Complete CF(¢,$)-grammars, Pumping Reductions, and Regularity.

František Mráz[1,*], Dana Pardubská[2], Martin Plátek[1,*], Martin Procházka[1] and Marta Vomlelová[1]

[1]Charles University, Department of Computer Science, Malostranské nám. 25, 118 00 Praha 1, Czech Republic
[2]Comenius University in Bratislava, Department of Computer Science, Mlynská Dolina, 84248 Bratislava, Slovakia

Abstract

Complete CF(¢,$)-grammar inspired by linguistic techniques can serve as a tool for studying the class of context-free languages that are closed under complement. Such grammar generates a complementary pair of context-free languages. Here, we introduce a restricted version of complete CF(¢,$)-grammars called PS-free pumping CF(¢,$)-grammars, which satisfy restrictions that extend the conditions of the pumping lemma for regular languages. PS-free pumping CF(¢,$)-grammars generate regular languages only. However, the conditions put on PS-free pumping CF(¢,$)-grammars are sufficient but not necessary for regularity.

1. Introduction

Since Chomsky's time, formal syntax for linguistics has been interested in the weak equivalence of formal grammars (equivalence by recognizing the same languages), and rather more in some types of a so-called strong equivalence. For lexicalized types of syntax, strong equivalence based on analysis by reduction is suitable, see, e.g., [1]. Here we use some tools that are different from restarting automata, see, e.g., [2], to develop new techniques for the study of analysis by reduction. This paper establishes some properties of reduction analysis that characterize the regularity of complete CF(¢, $)-grammars.

In [3], a complete CF(¢,$)-grammar was introduced, as a generalization and enhancement of previously introduced LR(¢,$)-grammars. Complete CF(¢,$)-grammars can serve as a tool to study the class of context-free languages that are closed under complement. Recall that the class of context-free languages is the only class from the Chomsky hierarchy that is not closed under complement. A complete CF(¢,$)-grammar G_C is a context-free grammar with two parts that generate acceptance and rejection languages. The acceptance and rejection languages are complementary.

Complete CF(¢,$)-grammars are used here to model correctness and error-preserving analysis by (pumping) reductions on each word over its terminal alphabet. Analysis by reduction is a notion used in linguistics; see, e.g. [1, 4]. It involves stepwise simplifying an input word (sentence, text, or discourse in linguistic terms) by removing at most two continuous parts of the current word while preserving its correctness and/or incorrectness. Each simplification step corresponds to removing portions of the current word that can be "pumped" according to the pumping lemma for context-free languages [5], and thus works only with terminals.

The paper [6] studied conditions that guarantee that a complete CF(¢,$)-grammar generates acceptance and rejection languages that are not regular. Here, we investigate conditions that guarantee the regularity of the acceptance and rejection languages of a complete CF(¢,$)-grammar.

We introduce a PS-free pumping property of complete CF(¢,$)-grammars. PS-free pumping property partially resembles the pumping lemma for regular languages by requiring that each long enough word can be simplified inside a prefix or suffix of a limited size. Additionally, it involves the independence of

ITAT'25: Information Technologies – Applications and Theory, September 26–30, 2025, Telgárt, Nízke Tatry, Slovakia
*Corresponding author.
frantisek.mraz@mff.cuni.cz (F. Mráz); pardubska@dcs.fmph.uniba.sk (D. Pardubská); martin.platek@mff.cuni.cz (M. Plátek); martproc@gmail.com (M. Procházka); marta.vomlelova@mff.cuni.cz (M. Vomlelová)
0000-0001-3869-3340 (F. Mráz); 0000-0001-9383-8117 (D. Pardubská); 0000-0003-3147-6442 (M. Plátek); 0000-0001-9104-804X (M. Vomlelová)

reductions in the prefix and suffix of a word. We show that each PS-free pumping CF(¢,$)-grammar G induces a language equivalence relation with finite index such that the acceptance language of G is one of the equivalence classes. Then, by applying Myhill-Nerode theorem [5, Theorem 3.1], we get that both acceptance and rejection languages of G are regular.

The next section defines the basic notions, introduces complete CF(¢,$)-grammars, and reviews their known properties. Section 3 introduces PS-free pumping grammars and proves the main result of the paper stating that each PS-free pumping grammar has regular acceptance and rejection languages. Section 4 shows examples that PS-free pumping grammar need not be one-sided (informally, its pumping reductions can remove only one segment of a word in one simplification step), and that a complete grammar generating a regular language need not be PS-free pumping. The last section contains conclusions and an outlook for future research.

2. Basic notions and results

An alphabet is an arbitrary finite set of elements called symbols. A word w over the alphabet Σ is a finite sequence of symbols from Σ. The set of all words over the alphabet Σ is denoted as Σ^*. If u and v are words, uv or $u \cdot v$ denotes their concatenation. By $|w|$ we denote the length of the word, that is, the number of symbols in w. The length of the empty word λ is 0.

A context-free grammar is a system $G = (N, \Sigma, S, R)$, where N is an alphabet of nonterminals, Σ is an alphabet of input symbols called terminals ($N \cap \Sigma = \emptyset$), $S \in N$ is an initial nonterminal, and R is a finite subset of $N \times (N \cup \Sigma)^*$, R is called a set of rules and its elements are written in the form $X \to \alpha$, where $X \in N$ and $\alpha \in (N \cup \Sigma)^*$.

