Non-Regularity of Complete CF(¢,$)-Grammars

Martin Plátek[1], František Mráz[1,*], Dana Pardubská[2] and Martin Procházka[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
We study pumping analysis by reduction represented by complete CF(¢,$)-grammars and their languages. A complete CF(¢,$)-grammar generates both a language and its complement. Complete CF(¢,$)-grammars serve as a tool to study the class of context-free languages that are closed under complement. Recall that the class of context-free grammars is the single class of languages from the Chomsky hierarchy that is not closed under the complement.

The pumping reductions used in this paper ensure a correctness- and error-preserving pumping analysis by reduction for each word over its input alphabet. We introduce tests for each pumping reduction, which serve as tests of non-regularity for accepted and rejected languages by corresponding grammars. That can help to develop natural error localization and error recovery techniques for languages defined by complete CF(¢,$)-grammars.

Keywords
complete CF(¢,$)-grammar, pure pumping infix, pumping test, non-regularity

1. Introduction

This paper is a continuation of the papers [1, 2, 3, 4], inspired also by [5, 6]. We introduce and study complete context-free grammars with sentinels, mainly their pumping reductions and pumping tests. These notions are motivated by the linguistic method called analysis by reduction (here mentioned as reduction analysis); see [7, 8, 9, 10].

Reduction analysis is a method for checking the correctness of an input word by stepwise rewriting some part of the current form with a shorter one until we obtain a simple word for which we can decide its correctness easily. In general, reduction analysis is nondeterministic, and in one step, we can rewrite a substring of a length limited by a constant with a shorter string. An input word is accepted if there is a sequence of reductions such that the final simple word is from the language. Then, intermediate words obtained during the analysis are also accepted. Each reduction must be *error-preserving*; that is, no word outside the target language can be rewritten into a word from the language.

This paper focuses mainly on a restricted version of the reduction analysis called *pumping reduction analysis*, which has several additional properties. In each step of the pumping reduction analysis, the current word is not rewritten. Instead, at most two continuous segments of the current word are deleted. In addition, we consider the pumping reduction analysis for languages generated by the so-called complete grammars.

Informally, a complete grammar (with sentinels ¢ and $) G_C is an extended context-free grammar (CFG) with two initial nonterminals S_A and S_R. Such grammar has a finite alphabet Σ of terminals not containing ¢ and $, a finite alphabet of nonterminals, and a set of rewriting rules of the form $X \to \alpha$, where X is a nonterminal and α is a string of terminals, nonterminals, and sentinels ¢, $. The language generated by the grammar is the set $\{¢\} \cdot \Sigma^* \cdot \{\$\}$. The set of words generated from the initial nonterminal S_A called the accepting language, is a language of the form $\{¢\} \cdot L \cdot \{\$\}$, where $L \subseteq \Sigma^*$, and the set of words generated from the second initial nonterminal S_R, called rejecting language, is exactly $\{¢\} \cdot (\Sigma^* \setminus L) \cdot \{\$\}$.

Pumping reduction analysis corresponds to a complete grammar G_C when for each pair of terminal words u, v such that u can be reduced to v, it holds that there are some terminal words x_1, x_2, x_3, x_4, x_5, and a nonterminal A satisfying $u = x_1x_2x_3x_4x_5$, $v = x_1x_3x_5$, and $S \Rightarrow^*_{G_C} x_1Ax_5 \Rightarrow^*_{G_C} x_1x_2Ax_4x_5 \Rightarrow^*_{G_C} x_1x_2x_3x_4x_5$, where S equals S_A or S_R. Additionally, there exists a constant c that depends only on grammar G_C, such that each word of length at least c can be reduced to a shorter word.

