Jumping Pure Grammars

Zbyněk Křivka, Jiří Kučera, Alexander Meduna

Brno University of Technology, Faculty of Information Technology, Centre of Excellence IT4Innovations, Božetěchova 2, 612 66 Brno, Czech Republic
Email: {krivka, ikucera, meduna}@fit.vutbr.cz

This paper introduces and studies *jumping pure grammars*, which are conceptualized just like classical pure grammars except that during the applications of their productions, they can jump over symbols in either direction within the rewritten strings. The paper compares the generative power of jumping pure grammars with that of classical pure grammars while distinguishing between their versions with and without erasing productions. Apart from sequential versions, the paper makes an analogical study in terms of parallel versions of jumping pure grammars represented by 0L grammars.

Keywords: Jumping Grammars; Pure Grammars; Jumping Rewriting; 0L Languages; Parallel Rewriting; Pure Context-free Languages

1. INTRODUCTION

Jumping versions of language-defining rewriting systems, such as grammars and automata, represent a brand new trend in formal language theory (see [16, 10, 4, 9, 2, 5, 8, 15, 17]. In essence, they work just like classical rewriting systems except that they work on strings discontinuously. That is, they apply a production so they erase an occurrence of its left-hand side in the rewritten string while placing the right-hand side anywhere in the string, so the position of the insertion may occur far away from the position of the erasure. The present paper contributes to this trend by investigating the generative power of jumping pure grammars.

Recall that the notion of a pure grammar G represents a language-generating rewriting system based upon an alphabet of symbols and a finite set of productions (as opposed to the notion of a general grammar, its alphabet of symbols is not divided into the alphabet of terminals and the alphabet of nonterminals). Each production represents a pair of the form (x, y), where x and y are strings over the alphabet of G. Customarily, (x, y) is written as $x \to y$, where x and y are referred to as the left-hand side and the right-hand side of $x \to y$, respectively. Starting from a special start string, G repeatedly rewrites strings according to its productions, and the set of all strings obtained in this way represents the language generated by G. In a greater detail, G rewrites a string z according to $x \to y$ so it (i) selects an occurrence of x in z, (ii) erases it, and (iii) inserts y precisely at the position of this erasure. More formally, let $z = uxv$, where u and v are strings. By using $x \to y$, G rewrites uxv as uyv. Recall that pure grammars were introduced in [6], and their properties are still intensively investigated in language theory (see [1, 18]). Recently, regulated versions of these grammars have been discussed, too (see Chapter 5 in [3] and [12, 11]).

The notion of a *jumping pure grammar*—that is, the key notion introduced in this paper—is conceptualized just like that of a classical pure grammar; however, it rewrites strings in a slightly different way. Consider G, described above, as a jumping pure grammar. Let z and $x \to y$ have the same meaning as above. G rewrites a string z according to $x \to y$ so it performs (i) and (ii) as described above, but during (iii), G can jump over a portion of the rewritten string in either direction and inserts y there. More formally, by using $x \to y$, G rewrites ucv as udv, where u, v, w, c, d are strings such that either (a) $c = xw$ and $d = wy$ or (b) $c = wx$ and $d = yw$. Otherwise, it coincides with the standard notion of a grammar.

The present paper compares the generative power of classical and jumping versions of pure grammars. It distinguishes between these grammars with and without erasing productions. Apart from these sequential versions of pure grammars, it also considers parallel versions of classical and jumping pure grammars represented by 0L grammars (see [20]). As a result, the paper studies the mutual relations between eight language families corresponding to the following derivations modes (see Definition 2.1) performed by pure grammars both with and without erasing productions:

- *classical sequential mode* (${}_s\!\Rightarrow$);
- *jumping sequential mode* (${}_j\!\Rightarrow$);
- *classical parallel mode* (${}_p\!\Rightarrow$);
- *jumping parallel mode* (${}_{jp}\!\Rightarrow$).

In essence, the paper demonstrates that any version of these grammars with erasing productions is stronger than the same version without them. Furthermore, it shows that almost all of the eight language families under considerations are pairwise incomparable——that is, any two families are not subfamilies of each other.