We say that a word $u \in (N \cup \Sigma)^*$ can be rewritten into a word $v \in (N \cup \Sigma)^*$ according to context-free grammar $G = (N, \Sigma, S, R)$ if there exist words $u_1, u_2, \alpha \in (N \cup \Sigma)^*$ and a nonterminal $X \in N$ such that $u = u_1 X u_2$, $v = u_1 \alpha u_2$, and $X \to \alpha$ is a rule from R. We write $u \Rightarrow v$. The reflexive and transitive closure of the relation $\Rightarrow$ is denoted as $\Rightarrow^*$. Then the language generated by the grammar G is $L(G) = \{w \in \Sigma^* \mid S \Rightarrow^* w\}$.

Definition 1 (**CF(¢,$)-grammars, [6]**). *Let N and Σ be two disjoint alphabets, $¢, \$ \notin (N \cup \Sigma)$ and $G = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a context-free grammar generating a language of the form $\{¢\} \cdot L \cdot \{\$\}$, where $L \subseteq \Sigma^*$, and S does not occur in the right-hand side of any rule from R. We say that G is a CF(¢,$)-grammar. The language L is the* internal language of *G, and it is denoted as $L_{\text{in}}(G)$.*

Closure properties of the class of context-free languages imply that for a CF(¢,$)-grammar G, both languages $L(G)$ and $L_{\text{in}}(G)$ are context-free. The added right sentinel $ facilitates the recognition of languages. For example, if L is a deterministic context-free language, it can be generated by an LR(1)-grammar. But $L \cdot \{\$\}$ and $\{¢\} \cdot L \cdot \{\$\}$ are both generated by simpler LR(0) grammars [7]. The left sentinel ¢ is included in CF(¢,$)-grammars for compatibility with a version of restarting pumping automata from [8]. The class $\mathscr{L}_{in}(CF(¢, \$))$ of all internal languages of CF(¢,$)-grammars characterizes the class CFL.

2.1. Pumping infixes and reductions

Definition 2 ([6]). *Let $G = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a CF(¢,$)-grammar, $x, u_1, v, u_2, y \in \Sigma^*$, $u_1 u_2 \neq \lambda$, $A \in N$, and*

$$S \Rightarrow^* ¢xAy\$ \Rightarrow^* ¢xu_1Au_2y\$ \Rightarrow^* ¢xu_1vu_2y\$. \tag{1}$$

We say that $(¢x, u_1, A, v, u_2, y\$)$ is a pumping infix, *and $¢xu_1vu_2y\$ \rightsquigarrow_G ¢xvy\$$ is a* pumping reduction *by G.*

The infix and the reduction are two-side *if both u_1 and u_2 are nonempty. They are* right-side *(*left-side, *respectively) if u_1 (u_2, respectively) is empty.*

The relation $\rightsquigarrow_G^*$ is the reflexive and transitive closure of the pumping reduction relation $\rightsquigarrow_G$.

Note that we have not omitted the sentinels in the pumping infix and pumping reduction.

If $(¢x, u_1, A, v, u_2, y\$)$ is a pumping infix by G, then all words of the form $¢xu_1^i vu_2^i y\$$, for all integers $i \geq 0$, belong to $L(G)$.

Let $G = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a CF(¢,$)-grammar, t be the number of nonterminals of G, and k be the maximal length of the right-hand side of the rules from R, where the sentinels ¢, $ are not counted. Let T be a derivation tree according to G. If T has more than k^t leaves from Σ, a path exists from a leaf to the root of T such that it contains at least $t + 1$ nodes labeled with nonterminals. As G has only t nonterminals, at least two nodes on the path are labeled with the same nonterminal A. In that case, there is a pumping reduction, corresponding to this word. We say $K_G = k^t + 2$ is the *grammar number of* G.

Note that for each word from $L(G)$ of length greater than K_G, some pumping infix by G must correspond. On the other hand, each word generated by G that is not pumped is at most of length K_G. In the following, we will separate words that can be pumped from those that cannot.

Note that in the above derivation (1), the length of the words x, u_1, v, u_2, y is not limited.

A pumping reduction $w \rightsquigarrow_G w'$ corresponds to removing a segment between any nodes r_1 and r_2 labeled with the same nonterminal A occurring on a path from the root of a derivation tree for w.

The following obvious propositions were proved in [3].

Proposition 1 (Pumping reductions are correctness preserving, [3]). *Let $G = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a CF(¢,$)-grammar. Let G generate a word w_1, and $w_1, \dots, w_n$, for some integer $n \geq 1$, be a sequence of words such that $w_i \rightsquigarrow_G w_{i+1}$, for all $i = 1, \dots, n-1$, be a sequence of pumping reductions, and there is no $w_{n+1} \in \Sigma^*$ such that $w_n \rightsquigarrow_G w_{n+1}$.*

Then $w_i \in L(G)$, for all $i = 1, \dots, n$ and $|w_n| \leq K_G$.