In general, it is undecidable whether an arbitrary context-free grammar generates a regular language [11]. This means that no algorithm can universally determine if a given CFG produces a regular language. We propose to use some tests to test the non-regularity of accepting and rejecting languages. The main result of the paper says that if a complete grammar (with sentinels ¢ and $)

ITAT'24: Information Technologies – Applications and Theory, September 20–24, 2024, Drienica, Slovakia
*Corresponding author.
martin.platek@mff.cuni.cz (M. Plátek);
frantisek.mraz@mff.cuni.cz (F. Mráz);
pardubska@dcs.fmph.uniba.sk (D. Pardubská);
martproc@gmail.com (M. Procházka)
0000-0003-3147-6442 (M. Plátek); 0000-0001-3869-3340 (F. Mráz);
0000-0001-9383-8117 (D. Pardubská)

G_C has a switching pumping test, then its accepting and rejecting languages are non-regular.

This paper is structured as follows. Section 2 introduces CF(¢,\$)-grammars, pumping infixes and reductions, and complete CF(¢,\$)-grammars. Section 3 presents the main result. It is followed by a section that discusses open problems and future work.

2. Basic notions

Definition 1 (CF(¢,\$)-grammars). *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 in R. We say that G is a CF(¢,\$)-grammar. The language L is the* internal language of G*, and is denoted $L_{\text{in}}(G)$.*

The 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. E.g., if L is a deterministic context-free language, then it can be generated by an LR(1)-grammar. But then, $L \cdot \{\$\}$ and $\{¢\} \cdot L \cdot \{\$\}$ are both generated by simpler LR(0) grammars. The left sentinel ¢ is included in CF(¢,\$)-grammars for compatibility with RP-automata. The class of all $L_{\text{in}}(G)$ characterizes the class CFL.

2.1. Pumping infixes and reductions

Definition 2. *Let $G = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a CF(¢,\$)-grammar, x, u_1, v, u_2, y be words over Σ, $|u_1 u_2| > 0$, and $A \in N$ be a nonterminal. If*

$$S \Rightarrow^* ¢xAy\$ \Rightarrow^* ¢xu_1Au_2y\$ \Rightarrow^* ¢xu_1vu_2y\$ \quad (1)$$

we say that $(¢x, u_1, A, v, u_2, y\$)$ is a pumping infix *by G, and that $¢xu_1vu_2y\$ \rightsquigarrow_{P(G)} ¢xvy\$$ is a* pumping reduction *by G.*

If both u_1 and u_2 are not empty, we say that $(¢x, u_1, A, v, u_2, y\$)$ is a two-side pumping infix *by G, and that $¢xu_1vu_2y\$ \rightsquigarrow_{P(G)} ¢xvy\$$ is a* two-side pumping reduction *by G.*

If $u_1 = \lambda$ we say that $(¢x, u_1, A, v, u_2, y\$)$ is a right-side pumping infix *by G, and $¢xvu_2y\$ \rightsquigarrow_{P(G)} ¢xvy\$$ is a* right-side pumping reduction *by G.*

If $u_2 = \lambda$ we say that $(¢x, u_1, A, v, u_2, y\$)$ is a left-side pumping infix *by G, and $¢xu_1vy\$ \rightsquigarrow_{P(G)} ¢xvy\$$ is a* left-side pumping reduction *by G.*

The relation $\rightsquigarrow^*_{P(G)}$ is the reflexive and transitive closure of the pumping reduction relation $\rightsquigarrow_{P(G)}$.

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 $¢xu_1^ivu_2^iy\$$, 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 in R. Let T be a derivation tree according to G. If T has more than k^t leaves, a path exists from a leaf to the root of T such that it contains at least $t + 1$ nodes labeled by 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 that $K_G = k^t$ 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 of length at most K_G.

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 $w \rightsquigarrow_{P(G)} w'$ corresponds to removing a part of the derivation tree between some two nodes r_1, r_2 labeled with the same nonterminal A occurring on a path from the root of the derivation tree for w.