The rest of the paper is organized as follows. Section 2 recalls all the terminology needed in this paper and introduces a variety of jumping pure grammars, illustrated by an example. Section 3 presents fundamental results achieved in this paper. Section 4 closes all the study by summing up ten open problems.

2. PRELIMINARIES AND DEFINITIONS

This paper assumes that the reader is familiar with the basic notions of the formal language theory (see [14, 21, 22]). Let A and B be two sets. By $A \subseteq B$, we denote that A is included in B and by $A \not\subseteq B$ that A is not included in B. $A \subset B$ denotes proper (or strict) inclusion. We say that A and B are *incomparable* iff $A \not\subseteq B$ and $B \not\subseteq A$. The cardinality of A is expressed as $\mathrm{card}(A)$. For some $n \geq 0$, A^n denotes the n-fold Cartesian product of set A. By $\mathbb{N}$, we denote the set of all positive integers. Let $I \subset \mathbb{N}$ be a finite nonempty set. Then, $\max I$ denotes a maximum of I. Let ϱ be a (binary) relation over X. By ϱ^i, ϱ^+ and ϱ^* are denoted the ith power of ϱ, for all $i \geq 0$, the transitive closure of ϱ and the reflexive and transitive closure of ϱ, respectively. For $x, y \in X$, instead of $(x, y) \in \varrho$, we write $x\varrho y$ throughout. Set $\mathrm{dom}(\varrho) = \{x \mid x\varrho y\}$. Let Σ be an alphabet (finite nonempty set). Then, Σ^* represents the free monoid generated by Σ under the operation of concatenation, with ε as the unit of Σ^*. Set $\Sigma^+ = \Sigma^* - \{\varepsilon\}$. For $w \in \Sigma^*$ and $a \in \Sigma$, $\#_a(w)$ denotes the number of occurrences of symbol a in w. By $\mathrm{substr}(w)$, we denote a set of all substrings of w, that is $\mathrm{substr}(w) = \{x \mid w = uxv, u, x, v \in \Sigma^*\}$. The length of w is denoted by $|w|$.

Let $n \geq 0$. A set $J \subseteq \mathbb{N}^n$ is said to be *linear* if there exist $\alpha, \beta_1, \beta_2, \ldots, \beta_m \in \mathbb{N}^n, m \geq 0$ such that

$$J = \{x \mid x = \alpha + k_1\beta_1 + k_2\beta_2 + \cdots + k_m\beta_m, \\ k_i \in \mathbb{N}, 1 \leq i \leq m\}.$$

If J is the union of a finite number of linear sets, we say that J is *semilinear*. If $\Sigma = \{a_1, a_2, \ldots, a_n\}$ is an alphabet, then for $w \in \Sigma^*$,

$$\phi(w) = (\#_{a_1}(w), \#_{a_2}(w), \ldots, \#_{a_n}(w))$$

denote the *commutative (Parikh) image of* w. For $L \subseteq \Sigma^*$, $\phi(L) = \{\phi(w) \mid w \in L\}$ denote the *commutative (Parikh) map of* L. We say that L is a *semilinear language* if and only if $\phi(L)$ is a semilinear set. A language family is semilinear if and only if it contains only semilinear languages.

Let S be a finite set. Define a permutation in the terms of bijective mappings as follows: Let $I = \{1, 2, \ldots, \mathrm{card}(S)\}$ be a set of indices. The set of all permutations of elements of S, $\mathrm{perm}(S)$, is a set of bijections from I to S such that $p \in \mathrm{perm}(S)$ iff $p(i) \in S$ for every $i \in I$.