Proposition 2 ([3]). *Let $G = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a CF(¢,$)-grammar, and G generates a word w_1 (that is, $w_1 \in L(G)$), and $|w_1| > K_G$. Then there is a sequence of words $w_1, \dots, w_n$, for some integer $n \geq 1$ such that, for all $i = 1, \dots, n-1$, $w_i \rightsquigarrow_G w_{i+1}$, $w_i \in L(G)$, for all $i = 1, \dots, n$, and $|w_n| \leq K_G$.*

2.2. Complete CF(¢,$)-grammars

In contrast to previous definitions (e.g., [3]), the following definition of complete CF(¢,$)-grammar requires that such a grammar to be reduced – it does not contain useless nonterminals (with a minor exception).

Definition 3. *Let $G_C = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a CF(¢,$)-grammar. Then G_C is called a* complete CF(¢,$)-grammar *if*

1. *$S \to S_A \mid S_R$, where $S_A, S_R \in N$, are the only rules in R containing the initial nonterminal S. No other rule of G_C contains S_A or S_R in its right-hand side.*
2. *The languages $L(G_A)$ and $L(G_R)$ generated by the grammars $G_A = (N, \Sigma \cup \{¢, \$\}, S_A, R)$ and $G_R = (N, \Sigma \cup \{¢, \$\}, S_R, R)$, respectively, are disjoint and complementary with respect to $\{¢\} \cdot \Sigma^* \cdot \{\$\}$. That is, $L(G_A) \cap L(G_R) = \emptyset$ and $L(G_C) = L(G_A) \cup L(G_R) = \{¢\} \cdot \Sigma^* \cdot \{\$\}$.*
3. *All nonterminals of G_C can be derived from S, and from all nonterminals of G_C (except for S_A and S_R) there are derivations of terminal words.*

We will denote the grammar as $G_C = (G_A, G_R)$. In addition, we will call G_A and G_R the acceptance and rejection grammar of the complete CF(¢,$)-grammar G_C, respectively.

The above definition implies that for each word of the form $¢w\$$, where $w \in \Sigma^*$, there is some derivation tree T according to G_C. The root of T has a single son labeled with one of the nonterminals S_A and S_R. If it is S_A, the word from the leaves of the tree T is from $L(G_A)$, otherwise it is from $L(G_R)$.

For any terminal word $w \in \{¢\} \cdot \Sigma^* \cdot \{\$\}$, there can exist several derivation trees. However, if $w \in L(G_A)$, all have S_A under their root. If $w \in L(G_R)$, they will have S_R under their root.

As $L(G_C) = \{¢\} \cdot \Sigma^* \cdot \{\$\}$ is an infinite language, there exist pumping reductions by G_C ([6]).

The condition that both acceptance and rejection grammar of a complete CF(¢,$)-grammar are context-free seems to be quite restrictive, but the class of deterministic context-free languages is closed under complement. Hence, if G_A is a CF(¢,$)-grammar which is LR(0), then there exists a complete CF(¢,$)-grammar $G_C = (G_A, G_R)$ which is LR (0) (see, e.g., [8]).

3. PS-Free pumping CF(¢,$)-grammars

In [6] it was shown that if an acceptance or rejection language of a complete CF(¢,$)-grammar $G_C = (G_A, G_R)$ enables pumping where two nonempty segments can be pumped and some other requirements are fulfilled, then the languages $L(G_A)$ and $L(G_R)$ are not regular.

In the following, we will use conditions similar to those in the well-known pumping lemma for regular languages.

Proposition 3 (See [9]). *Let A be a nondeterministic finite-state acceptor with k states. Suppose ζ is in $L(A)$ and $\zeta = \mu\xi\nu$, where $|\xi| \geq k$ and μ, ν are any (possibly empty) strings. It then follows that ξ can be written in the form $\xi = \alpha\beta\gamma$ where $1 \leq |\beta| \leq k$ and where $\mu\alpha\beta^i\gamma\nu$ is in $L(A)$, for all $i \geq 0$.*

Corollary 1. *Let L be a regular language, then there exists a constant $c > 0$ such that, for all words w_1, w_2 such that $w_1 w_2$ is in L, $|w_1| > c$, w_2 is any (possibly empty) word, there exists a word w_1' so that $|w_1'| < |w_1|$, and $w_1' w_2$ is in L.*

Proof: As $w_1 w_2 \in L$, we can apply Proposition 3, where $\mu = \lambda$, $\xi = w_1$, $\nu = w_2$, and c is the number of states of a finite-state automaton accepting the language L. Then we can take $w_1' = \alpha\gamma$ and the statement of Corollary 1 follows for $i = 0$. □

In other words, each long enough word from L can be shortened in its prefix of length at least c into a shorter word from L. As the class of regular languages is closed under reversal, an analogous statement also holds for suffixes of limited size. Nevertheless, it is known that Proposition 3 (and therefore also Corollary 1) is not a sufficient condition for the regularity of a language. Below, we use similar conditions for prefixes and suffixes together with further independence conditions of reductions in the prefixes and suffixes.

Here, we introduce a restricted version of complete CF(¢,$)-grammar that guarantees that both $L(G_A)$ and $L(G_R)$ are regular.