2.2. Complete CF(¢,\$)-grammars

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 $L(G_C) = L(G_A) \cup L(G_R) = \{¢\} \cdot \Sigma^* \cdot \{\$\}$.*

We will denote the grammar as $G_C = (G_A, G_R)$. Further, we will call G_A and G_R as accepting and rejecting grammar of the complete CF(¢,\$)-grammar G_C, respectively.

For each word of the form $¢w\$$, where $w \in \Sigma^*$, there is some derivation tree T according to G_C. The node under the root of T is labeled either S_A or S_R. If it is S_A, the word is generated by the accepting grammar G_A. Otherwise, it is generated by the rejecting grammar G_R.

Moreover, for each word, two or more derivation trees can exist, but all of them are accepting or all of them are rejecting.

3. Non-regularity by complete CF(¢,$)-grammars

If G_C is a complete CF(¢,$)-grammar, then both $L(G_A)$ and $L(G_R)$ are context-free languages. How can we decide whether those languages are regular or non-regular? In this section, we show some properties that help answer that question.

At first, we introduce a weaker notion of pumping infix that does not contain the information on which nonterminal is pumped.

Definition 4 (Pure pumping infix/reduction). *Let $G_C = (G_A, G_R) = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a complete CF(¢, \$)-grammar, x, u_1, v, u_2, y be some words, $x \in \{¢\} \cdot \Sigma^*$, $u_1, u_2 \in \Sigma^*$, $|u_1u_2| > 0$, $y \in \Sigma^* \cdot \{\$\}$.*

- *If $xu_1^n vu_2^n y$ is in $L(G_A)$, for each integer $n \geq 0$, we say that (x, u_1, v, u_2, y) is a* pure pumping infix *by G_A. We say that the pair of words xu_1vu_2y, xvy is* pure pumping reduction *by G_A and write $xu_1vu_2y \equiv\!>_{G_A} xvy$.*

- *If $xu_1^n vu_2^n y$ is in $L(G_R)$, for each integer $n \geq 0$, we say that (x, u_1, v, u_2, y) is a* pure pumping infix *by G_R. We say that the pair of words xu_1vu_2y, xvy is* pure pumping reduction *by G_R. We write $xu_1vu_2y \equiv\!>_{G_R} xvy$.*

We say that (x, u_1, v, u_2, y) is a pure pumping infix *by G_C if it is a pure pumping infix by G_A or by G_R. We say that the pair of words xu_1vu_2y, xvy is a* pure pumping reduction *by G_C if it is a pumping reduction by G_A or by G_R. We write $xu_1vu_2y \equiv\!>_{G_C} xvy$.*

Actually, pure pumping infix need not directly correspond to any pumping infix by the given complete CF(¢,$)-grammar. This is illustrated with the following example.

Example 1. *Let $G_C = (G_A, G_R)$, $G_C = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a complete CF(¢,\$)-grammar, where $N = \{S, S_A, S_R, A, B, C, D, E\}$, $\Sigma = \{a, b\}$, S_A and S_R are the initial nonterminals of the grammars G_A and G_R, respectively, and R consists of the following rules:*

$$\begin{array}{lll}
S & \to & S_A \mid S_R, \\
S_A & \to & ¢\$ \mid ¢A\$ \mid ¢AC\$ \mid ¢C\$, \\
A & \to & aA \mid a, \\
C & \to & aDb \mid ab, \\
D & \to & aCb \mid ab, \\
S_R & \to & ¢B\$ \mid ¢CB\$ \mid ¢ba\$ \mid ¢Eab\$ \mid \\
 & & ¢abE\$ \mid ¢Eab\$, \\
B & \to & bB \mid b, \\
E & \to & aE \mid bE \mid a \mid b.
\end{array}$$

Clearly, grammar G_A generates the language $L(G_A) = \{¢\} \cdot L_A \cdot \$$, where $L_A = \{a^n b^m \mid n \geq m \geq 0\}$ and grammar G_R generates the language $L(G_R) = \{¢\} \cdot L_R \cdot \$$, where $L_R = \{a, b\}^* \setminus L_A$.