An *unrestricted grammar* is a quadruple $G = (V, \Sigma, P, \sigma)$, where V is a total alphabet, $\Sigma \subseteq V$ is an alphabet of terminal symbols, $P \subseteq V^+ \times V^*$ is a finite relation, and $\sigma \in V^+$ is the start string of G, called *axiom*. Members of P are called *productions*. Instead of $(x, y) \in P$, we write $x \to y$ throughout. For brevity, we sometimes denote a production $x \to y$ with a unique label r as $r\colon x \to y$, and instead of $x \to y \in P$, we simply write $r \in P$. We say that $x \to y$ is a *unit production* if $x, y \in V$. A relation of direct derivation in G, denoted $\Rightarrow$, is defined as follows: If $u, v, x, y \in V^*$ and $x \to y \in P$, then $uxv \Rightarrow uyv$. The language generated by G, denoted $L(G)$, is defined as $L(G) = \{w \mid \sigma \Rightarrow^* w, w \in \Sigma^*\}$. The language generated by G is said to be context-free iff for every production $x \to y \in P$, $|x| = 1$. Furthermore, the language generated by G is said to be context-sensitive iff for every derivation $\sigma \Rightarrow^* z \Rightarrow^* w$ holds $|\sigma| \leq |z| \leq |w|$, $z \in V^*$. By **CF** and **CS**, we denote the family of context-free and context-sensitive languages, respectively.

Next, we give the formal definition of *pure grammar* (see [13, 3]), together with six modes of derivations.

Definition 2.1. *Let $G = (V, \Sigma, P, \sigma)$ be an unrestricted grammar. G is a* pure grammar *(PG for short), if $V = \Sigma$. For brevity, we simplify $G = (V, \Sigma, P, \sigma)$ to $G = (\Sigma, P, \sigma)$. We say that G is* propagating *or* without erasing productions *iff for every production $x \to y \in P$, $y \neq \varepsilon$.*

Next, we introduce the six modes of direct derivation steps *as derivation relations over Σ^*. Let $u, v \in \Sigma^*$. The six derivation relations are defined as follows*

(i) $u \;_s\!\Rightarrow v$ in G iff there exists $x \to y \in P$ and $w, z \in \Sigma^*$ such that $u = wxz$ and $v = wyz$;

(ii) $u \;_{lj}\!\Rightarrow v$ in G iff there exists $x \to y \in P$ and $w, t, z \in \Sigma^*$ such that $u = wtxz$ and $v = wytz$;

(iii) $u \;_{rj}\!\Rightarrow v$ in G iff there exists $x \to y \in P$ and $w, t, z \in \Sigma^*$ such that $u = wxtz$ and $v = wtyz$;

(iv) $u \;_j\!\Rightarrow v$ in G iff $u \;_{lj}\!\Rightarrow v$ or $u \;_{rj}\!\Rightarrow v$ in G;

(v) $u \;_p\!\Rightarrow v$ in G iff there exist $x_1 \to y_1, x_2 \to y_2, \ldots, x_n \to y_n \in P$ such that $u = x_1x_2\ldots x_n$ and $v = y_1y_2\ldots y_n$, where $n \geq 0$;

(vi) $u \;_{jp}\!\Rightarrow v$ in G iff there exist $x_1 \to y_1, x_2 \to y_2, \ldots, x_n \to y_n \in P$ such that $u = x_1x_2\ldots x_n$ and $v = y_{p(1)}y_{p(2)}\ldots y_{p(n)}$, where $p \in$ $\mathrm{perm}(\{1, 2, \ldots, n\})$, $n \geq 0$.

Let $_h\!\Rightarrow$ be one of the six derivation relations (i) through (vi) over Σ^*. To express that G applies production r during $u \;_h\!\Rightarrow v$, we write $u \;_h\!\Rightarrow v\,[r]$, where $r \in P$. By $u \;_h\!\Rightarrow^* v\,[\pi]$, where π is a sequence of productions from P, we express that G makes $u \;_h\!\Rightarrow^* v$ by using π.

The language that G generates by using ${}_h\Rightarrow$, $L(G, {}_h\Rightarrow)$, is defined as

$$L(G, {}_h\Rightarrow) = \{x \mid \sigma \; {}_h\Rightarrow^* x, x \in \Sigma^*\}.$$

The set of all PGs and the set of all PGs without erasing productions are denoted Γ_{PG} and $\Gamma_{\mathrm{PG}^{-\varepsilon}}$, respectively.