Definition 4. *Let $G_C = (G_A, G_R)$ be a complete CF(¢, $)-grammar and $c > 0$ be a constant such that*

(P) for each $xs \in L(G_C)$ such that $|x| > c$, there is x_1 such that $xs \rightsquigarrow_{G_C} x_1 s$, and
(S) for each $xs \in L(G_C)$ such that $|s| > c$, there is s_1 such that $xs \rightsquigarrow_{G_C} xs_1$.
(C) For all words x, s, x′ and s′ such that $xs \in L(G_C)$, $|x| > c$, $|s| > c$, $xs \rightsquigarrow_{G_C} x's \rightsquigarrow_{G_C} x's'$ implies $xs \rightsquigarrow_{G_C} xs' \rightsquigarrow_{G_C} x's'$.
(D) For each $xs', x's \in L(G_C)$ such that $|x| > c$, $|s| > c$, $xs' \rightsquigarrow_{G_C} x's'$ and $x's \rightsquigarrow_{G_C} x's'$ it holds $xs \rightsquigarrow_{G_C} x's$ and $xs \rightsquigarrow_{G_C} xs'$.

We say that G_C is a PS-free pumping *CF(¢, $)-grammar of width c.*

The additional conditions (C) and (D) in Definition 4 add "independence" of reductions in prefix and suffix. We will see that all these conditions together guarantee the regularity of the languages $L(G_A)$ and $L(G_R)$.
Recall that each pumping reduction by G_C on $xs \in L(G_A)$ is a pumping reduction by G_A. Similarly, each pumping reduction by G_C on $xs \in L(G_R)$ is a pumping reduction by G_R.

Definition 5 (Prefix-suffix reduction). *Let $G_C = (G_A, G_R)$ be a complete CF(¢, $)-grammar of width c.*

- *Let ρ be a pumping reduction of the form $xs \rightsquigarrow_{G_C} x_1 s$, and $|x| > c$. We say that ρ is a* prefix reduction (P-reduction) *by G_C of width c. The word x is the prefix being reduced, and s is the (fixed) suffix of ρ. Note that the length of the suffix s is not limited.*

- *Let ρ be a pumping reduction of the form $xs \rightsquigarrow_{G_C} xs_1$, and $|s| > c$. We say that ρ is a* suffix reduction (S-reduction) *by G_C of width c. The word s is the suffix being reduced, and x is the (fixed) prefix of ρ.*
- *Let ρ be a pumping reduction of the form $xs \rightsquigarrow_{G_C} x_1s_1$, and let ρ be either a P-reduction or an S-reduction. We say that ρ is a* PS-reduction *by G_C of width c. We write $xs \rightsquigarrow_{(G_C,PS)} x_1s_1$.*
  The relation $\rightsquigarrow^*_{(G_C,PS)}$ is the reflexive and transitive closure of the relation $\rightsquigarrow_{(G_C,PS)}$.

We also say that x, where $|x| > 0$, is a PS-prefix and that s, $|s| > 0$, is a PS-suffix.

Observation Let $G_C = (G_A, G_R)$ be a PS-free pumping CF(¢, \$)-grammar of width c. Then the following assertions hold:

(a1) For each $xs \in L(G_C)$ such that $xs \rightsquigarrow^*_{(G_C,PS)} x_1s_1$ we have the following $xs \in L(G_A) \Leftrightarrow x_1s_1 \in L(G_A)$. That is, all PS-reductions are error- and correctness-preserving.
(a2) For each $xs \in L(G_C)$ there is a reduction $xs \rightsquigarrow^*_{(G_C,PS)} x_1s_1$ such that $|x_1| \le c, |s_1| \le c$.

Condition (C) of Definition 4 of the free PS-pumping CF(¢, \$)-grammar can be extended for reductions of an arbitrary length.

Lemma 1. *Let $G_C = (G_A, G_R)$ be a PS-free pumping CF(¢, \$)-grammar of width c. Then, for all words x, s, x' and s' such that $xs \in L(G_C)$, $|x| > c$, $|s| > c$, $xs \rightsquigarrow^*_{G_C} x's \rightsquigarrow^*_{G_C} x's'$ implies $xs \rightsquigarrow^*_{G_C} xs' \rightsquigarrow^*_{G_C} x's'$.*

Proof: According to Definition 4, the statement holds for two-step pumping reductions. E.g., if in the reduction $xs \rightsquigarrow^*_{G_C} x's \rightsquigarrow^*_{G_C} x's'$, the first part $xs \rightsquigarrow^*_{G_C} x's$ requires two steps: $xs \rightsquigarrow_{G_C} x_1s \rightsquigarrow_{G_C} x's$ and the second part only one step $x's \rightsquigarrow_{G_C} x's'$, we have a three step reduction $xs \rightsquigarrow_{G_C} x_1s \rightsquigarrow_{G_C} x's \rightsquigarrow_{G_C} x's'$ and we can apply condition (C) to the last two steps. We get the sequence of reductions $xs \rightsquigarrow_{G_C} x_1s \rightsquigarrow_{G_C} x_1s' \rightsquigarrow_{G_C} x's'$. Next, we apply condition (C) to the first two reductions and get $xs \rightsquigarrow_{G_C} xs' \rightsquigarrow_{G_C} x_1s' \rightsquigarrow_{G_C} x's'$. Hence, $xs \rightsquigarrow^*_{G_C} xs' \rightsquigarrow^*_{G_C} xs'$.

Using similar reduction reordering, we can prove the statement of the lemma by induction on the number of steps in the series of reductions. □

Next, we introduce PS-prefixes and PS-suffixes with a limited size.