As we have the following derivation according to G_C

$$\begin{array}{l}
S \Rightarrow_{G_C} S_A \Rightarrow_{G_C} ¢C\$ \Rightarrow_{G_C} ¢aDb\$ \Rightarrow_{G_C} \\
\quad ¢aaCbb\$ \Rightarrow_{G_C} ¢aaaDbbb\$ \Rightarrow_{G_C} \\
\quad ¢aaaaCbbbb\$ \Rightarrow_{G_C} ¢aaaaabbbbb\$,
\end{array}$$

the pumping infix $(¢aa, aa, C, ab, bb, bb\$)$ is a pumping infix by G_C and by G_A. On the other hand, $(¢, a, ab, b, \$)$ is a pure pumping infix by G_C and by G_A such that there does not exist any pumping infix by G_C of the form $(¢, a, X, ab, b, \$)$, where X is a nonterminal of grammar G_C.

Theorem 1. *Let $G_C = (G_A, G_R)$, $G_C = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a complete CF(¢,\$)-grammar, and at least one of the following conditions is fulfilled (for some words $x \in \{¢\} \cdot \Sigma^*$, $u_1, v, u_2 \in \Sigma^*$, and $y \in \Sigma^* \cdot \{\$\}$):*

(ARl) *The words u_1 and u_2 are nonempty, (x, u_1, v, u_2, y) is a pure pumping infix by G_A, and there are integers $i \geq 0, j > 0$ such that*

$$xu_1^{j \cdot m} u_1^{i+j \cdot n} vu_2^{i+j \cdot n} y \in L(G_R),$$

for each $m > 0, n \geq 0$.

(ARr) *There exists $w = xu_1vu_2y \in L(G_A)$ such that u_1 and u_2 are nonempty, (x, u_1, v, u_2, y) is a pure pumping infix by G_A, and there are integers $i \geq 0$, $j > 0$ such that*

$$xu_1^{i+j \cdot n} vu_2^{i+j \cdot n} u_2^{j \cdot m} y \in L(G_R),$$

for each $m > 0, n \geq 0$.

(RAl) *There exists $w = xu_1vu_2y \in L(G_R)$ such that u_1 and u_2 are nonempty, (x, u_1, v, u_2, y) is a pure pumping infix by G_R, and there are integers $i \geq 0$, $j > 0$ such that*

$$xu_1^{j \cdot m} u_1^{i+j \cdot n} vu_2^{i+j \cdot n} y \in L(G_A),$$

for each $m > 0, n \geq 0$.

(RAr) *There exists $w = xu_1vu_2y \in L(G_R)$ such that u_1 and u_2 are nonempty, (x, u_1, v, u_2, y) is a pure pumping infix by G_R, and there are integers $i \geq 0$, $j > 0$ such that*

$$xu_1^{i+j \cdot n} vu_2^{i+j \cdot n} u_2^{j \cdot m} y \in L(G_A),$$

for each $m > 0, n \geq 0$.

Then $L(G_A)$ and $L(G_R)$ are non-regular languages.

Proof: We prove the case (ARl), whose name comes from Accept-Reject-left with the meaning that the words of the form $x, u_1^r v u_2^r y$ are generated by the accepting grammar G_A and the words of the form $x u_1^{j\cdot m} u_1^{i+j\cdot n} v u_2^{i+j\cdot n} y$ are generated by the rejecting grammar G_R, and they contain more copies of u_1 on the left from v than the number of copies of u_2 to the right from v. Then, the cases (ARr) (Accept-Reject-right), (RAl) (Reject-Accept-left), and (RAr) (Reject-Accept-right) can be shown analogously.

Let $w = x u_1 v u_2 y \in L(G_A)$, where u_1 and u_2 are non-empty, (x, u_1, v, u_2, y) be a pure pumping infix by G_A, and $i \geq 0, j > 0$ be integers such that

$$x u_1^{j\cdot m} u_1^{i+j\cdot n} v u_2^{i+j\cdot n} y \in L(G_R),$$

for each $m > 0, n \geq 0$.