Let $G = (\Sigma, P, \sigma)$ be a PG. G is said to be a *pure context-free grammar* (PCFG for short) if every $x \to y \in P$ satisfies $x \in \Sigma$. The set of all PCFGs and the set of all PCFGs without erasing productions are denoted Γ_{PCFG} and $\Gamma_{\mathrm{PCFG}^{-\varepsilon}}$, respectively.

REMARK 1. The inclusions $\Gamma_{\mathrm{PCFG}} \subseteq \Gamma_{\mathrm{PG}}$, $\Gamma_{\mathrm{PCFG}^{-\varepsilon}} \subseteq \Gamma_{\mathrm{PCFG}}$, and $\Gamma_{\mathrm{PG}^{-\varepsilon}} \subseteq \Gamma_{\mathrm{PG}}$ are obvious.

Set

(1) $\mathbf{SP} = \{L(G, {}_s\Rightarrow) \mid G \in \Gamma_{\mathrm{PG}}\}$;

(2) $\mathbf{SP}^{-\varepsilon} = \{L(G, {}_s\Rightarrow) \mid G \in \Gamma_{\mathrm{PG}^{-\varepsilon}}\}$;

(3) $\mathbf{JSP} = \{L(G, {}_j\Rightarrow) \mid G \in \Gamma_{\mathrm{PG}}\}$;

(4) $\mathbf{JSP}^{-\varepsilon} = \{L(G, {}_j\Rightarrow) \mid G \in \Gamma_{\mathrm{PG}^{-\varepsilon}}\}$;

(5) $\mathbf{PP} = \{L(G, {}_p\Rightarrow) \mid G \in \Gamma_{\mathrm{PG}}\}$;

(6) $\mathbf{PP}^{-\varepsilon} = \{L(G, {}_p\Rightarrow) \mid G \in \Gamma_{\mathrm{PG}^{-\varepsilon}}\}$;

(7) $\mathbf{JPP} = \{L(G, {}_{jp}\Rightarrow) \mid G \in \Gamma_{\mathrm{PG}}\}$;

(8) $\mathbf{JPP}^{-\varepsilon} = \{L(G, {}_{jp}\Rightarrow) \mid G \in \Gamma_{\mathrm{PG}^{-\varepsilon}}\}$;

(9) $\mathbf{SPCF} = \{L(G, {}_s\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}}\}$;

(10) $\mathbf{SPCF}^{-\varepsilon} = \{L(G, {}_s\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}^{-\varepsilon}}\}$;

(11) $\mathbf{JSPCF} = \{L(G, {}_j\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}}\}$;

(12) $\mathbf{JSPCF}^{-\varepsilon} = \{L(G, {}_j\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}^{-\varepsilon}}\}$;

(13) $\mathbf{PPCF} = \{L(G, {}_p\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}}\}$;

(14) $\mathbf{PPCF}^{-\varepsilon} = \{L(G, {}_p\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}^{-\varepsilon}}\}$;

(15) $\mathbf{0L} = \{L(G, {}_p\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}}, G = (\Sigma, P, \sigma), \mathrm{dom}(P) = \Sigma\}$ (see [20]);

(16) $\mathbf{0L}^{-\varepsilon} = \{L(G, {}_p\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}^{-\varepsilon}}, G = (\Sigma, P, \sigma), \mathrm{dom}(P) = \Sigma\}$ (see [20]);

(17) $\mathbf{JPPCF} = \{L(G, {}_{jp}\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}}\}$;

(18) $\mathbf{JPPCF}^{-\varepsilon} = \{L(G, {}_{jp}\Rightarrow) \mid G \in \Gamma_{\mathrm{PCFG}^{-\varepsilon}}\}$.