Basic PS-prefix and PS-suffix by G_C of width c. Let $G_C = (G_A, G_R)$ be a PS-free pumping CF(¢, \$)-grammar of width c, and xs be a word in $L(G_C)$.

- If $0 < |x| \le c$, we say that x is a *basic PS-prefix by G_C of width c.*
- If $0 < |s| \le c$, we say that s is a *basic PS-suffix by G_C of width c.*

The set of all PS-prefixes for G_C will be denoted as

$$Pref(G_C) = \{x \mid x \neq \lambda, \exists s \in (\Sigma \cup \{¢, \$\})^* : xs \in \{¢\} \cdot \Sigma^* \cdot \{\$\}\}.$$

The set of all basic PS-prefixes will be denoted as

$$BPref(G_C, c) = \{\{¢\} \cdot w \mid w \in \Sigma^*, \text{ and } |w| < c\}.$$

Similarly, the set of all basic PS-suffixes will be denoted as

$$BSuff(G_C, c) = \{w \cdot \{\$\} \mid w \in \Sigma^*, \text{ and } |w| < c\}.$$

In what follows, for a set X, $\mathscr{P}(X)$ denotes the set of all its subsets.

Definition 6 (Prefix characteristic function). *Let $G_C = (G_A, G_R)$ be a PS-free pumping CF(¢, \$)-grammar of width $c > 0$.*

A characteristic function of a PS-prefix x is the function

$$Ch(c) : Pref(G_C) \times BSuff(G_C, c) \rightarrow \mathscr{P}(BPref(G_C, c))$$

that for each PS-prefix x and a basic suffix s_b assigns the set of all basic PS-prefixes to which the prefix x can be reduced, when we reduce the word xs_b. Formally:

$$Ch(c)(x, s_b) = \{x_b \in BPref(G_C, c) \mid xs_b \rightsquigarrow^*_{(G_C,PS)} x_bs_b\}.$$

As the sets $BPref(G_C, c)$ and $BSuff(G_C, c)$ are finite, for a fixed c and x, there exist only finitely many different functions $Ch(c)(x, \cdot)$.

For a PS-prefix x and $s_b \in BSuff(G_C, c)$, the result $Ch(c)(x, s_b)$ is always a nonempty set, as either

- $|x| \leq c$ and $x \in Ch(c)(x, s_b)$, or
- $|x| > c$ and, according to Observation (a2) above, there exists a basic PS-prefix x_b such that $xs_b \rightsquigarrow^*_{(G_C,PS)} x_b s_b$ and $x_b \in Ch(c)(x, s_b)$.

3.1. Equivalence of PS-prefixes and regularity of languages generated by PS-free pumping grammars

Based on the characteristic functions of PS-prefixes, we will define an equivalence on PS-prefixes.

Definition 7. *Let $G_C = (G_A, G_R)$ be a PS-free CF(¢,\$)-grammar of width $c > 0$. We say that two PS-prefixes x and z are equivalent, we write $x \cong z$, if for all $s_b \in BSuff(G_C, c)$ it holds $Ch(c)(x, s_b) = Ch(c)(z, s_b)$.*

Lemma 2. *Let $G_C = (G_A, G_R)$ be a PS-free CF(¢,\$)-grammar of width $c > 0$. The relation $\cong$ on the set of PS-prefixes by G_C is an equivalence relation of finite index.*

Proof: It is easy to see that the relation $\cong$ is reflexive, symmetric, and transitive, as the equality relation $=$ has all these properties.

The number of different characteristic functions is finite. Therefore, the number of equivalence classes of $\cong$ is finite. □

The following lemma is the key property for showing that the acceptance and rejection languages of a PS-free pumping grammar are regular.

Lemma 3. *Let $G_C = (G_A, G_R)$ be a PS-free CF(¢,\$)-grammar of width $c > 0$, and let $x \cong y$ for some PS-prefixes x, y by G_C. Then, for each PS-suffix s, $xs \in L(G_A) \Leftrightarrow ys \in L(G_A)$.*

Proof: Let x and y be two PS-prefixes by G_C such that $x \cong y$, and let s be a PS-suffix by G_C. Let us suppose $xs \in L(G_A)$. This means that $xs \rightsquigarrow^*_{(G_C,PS)} x_b s \rightsquigarrow^*_{(G_C,PS)} x_b s_b$, for some basic PS-prefix $x_b \in BPref(G_C, c)$ and a basic PS-suffix $s_b \in BSuff(G_C, c)$.

Condition (C) of Definition 4 implies that $xs_b \rightsquigarrow^*_{(G_C,PS)} x_b s_b$. Then, according to Definition 7, $x_b \in Ch(c)(x, s_b) = Ch(c)(y, s_b)$. This means that $ys_b \rightsquigarrow^*_{(G_C,PS)} x_b s_b$. Using condition (D) from the definition of PS-free pumping grammar (Definition 4), we get $ys \rightsquigarrow^*_{(G_C,PS)} x_b s$. The final step follows from the fact that $x_b s \rightsquigarrow^*_{(G_C,PS)} x_b s_b$ and $ys_b \rightsquigarrow^*_{(G_C,PS)} x_b s_b$. The PS-reduction $\rightsquigarrow^*_{(G_C,PS)}$ is error- and correctness-preserving, all words $x_b s$, $x_b s_b$, ys_b, and ys are in $L(G_A)$.