Assume for a contradiction that $L(G_A)$ and $L(G_R)$ are regular languages. According to Myhill-Nerode Theorem [12], a right congruence $\equiv$ with a finite index r exists such that language $L(G_A)$ is a union of some of its equivalence classes.

Consider the set of words $\left\{x u_1^{i+j\cdot 1}, x u_1^{i+j\cdot 2}, \ldots, x u_1^{i+j\cdot(r+1)}\right\}$. Obviously, there are $1 \leq k_1 < k_2 \leq r+1$ such that $x u_1^{i+j\cdot k_1}$ and $x u_1^{i+j\cdot k_2}$ belong to the same equivalence class Cl of the equivalence $\equiv$. By appending $v u_2^{i+j\cdot k_1} y$ to $x u_1^{i+j\cdot k_1}$, we obtain $x u_1^{i+j\cdot k_1} v u_2^{i+j\cdot k_1} y \in L(G_A)$, since (x, u_1, v, u_2, y) is a pure pumping infix by G_A. On the other hand, $k_2 = k_1 + m_1$ for some $m_1 > 0$. According to condition (ARl), by appending the same word $v u_2^{i+j\cdot k_1} y$ to $x u_1^{i+j\cdot k_2}$, we obtain that $x u_1^{i+j\cdot k_2} v u_2^{i+j\cdot k_1} y = x u_1^{j\cdot m_1} u_1^{i+j\cdot k_1} v u_2^{i+j\cdot k_1} y$ is in $L(G_R)$. Thus, $x u_1^{i+j\cdot k_1}$ and $x u_1^{i+j\cdot k_2}$ cannot be in the same equivalence class Cl. This contradiction implies that the language $L(G_A)$ is not regular. Since the class of regular languages is closed under the complement and intersection, the language $L(G_R)$ must also be non-regular. That finishes the proof of this case. $\square$

As a direct consequence of Theorem 1, we get the analogous statement for (non-pure) pumping infixes.

Corollary 1. *Let $G_C = (G_A, G_R) = (N, \Sigma \cup \{¢, \$\}, S, R)$ be a complete CF(¢,\$)-grammar, and at least one of the following conditions is fulfilled (for some words $x \in \{¢\} \cdot \Sigma^*$, $u_1, v, u_2 \in \Sigma^*$, $y \in \Sigma^* \cdot \{\$\}$, and a nonterminal $A \in N$):*

(ARl') There exists $w = x u_1 v u_2 y \in L(G_A)$ such that u_1 and u_2 are nonempty, (x, u_1, A, v, u_2, y) is a pumping infix by G_A, and there are integers $i \geq 0, j > 0$ such that

$$x u_1^{j\cdot m} u_1^{i+j\cdot n} v u_2^{i+j\cdot n} y \in L(G_R),$$

for each $m > 0, n \geq 0$.

(ARr') There exists $w = x u_1 v u_2 y \in L(G_A)$ such that u_1 and u_2 are nonempty, (x, u_1, A, v, u_2, y) is a pure pumping infix by G_A, and there are integers $i \geq 0, j > 0$ such that

$$x u_1^{i+j\cdot n} v u_2^{i+j\cdot n} u_2^{j\cdot m} y \in L(G_R),$$

for each $m > 0, n \geq 0$.

(RAl') There exists $w = x u_1 v u_2 y \in L(G_R)$ such that u_1 and u_2 are nonempty, (x, u_1, A, v, u_2, y) is a pure pumping infix by G_R, and there are integers $i \geq 0, j > 0$ such that

$$x u_1^{j\cdot m} u_1^{i+j\cdot n} v u_2^{i+j\cdot n} y \in L(G_A),$$

for each $m > 0, n \geq 0$.

