarrow \langle q,2 \rangle$ to P;
4. if $(a,q,y,p) \in R$, where $a \in V-T, p,q \in U-D, y \in T^*$,
   then add $\langle q,2 \rangle \rightarrow y\langle p,2 \rangle v$ to P, for all $v \in \mu(p)$;

5. if $(a,q,y,p) \in R$, where $a \in V-T, q \in U-D, y \in T^*$, and $p \in D$, then add $\langle q,2\rangle \to y\#$ to P.

Set $W = \{0,1,\#\}$ and $\Omega = \{xy\#z \in \phi(G) \mid x \in \{0,1\}^+, y \in T^*, z = x^R\}$.

Claim 6. Every $h \in \Omega$ is generated by G in this way

$$
\begin{array}{ll}
 & S \\
\Rightarrow & g_1\langle q_1,1\rangle t_1 \Rightarrow g_2\langle q_2,1\rangle t_2 \Rightarrow \ldots \Rightarrow g_k\langle q_k,1\rangle t_k \Rightarrow g_k\langle q_k,2\rangle t_k \\
\Rightarrow & g_k y_1\langle q_{k+1},2\rangle t_{k+1} \Rightarrow g_k y_1 y_2\langle q_{k+2},2\rangle t_{k+2} \Rightarrow \ldots \Rightarrow g_k y_1 y_2 \ldots y_{m-1}\langle q_{k+m-1},2\rangle t_{k+m-1} \\
\Rightarrow & g_k y_1 y_2 \ldots y_{m-1} y_m \# t_{k+m}
\end{array}
$$

in G, where $k,m \geq 1; q_1,\ldots,q_{k+m-1} \in U-D; y_1,\ldots,y_m \in T^*; t_i \in \mu(q_i \ldots q_1)$ for $i = 1,\ldots,k+m; g_j \in \nu(d_1\ldots d_j)$ with $d_1,\ldots,d_j \in (V-T)^*$ for $j = 1,\ldots,k; d_1\ldots d_k = a_1\ldots a_{k+m}$ with $a_1,\ldots,a_{k+m} \in V-T$ (that is, $g_k \in \nu(a_1\ldots a_{k+m})$ with $g_k = (t_{k+m})^R); h = y_1 y_2 \ldots y_{m-1} y_m$.

Proof. Examine the construction of P. Observe that every derivation begins with an application of a production having S on its left-hand side. Set 1-$J = \{\langle p,1\rangle \mid p \in U\}$, 2-$J = \{\langle p,2\rangle \mid p \in U\}$, 1-$P = \{p \mid p \in P$ and $lhs(p) \in$ 1-$J\}$, 2-$P = \{p \mid p \in P$ and $lhs(p) \in$ 2-$J\}$. Observe that in every successful derivation of h, all applications of productions from 1-P precede the applications of productions from 2-P. Thus, the generation of h can be expressed as

$$
\begin{array}{ll}
 & S \\
\Rightarrow & g_1\langle q_1,1\rangle t_1 \Rightarrow g_2\langle q_2,1\rangle t_2 \Rightarrow \ldots \Rightarrow g_k\langle q_k,1\rangle t_k \Rightarrow g_k\langle q_k,2\rangle t_k \\
\Rightarrow & g_k y_1\langle q_{k+1},2\rangle t_{k+1} \Rightarrow g_k y_1 y_2\langle q_{k+2},2\rangle t_{k+2} \Rightarrow \ldots \Rightarrow g_k y_1 y_2 \ldots y_{m-1}\langle q_{k+m-1},2\rangle t_{k+m-1} \\
\Rightarrow & g_k y_1 y_2 \ldots y_{m-1} y_m \# t_{k+m}
\end{array}
$$

where all the involved symbols have the meaning stated in Claim 6. □

Claim 7. Every $h \in L(Q)$ is generated by Q in this way

$$
\begin{array}{lll}
 & \#a_0q_0 & \\
\Rightarrow & a_0\#x_0q_1 & [(a_0,q_0,z_0,q_1)] \\
\Rightarrow & a_0a_1\#x_1q_2 & [(a_1,q_1,z_1,q_2)] \\
\ldots & & \\
\Rightarrow & a_0a_1\ldots a_k\#x_kq_{k+1} & [(a_k,q_k,z_k,q_{k+1})] \\
\Rightarrow & a_0a_1\ldots a_ka_{k+1}\#x_{k+1}q_{k+2} & [(a_{k+1},q_{k+1},y_1,q_{k+2})] \\
\ldots & & \\
\Rightarrow & a_0a_1\ldots a_ka_{k+1}\ldots a_{k+m-1}\#x_{k+m-1}y_1\ldots y_{m-1}q_{k+m} & [(a_{k+m-1},q_{k+m-1},y_{m-1},q_{k+m})] \\
\Rightarrow & a_0a_1\ldots a_ka_{k+1}\ldots a_{k+m}\#y_1\ldots y_mq_{k+m+1} & [(a_{k+m},q_{k+m},y_m,q_{k+m+1})]
\end{array}
$$

where $k,m \geq 1$, $a_i \in V-T$ for $i = 0,\ldots,k+m$, $x_j \in (V-T)^*$ for $j = 1,\ldots,k+m$, $s = a_0q_0$, $a_jx_j = x_{j-1}z_j$ for $j = 1,\ldots,k$, $a_1\ldots a_kx_{k+1} = z_0\ldots z_k$, $a_{k+1}\ldots a_{k+m} = x_k$, $q_0,q_1,\ldots,q_{k+m} \in U-D$ and $q_{k+m+1} \in D$, $z_1,\ldots,z_k \in (V-T)^*$, $y_1,\ldots y_m \in T^*$, $h = y_1y_2\ldots y_{m-1}y_m$.