The corresponding statement holds for $L(G_R)$, which completes the proof that $xs \in L(G_A) \Leftrightarrow ys \in L(G_A)$. □

Let R_L denote the equivalence with respect to a language L defined in the following way: xR_Ly if and only if for all words z, $xz \in L \Leftrightarrow yz \in L$. According to Theorem 3.1 from [5], the language L is regular if and only if R_L has a finite index.

Lemma 4. *Let $G_C = (G_A, G_R)$ be a PS-free pumping CF(¢,\$)-grammar of width $c > 0$ and let R_L denote the equivalence with respect to a language $L = L(G_A)$ defined in the following way: xR_Ly if and only if for each word z, it holds that $xz \in L \Leftrightarrow yz \in L$. The relation R_L has a finite index.*

Proof: Let us consider the relation R_L where $L = L(G_A)$. As $L(G_A) \subseteq \{¢\} \cdot \Sigma^* \cdot \{\$\}$, all words that are not prefixes of words from $\{¢\} \cdot \Sigma^* \cdot \{\$\}$ are R_L-equivalent.

If $\hat{L}$ is an arbitrary language, all words in $\hat{L}$ do not need to be equivalent to $R_{\hat{L}}$. For example, there can exist two words $x, y \in \hat{L}$ such that $xs \in \hat{L}$ and $ys \notin \hat{L}$. However, all words from $L = L(G_A) \subseteq \{¢\} \cdot \Sigma^* \cdot \{\$\}$ end with a single symbol \$. All words in $L = L(G_A)$ are equivalent and there are no words outside L that can be equivalent to a word in L, because the only word that we can append to a word from $L(G_A)$ to obtain a word from $L(G_A)$ is the empty word λ.

Further, we will show that all *proper* prefixes of words from $\{¢\} \cdot \Sigma^* \cdot \{\$\}$ belong to a finite number of different equivalence classes with respect to R_L.

For a contradiction, assume that the number of equivalence classes with respect to R_L is infinite. Then, for each $n \geq 1$, there exist words $x_1, x_2, \ldots, x_n$ that are pairwise not equivalent.

If two words are proper prefixes of words from $\{¢\} \cdot \Sigma^* \cdot \{\$\}$, then these words are PS-prefixes. If n is greater than the number of equivalence classes with respect to $\cong$, then there exist two words x_i, x_j, $x_i \neq x_j$, such that $x_i \cong x_j$. However, according to Lemma 3, the words x_i and x_j are R_L equivalent – a contradiction to the assumption that R_L has an infinite index. □

Corollary 2. *Let $G_C = (G_A, G_R)$ be a PS-free CF(¢,\$)-grammar of width $c > 0$. Then the languages $L(G_A)$ and $L(G_R)$ are regular.*

Proof: Lemma 4 implies that the relation R_L for $L = L(G_A)$ has a finite index, and according to Theorem 3.1 from [5], the language $L(G_A)$ is regular. As $L(G_R) = \{¢\} \cdot \Sigma^* \cdot \{\$\} \setminus L(G_A)$, the regularity of the language $L(G_R)$ follows from the closure properties of regular languages. □

Corollary 3. *Let $G_C = (G_A, G_R)$ be a complete CF(¢, \$)-grammar, and $L(G_A)$ be a non-regular language. Then G_C is not a PS-free pumping CF(¢, \$)-grammar.*

Lemma 5. *For each regular language $L_r \subseteq \Sigma^*$, where Σ does not contain sentinels ¢, \$ there is a CF(¢,\$)-grammar $G_{C,r} = (G_{A,r}, G_{R,r})$ such that $\{¢\} \cdot L_r \cdot \{\$\} = L(G_{A,r})$, and $G_{C,r}$ is a complete PS-free pumping CF(¢,\$)-grammar.*

Proof: A context-free grammar is a right-linear grammar if each of its rules has at most one nonterminal symbol; the nonterminal appears on the right end of the rule. It is well known that for each regular language L there is a right-linear (that is, regular) grammar G in the Chomsky normal form such that $L(G) = L$. Additionally, each rule of grammar G will have at most two symbols on the right-hand side. During each derivation according to G, the sentential form contains at most one nonterminal, and a nonterminal must be repeated in any sequence of derivation steps longer than the number of nonterminals of G.

Let L_r be a regular language. Then, its complement $\bar{L}_r$ is also a regular language. Let G_A and G_R be right-linear grammars in Chomsky normal form such that $L(G_A) = L_r$ and $L(G_R) = \bar{L}_r$.

It is easy to transform both the grammars G_A and G_R into the right-linear grammars $G_{A,r}$ and $G_{R,r}$ such that $L(G_{A,r}) = \{¢\} \cdot L(G_A) \cdot \{\$\}$ and $L(G_{R,r}) = \{¢\} \cdot L(G_R) \cdot \{\$\}$, respectively.

Let c be the number of nonterminals of $G_{C,r}$. It is easy to see that $G_{C,r} = (G_{A,r}, G_{R,r})$ is a complete PS-free pumping CF(¢,\$)-grammar of width $c + 1$. □

The next theorem says that the internal languages of acceptance languages of complete PS-free pumping CF(¢, \$)-grammars characterize the class of regular languages.