(RAr') There exists $w = x u_1 v u_2 y \in L(G_R)$ such that u_1 and u_2 are nonempty, (x, u_1, A, v, u_2, y) is a pure pumping infix by G_R, and there are integers $i \geq 0, j > 0$ such that

$$x u_1^{i+j\cdot n} v u_2^{i+j\cdot n} u_2^{j\cdot m} y \in L(G_A),$$

for each $m > 0, n \geq 0$.

Then $L(G_A)$ and $L(G_R)$ are not regular languages.

Any of the conditions (ARl), (ARr), (RAl), (RAr), (ARl'), (ARr'), (RAl'), and (RAr') is sufficient for non-regularity of a complete CF(¢,\$)-grammar. Now, we examine whether the previous sufficient conditions for non-regularity are also necessary for non-regularity. We start with the definition of pumping test sets and a rather technical definition of preserving and switching tests. Based on these notions, we get another condition for non-regularity in Theorem 2.

If we know that (x, u_1, v, u_2, y) is a pure pumping infix by a grammar G_A, we have that $x u_1^r v u_2^r y$ is in $L(G_A)$, for all integers $r \geq 0$. This could indicate a context-free dependence between the number of copies of u_1 in front of the factor v and the number of copies of u_2 after the factor v. However, it is still possible that all words of the form $x u_1^m v u_2^n y$ belong to $L(G_A)$. Therefore, in the following, we define a switching test that should detect the situation in which *there is a dependence* between the number of occurrences of u_1 before and the number of occurrences of u_2 after v.

We define two types of test sets. The first one with a subscript 'left' should detect the situation when the number of copies of u_1 can be "pumped" more times than the number of copies of u_2. A symmetric test set should detect the situation where the number of copies of u_2 can be "pumped" more times than the number of copies of u_1.

Let us introduce the 'left' test set. A pumping may require pumping several, say j, copies of u_1 and u_2

simultaneously in one step. Furthermore, several, say i, copies of u_1 and symmetrically of u_2 could be produced together with the prefix x and suffix y, respectively. Hence, the left test set below contains words of the form $xu_1^i u_1^{j\cdot m} u_1^{j\cdot n} v u_2^{j\cdot n} u_2^i y$, for all $m > 0$ and $n \geq 0$.

In order to restrict the set of pumping test sets, we require that i is not greater than $K_{G_C} + 2t$, and j is not greater than $2t$, where t denotes the number of nonterminals of G_C.

Definition 5 (Pumping test set). *Let*

$$G_C = (G_A, G_R)$$

be a complete CF(¢, \$)-grammar with t nonterminals. Let $\iota = (x, u_1, v, u_2, y)$, where $|u_1| > 0$, $|u_2| > 0$, be a pure pumping infix by G_C and i, j be integers such that $i \leq K_{G_C} + 2t$, and $0 < j \leq 2t$:

1. *The set $T_{\text{left}}(\iota, i, j) = \{xu_1^i u_1^{j\cdot m} u_1^{j\cdot n} v u_2^{j\cdot n} u_2^i y \mid m > 0, n \geq 0\}$ is called the* left test set *of ι.*
2. *The set $T_{\text{right}}(\iota, i, j) = \{xu_1^i u_1^{j\cdot n} v u_2^{j\cdot n} u_2^{j\cdot m} u_2^i y \mid m > 0, n \geq 0\}$ is called the* right test set *of ι.*

We say that the triple

$$Tp(G_C, \iota, i, j) = [\iota, T_{\text{left}}(\iota, i, j), T_{\text{right}}(\iota, i, j)]$$

is a pumping test set *by G_C.*

Definition 6 (Preserving/switching test set). *Let $G_C = (G_A, G_R)$ be a complete CF(¢, \$)-grammar with t nonterminals, $\iota = (x, u_1, v, u_2, y)$, where $|u_1| > 0$, $|u_2| > 0$, be a pure pumping infix by G_C, and i, j be integers such that $i \leq K_{G_C} + 2t$ and $0 < j \leq 2t$. We say that the pumping test set $\tau = Tp(G_C, \iota, i, j) = [\iota, T_{\text{left}}(\iota, i, j), T_{\text{right}}(\iota, i, j)]$ is* preserving *if*