Proof. Recall that Q satisfies the properties given in Lemma 4. These properties implies that Claim 7 holds true. □

Claim 8. $L(G,\{w\#w^R \mid w \in \{0,1\}^*\}) = L(Q)$.

Proof. To prove that $L(G,F) \subseteq L(Q)$, take any $h \in \Omega$ generated in the way described in Claim 6. From ${}_W\omega(h) \in \{w\#w^R \mid w \in \{0,1\}^*\}$, it follows that $xy\#z$ with $z = x^R$ where $x = g_k, y = y_1\ldots y_m, z = t_{k+m}$. At this point R contains (a_0, q_0, z_0, q_1), ...,(a_k, q_k, z_k, q_{k+1}), $(a_{k+1}, q_{k+1}, y_1, q_{k+2})$, ..., $(a_{k+m-1}, q_{k+m-1}, y_{m-1}, q_{k+m})$, $(a_{k+m}, q_{k+m}, y_m, q_{k+m+1})$, where z_1, ..., $z_k \in (V-T)^*$, and y_1, ..., $y_m \in T^*$. Then, Q makes the generation of ${}_T\omega(h)$ in the way described in Claim 7. Thus ${}_T\omega(h) \in L(Q)$.

To prove $L(Q) \subseteq L(G, \{w\#w^R \,|\, w \in \{0,1\}^*\})$, take any $h \in L(Q)$. Recall that h is generated in the way described in Claim 7. Consider the rules used in this generation. Furthermore, consider the definition of ν and μ. Based on this consideration, observe that from the construction of P, it follows that $S \Rightarrow^* oh\#\overline{o}$ in G for some $o, \overline{o} \in \{0,1\}^+$ with $\overline{o} = o^R$. Thus, ${}_W\omega(oh\#\overline{o}) \in \{w\#w^R \,|\, w \in \{0,1\}^*\}$, so consequently, $h \in L(G, \{w\#w^R \,|\, w \in \{0,1\}^*\})$. □

Claims 6 through 8 imply that Lemma 5 holds true.

Theorem 9. A language $L \in \mathbf{RE}$ if and only if $L = L(G, \{w\#w^R \,|\, w \in \{0,1\}^*\})$, where G is a propagating CFG.

Proof. This theorem follows from Lemmas 3 through 5.

Corollary 10. $\mathbf{RE} = \mathbf{CF_{PAL}}$.

Consider $\{w\#w^R \,|\, w \in \{0,1\}^*\}$ without #—that is $\{ww^R \,|\, w \in \{0,1\}^*\}$. On the one hand, this language is out of $\mathbf{CF_{PAL}}$ because the central symbol # does not occur in it. On the other hand, it is worth pointing out that Theorem 9 can be based upon this purely binary language as well.

Corollary 11. A language $L \in \mathbf{RE}$ if and only if $L = L(G, \{ww^R \,|\, w \in \{0,1\}^*\})$, where G is propagating.

Proof. Prove this corollary by analogy with the way Theorem 9 is demonstrated.

Before closing this paper, we point out an open problem. As its main results, the paper has demonstrated that every recursively enumerable language can be generated by a propagating context-free grammar G finalized by $\{w\#w^R \,|\, w \in \{0,1\}^*\}$ (see Theorem 9). Can this results be established with G having a limited number of nonterminals and/or productions?

References

[1] Alfred V. Aho, Monica S. Lam, Ravi Sethi & Jeffrey D. Ullman (2006): *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Addison-Wesley Longman Publishing Co., Inc., USA.

[2] H. C. M. Kleijn & G. Rozenberg (1983): *On the Generative Power of Regular Pattern Grammars*. Acta Informatica 20, pp. 391–411.

[3] Alexander Meduna (2000): *Generative Power of Three-Nonterminal Scattered Context Grammars*. Theoretical Computer Science 2000(246), pp. 279–284.

[4] Alexander Meduna (2008): *Elements of Compiler Design*. Taylor and Francis, Taylor & Francis Informa plc.

[5] Alexander Meduna (2014): *Formal Languages and Computation*. Taylor and Francis, Taylor & Francis Informa plc, doi:10.1201/b16376.

[6] Alexander Meduna & Petr Zemek (2014): *Regulated Grammars and Automata*. Springer US.

[7] G. Rozenberg & A. Salomaa, editors (1997): *Handbook of Formal Languages, Vol. 1: Word, Language, Grammar*. Springer, New York.

[8] Arto Salomaa (1973): *Formal Languages*. ACM monograph series, Academic Press.