Theorem 1. *A language L is a regular language if and only if there exists a complete PS-free pumping CF(¢,\$)-grammar $G_C = (G_A, G_R)$, such that $L = L_{in}(G_A)$.*

Proof: The theorem is a consequence of Corollary 2 and Lemma 5.

□

4. Comparing PS-free and one-side pumping CF(¢,\$)-grammars

This section relates PS-free pumping grammars with one-side pumping complete CF(¢,\$)-grammars studied in [3] (for their definition see below). Using sample complete CF(¢,\$)-grammars, we show that the notions of one-side or PS-free pumping represent different restrictions of complete CF(¢,\$)-grammars that both restrict them to generate only regular languages.

Let (x, u_1, A, v, u_2, y) be a pumping infix by a CF(¢,$)-grammar G. The pumping infix is a *core pumping infix* if there is a derivation tree T by G that corresponds to the derivation

$$S \Rightarrow^* ¢xAy\$ \Rightarrow^* ¢xu_1Au_2y\$ \Rightarrow^* ¢xu_1vu_2y\$ \quad (2)$$

such that the path between the root r_1 of the subtree corresponding to the derivation of u_1Au_2 from A in (2) to the root r_2 of the subtree corresponding to the derivation of v (but without r_2) does not contain two distinct nodes labeled with the same nonterminal.

One-side/Two-side pumping CF(¢,$)-grammars. Let G be a CF(¢,$)-grammar. We say that G is a *left-side pumping CF(¢,$)-grammar* if all its core pumping infixes are left-side pumping infixes (Definition 2). Similarly, we say that G is a *right-side pumping CF(¢,$)-grammar* if all its core pumping infixes are right-side pumping infixes. We say that G is a *one-side pumping CF(¢,$)-grammar* if it is either left-side or right-side pumping CF(¢,$)-grammar. Finally, we say that G is a *two-side pumping CF(¢,$)-grammar* if any of its core pumping infixes is a two-side pumping infix.

The paper [3] showed that one-side pumping complete CF(¢,$)-grammars characterize the class of regular languages.

Proposition 4 ([3]). *Let $G_C = (G_A, G_R)$ be a complete one-side pumping CF(¢,$)-grammar. Then, both $L(G_A)$ and $L_{\text{in}}(G_A)$ are regular languages.*

On the other hand, the next example shows that a PS-free pumping CF(¢,$)-grammar need not be one-side pumping.

Example 1. *Let $G_C^{(1)} = (\{S, S_A, S_R, B, C\}, \{a, b, ¢, \$\}, S, R)$ be a complete CF(¢,$)-grammar with the following set of rules:*

$$\begin{array}{lcl} S & \rightarrow & S_A \mid S_R, \\ S_A & \rightarrow & ¢A\$ \mid ¢B\$ \mid ¢\$, \\ A & \rightarrow & aAb \mid ab, \\ B & \rightarrow & aB \mid bB \mid a \mid b. \end{array}$$

The grammar $G_C^{(1)}$ is a complete CF(¢,$)-grammar $G_C^{(1)} = (G_A^{(1)}, G_R^{(1)})$, where $G_A^{(1)}$ has starting symbol S_A, $G_R^{(1)}$ has starting symbol S_R, and obviously, $L(G_C^{(1)}) = L(G_A^{(1)}) = \{¢\} \cdot \{a, b\}^* \cdot \{\$\}$, and $L(G_R) = \emptyset$.

As $(¢aa, a, A, ab, b, bb\$)$ is a two-side core pumping infix by $G_A^{(1)}$, the word $w = ¢aaaabbbb\$$ is in $L(G_A^{(1)})$. However, the word w can also be derived using rules of $G_A^{(1)}$ that do not contain the nonterminal A, and all corresponding pumping infixes are one-side pumping infixes.

Hence, $G_C^{(1)} = (G_A^{(1)}, G_R^{(1)})$ is a PS-free pumping CF(¢,$)-grammar of width 10 that is not a one-side pumping grammar.

Actually, we could omit all the rules that include A from the grammar, and we would obtain a PS-free pumping grammar that generates the same language as the original grammar and is left-side pumping.

The next corollary is a consequence of Example 1.

Corollary 4. *There is a PS-free pumping complete CF(¢, \$)-grammar whose acceptance language is regular and which is not a one-side pumping CF(¢,$)-grammar.*

The next sample complete CF(¢,$)-grammar generates a regular language, but it is not PS-free pumping grammar.

Example 2. *Let us consider a complete CF(¢,$)-grammar $G_C^{(2)} = (G_A^{(2)}, G_R^{(2)})$ generating the acceptance language $\{¢a^nb^m\$ \mid n + m > 0\}$.*

Let $G_C^{(2)}$ use the following rules:

$$\begin{array}{lcl} S & \to & S_A \mid S_R \\ S_A & \to & ¢A\$ \mid ¢AD\$ \mid ¢B\$ \mid ¢DB\$ \mid ¢D\$ \\ A & \to & aA \mid a \\ B & \to & bB \mid b \\ D & \to & aDb \mid ab \\ S_R & \to & ¢C\$ \\ C & \to & AC \mid CB \mid CA \mid BC \mid ba \end{array}$$

The complete CF(¢,\$)-grammar $G_C^{(2)}$ is not a PS-free pumping CF(¢,\$)-grammar, as strings of the form ¢$a^n b^n$\$, for $n > 0$, do not allow pumping reductions in the suffix of b's only by $G_C^{(2)}$. All derivations of $G_A^{(2)}$ that use nonterminal B derive words with more b-s than a-s. A similar assertion holds for a's. Hence, all words of the form ¢$a^n b^n$\$ are generated using nonterminal D, which induces a two-side pumping infixes and the condition (P) of Definition 4 is not satisfied. This means that $G_C^{(2)}$ is not a PS-free pumping CF(¢,\$)-grammar. The grammar is not one-side pumping grammar either, as, e.g., $(¢, a, D, ab, b, \$)$ is a two-side pumping infix by $G_C^{(2)}$.