(Aaa) *$xu_1vu_2y \in L(G_A)$ and both sets $T_{\text{left}}(\iota, i, j)$ and $T_{\text{right}}(\iota, i, j)$ are subsets of $L(G_A)$; or*

(Rrr) *$xu_1vu_2y \in L(G_R)$ and both sets $T_{\text{left}}(\iota, i, j)$ and $T_{\text{right}}(\iota, i, j)$ are subsets of $L(G_R)$.*

We say that τ is switching *if one of the following two cases is true:*

(AR) *$xu_1vu_2y \in L(G_A)$ and [$T_{\text{left}}(\iota, i, j) \subseteq L(G_R)$ or $T_{\text{right}}(\iota, i, j) \subseteq L(G_R)$],*

(RA) *$xu_1vu_2y \in L(G_R)$ and [$T_{\text{left}}(\iota, i, j) \subseteq L(G_A)$ or $T_{\text{right}}(\iota, i, j) \subseteq L(G_A)$].*

The following theorem is a direct consequence of Theorem 1 and the definition of the switching pumping test set.

Theorem 2. *Let $G_C = (G_A, G_R)$ be a complete CF(¢, \$)-grammar, and suppose there exists a switching pumping test set by G_C. Then $L(G_A)$, and $L(G_R)$ are non-regular languages.*

Proof: We prove that both $L(G_A)$ and $L(G_R)$ are non-regular languages when the condition (AR) holds. The other cases can be shown similarly.

Let $G_C = (G_A, G_R)$ be a complete CF(¢,\$)-grammar with t nonterminals, $\iota = (x, u_1, v, u_2, y)$, where $|u_1| > 0$, $|u_2| > 0$, be a pure pumping infix by G_C and i, j be integers such that $i \leq K_{G_C} + 2t$ and $0 < j \leq 2t$, and

$$\tau = Tp(G_C, \iota, i, j) = [\iota, T_{\text{left}}(\iota, i, j), T_{\text{right}}(\iota, i, j)]$$

be a switching pumping test set such that $xu_1vu_2y \in L(G_A)$ and at least one of the following conditions is true:

1. $T_{\text{left}}(\iota, i, j) \subseteq L(G_R)$, or
2. $T_{\text{right}}(\iota, i, j) \subseteq L(G_R)$.

As $xu_1vu_2y \in L(G_A)$ and $\iota = (x, u_1, v, u_2, y)$ is a pure pumping infix by G_C, ι is a pure pumping infix by G_A.

In case 1, the condition (ARl) of Theorem 1 is satisfied. Hence, according to Theorem 1, both languages $L(G_A)$ and $L(G_R)$ are not regular.

In case 2, the condition (ARr) of Theorem 1 is satisfied. Hence, according to Theorem 1, both languages $L(G_A)$ and $L(G_R)$ are not regular.

Similarly, we can show the case where the condition (RA) holds. □

4. Open problems and future work

Many open problems are left related to our original effort to compare regularity and non-regularity connected with complete CF(¢, \$)-grammars. This section gives a partial idea of our plans for the future. In general, we will try to solve the decidability questions connected with (non-)regularity of complete CF(¢, \$)-grammars.

Test languages. Let $G_C = (G_A, G_R)$ be a complete CF(¢, \$)-grammar. Let u_1 and u_2 be nonempty words, and $\iota = (x, u_1, v, u_2, y)$ be a pure pumping infix by G_C.

We say that the languages

$$L(G_A) \cap \{xu_1^n v u_2^m y \mid n, m \geq 0\} \text{ and}$$
$$L(G_R) \cap \{xu_1^n v u_2^m y \mid n, m \geq 0\}$$

are *test languages* of ι. We also say that the languages are test languages of G_C.