The next corollary follows from the previous example.

Corollary 5. *There is a complete CF(¢, \$)-grammar $G_C^{(2)} = (G_A^{(2)}, G_R^{(2)})$ which is not a PS-free pumping CF(¢,\$) grammar such that both $L(G_A^{(2)})$ and $L(G_R^{(2)})$ are regular languages.*

The previous example motivates further work. We should study more deeply the conditions for the separation of regular and non-regular complete CF(¢,\$)-grammars.

5. Conclusion and future work

In this paper, we were looking for possibly maximally relaxed constraints based on pumping reductions for a complete CF(¢, \$)-grammar, which ensure that the grammar generates regular acceptance and rejection languages. Our approach can be seen as extending the pumping lemma for regular languages.

We have succeeded in the sense that PS-free pumping CF(¢,\$)-grammars generate regular languages only. However, the conditions for PS-free pumping grammars are sufficient but not necessary to limit the generated languages to the class of regular languages. The obvious open problem is finding conditions necessary and sufficient to limit the power of complete CF(¢,\$)-grammar to regular languages.

The way to achieve that could start with comparing constraints for non-regularity (from [6]) and for regularity (from this paper) of complete CF(¢, \$)-grammars in a uniform way.

Until now, we have not studied the computability and decidability questions connected with complete CF(¢, \$)-grammars. New insight could be obtained by comparing the differences between the grammar complexity of complete CF(¢, \$)-grammars and the complexity of their languages.

Acknowledgments

Marta Vomlelová thanks for support by the institutional support RVO:m 67985807, and by GAČR project 25-18003S. František Mráz and Martin Plátek thank for support by the institutional support RVO:m 67985807.

Declaration on generative AI

The authors have not employed any generative AI tools.

References

[1] M. Lopatková, M. Plátek, P. Sgall, Towards a formal model for functional generative description: Analysis by reduction and restarting automata, Prague Bull. Math. Linguistics 87 (2007) 7–26. URL: http://ufal.mff.cuni.cz/pbml/87/lopatkova-et-al.pdf.

[2] F. Otto, Restarting Automata: The Standard Type of Restarting Automaton and Its Variants, Springer Nature Switzerland, Cham, 2024. URL: https://doi.org/10.1007/978-3-031-70094-1. doi:10.1007/978-3-031-70094-1.

[3] F. Mráz, M. Plátek, D. Pardubská, D. Průša, One-side pumping and two-side pumping by complete CF(¢, $)-grammars, in: B. Brejová, L. Ciencialová, M. Holeňa, R. Jajcay, T. Jajcayová, M. Lexa, F. Mráz, D. Pardubská, M. Plátek (Eds.), Proceedings of the 23rd Conference Information Technologies - Applications and Theory (ITAT 2023), volume 3498 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 110–120. URL: https://ceur-ws.org/Vol-3498/paper14.pdf.

[4] F. Otto, Restarting automata, in: Z. Ésik, C. Martín-Vide, V. Mitrana (Eds.), Recent Advances in Formal Languages and Applications, volume 25 of *Studies in Computational Intelligence*, Springer, 2006, pp. 269–303. URL: https://doi.org/10.1007/978-3-540-33461-3_11.

[5] J. E. Hopcroft, J. D. Ullman, Formal languages and their relation to automata, Addison-Wesley series in computer science and information processing, Addison-Wesley, 1969. URL: https://www.worldcat.org/oclc/00005012.

[6] M. Plátek, F. Mráz, D. Pardubská, M. Procházka, Non-regularity of complete CF(¢, $)-grammars, in: L. Ciencialová, M. Holeňa, R. Jajcay, T. Jajcayová, F. Mráz, D. Pardubská, M. Plátek (Eds.), Proceedings of the 24th Conference Information Technologies – Applications and Theory (ITAT 2024), volume 3792 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 75–80. URL: https://ceur-ws.org/Vol-3792/paper8.pdf.

[7] M. A. Harrison, Introduction to Formal Language Theory, Addison-Wesley, USA, 1978.

[8] M. Plátek, F. Mráz, D. Pardubská, D. Průša, On pumping RP-automata controlled by complete LRG(¢, $)-grammars, in: L. Ciencialová, M. Holeňa, R. Jajcay, T. Jajcayová, F. Mráz, D. Pardubská, M. Plátek (Eds.), Proceedings of the 22nd Conference Information Technologies – Applications and Theory (ITAT 2022), volume 3226 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 111–121. URL: https://ceur-ws.org/Vol-3226/paper13.pdf.

[9] W. J. Savitch, Abstract machines and grammars, Little, Brown and Company, 1982.