Concerning the test languages, we have several conjectures.

Conjecture 1. *Let* $G_C = (G_A, G_R)$ *be a complete CF*$(¢, \$)$*-grammar. Let* $\iota = (x, u_1, v, u_2, y)$*, where* $|u_1| > 0$ *and* $|u_2| > 0$*, be a pure pumping infix by* G_C*. Let all pumping tests sets* $Tp(G_C, \iota, i, j) = [\iota, T_{\text{left}}(\iota, i, j), T_{\text{right}}(\iota, i, j)]$ *of* ι*, for all integers* i, j*, such that* $i \leq K_{G_C} + 2t$ *and* $0 < j \leq 2t$*, are preserving. Then, the test languages of* ι *are regular.*

Conjecture 2. *Let* $G_C = (G_A, G_R)$ *be a complete CF*$(¢, \$)$*-grammar, and there does not exist any switching pumping test by* G_C*. Then, each test language of* G_C *is regular.*

Conjecture 3. *Let* $G_C = (G_A, G_R)$ *be a complete CF*$(¢, \$)$*-grammar. Then* $L(G_A)$ *and* $L(G_R)$ *are regular if and only if all test languages of* G_C *are regular.*

Remark. Note that the notions of switching test and preserving test give an opportunity to introduce degrees of regularity and degrees for non-regularity of complete CF$(¢, \$)$-grammars. That will also be one direction of our efforts in the future.

References

[1] F. Mráz, D. Pardubská, M. Plátek, J. Šíma, Pumping deterministic monotone restarting automata and DCFL, in: M. Holeňa, T. Horváth, A. Kelemenová, F. Mráz, D. Pardubská, M. Plátek, P. Sosík (Eds.), Proceedings of the 20th Conference Information Technologies – Applications and Theory (ITAT 2020), volume 2718 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 51–58. URL: http://ceur-ws.org/Vol-2718/paper13.pdf.

[2] M. Plátek, F. Mráz, D. Pardubská, D. Průša, J. Šíma, On separations of LR(0)-grammars by two types of pumping patterns, in: B. Brejová, L. Ciencialová, M. Holeňa, F. Mráz, D. Pardubská, M. Plátek, T. Vinař (Eds.), Proceedings of the 21st Conference Information Technologies – Applications and Theory (ITAT 2021), volume 2962 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021, pp. 140–146. URL: http://ceur-ws.org/Vol-2962/paper05.pdf.

[3] 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.

[4] 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.

[5] P. Jančar, J. Šíma, The simplest non-regular deterministic context-free language, in: F. Bonchi, S. J. Puglisi (Eds.), 46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23–27, 2021, Tallinn, Estonia, volume 202 of *LIPIcs*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, pp. 63:1–63:18. doi:10.4230/LIPIcs.MFCS.2021.63.

[6] J. Šíma, M. Plátek, One analog neuron cannot recognize deterministic context-free languages, in: T. Gedeon, K. W. Wong, M. Lee (Eds.), Neural Information Processing – 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part III, volume 11955 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 77–89. doi:10.1007/978-3-030-36718-3_7.

[7] 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.

[8] P. Jančar, F. Mráz, M. Plátek, J. Vogel, Restarting automata, in: H. Reichel (Ed.), Fundamentals of Computation Theory, FCT '95, volume 965 of *Lecture Notes in Computer Science*, Springer, 1995, pp. 283–292. doi:10.1007/3-540-60249-6_60.

[9] P. Jančar, F. Mráz, M. Plátek, J. Vogel, On monotonic automata with a restart operation, J. Autom. Lang. Comb. 4 (1999) 287–311. doi:10.25596/jalc-1999-287.

[10] 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. doi:10.1007/978-3-540-33461-3_11.

[11] J. O. Shallit, A Second Course in Formal Languages and Automata Theory, Cambridge University Press, 2008.

[12] J. Hopcroft, J. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, N. Reading, MA, 1980.