EXAMPLE 1. Consider the following PCFG

$$G = (\Sigma = \{a, b, c, d\}, P, a)$$

where $P = \{a \to abcd, a \to a, b \to b, c \to c, d \to d\}$. Observe that if G makes its derivations by ${}_s\Rightarrow$, we have $L(G, {}_s\Rightarrow) = L(G, {}_p\Rightarrow) = \{a\}\{bcd\}^*$, which is a regular language. But if G performs its derivations by ${}_j\Rightarrow$, we have $L(G, {}_j\Rightarrow) = L(G, {}_{jp}\Rightarrow) = \{w \mid \#_a(w) = 1, \#_b(w) = \#_c(w) = \#_d(w), w \in \Sigma^+\}$, which is a non-context-free language.

3. RESULTS

The organization of this section is divided into three parts. First, we give an overview about several elementary properties of pure grammars. Second, we investigate the mutual relations of **SPCF**, **JSPCF**, **PPCF**, **JPPCF**, **CF**, and **CS** and summarize the results by Venn diagram in Figure 1. Finally, we study the former without erasing productions and sum up the investigated relations in Table 1.

Elementary properties

Many properties about pure grammars can be found in [21, 13]. Recall that[1] $\mathbf{SPCF} \subset \mathbf{CF}$ (see [21, 13]). Furthermore, observe that $\mathbf{0L} \subset \mathbf{PPCF}$; the inclusion $\mathbf{0L} \subseteq \mathbf{PPCF}$ is obvious and in addition there exist languages that can be generated by parallel PCFG but cannot be generated by any 0L system (such a language is, for example, $\{a, aab\}$).

LEMMA 3.1. *Let $X \in \{$**SP**, **JSP**, **PP**, **JPP**, **SPCF**, **JSPCF**, **PPCF**, **JPPCF**, **0L**$\}$. Then, $X^{-\varepsilon} \subseteq X$.*

Proof. Obvious. □

THEOREM 3.1. **SPCF** *and* **JSPCF** *are semilinear.*

Proof. Since $\mathbf{SPCF} \subset \mathbf{CF}$ and **CF** is semilinear (see [19]), **SPCF** must be also semilinear. Consider PCFG $G = (\Sigma, P, \sigma)$. From the definitions of ${}_s\Rightarrow$ and ${}_j\Rightarrow$ it follows that $\phi(L(G, {}_s\Rightarrow)) = \phi(L(G, {}_j\Rightarrow))$. Thus, **JSPCF** is semilinear as well. □

THEOREM 3.2. $\mathbf{SPCF} \subset \mathbf{PPCF}$.

Proof. First, we proof the inclusion $\mathbf{SPCF} \subseteq \mathbf{PPCF}$. The proof is based on the proof of Theorem 4.2 in [20]. Let Σ be an alphabet. We claim that for every PCFG $G = (\Sigma, P, \sigma)$ such that $L(G, {}_s\Rightarrow) \in \mathbf{SPCF}$, there is a PCFG $G' = (\Sigma, P', \sigma')$ such that $L(G', {}_p\Rightarrow) = L(G, {}_s\Rightarrow)$. Set

$$P' = P \cup \{a \to a \mid a \in \Sigma\} \quad \text{and} \quad \sigma' = \sigma$$

Now, we prove the following two claims.

CLAIM 3.3. Let $\sigma \; {}_s\Rightarrow^m w$ in G, where $w \in \Sigma^*$. Then $\sigma' \; {}_p\Rightarrow^* w$ in G'.

Proof. This claim is proved by induction on $m \geq 0$.

Basis. For $m = 0$, we have $\sigma \; {}_s\Rightarrow^0 \sigma$ in G. Since $\sigma = \sigma'$, we also have $\sigma' \; {}_p\Rightarrow^* \sigma$ and the basis holds.

Induction Hypothesis. Assume that the claim holds for all $0 \leq m \leq k$, where k is a non-negative integer.

Induction Step. Let $\sigma \; {}_s\Rightarrow^{k+1} w$ in G, where $w \in \Sigma^*$. Express $\sigma \; {}_s\Rightarrow^{k+1} w$ as $\sigma \; {}_s\Rightarrow^k uav \; {}_s\Rightarrow uxv$, where $u, v, x \in \Sigma^*$, $a \in \Sigma$, $a \to x \in P$, and $uxv = w$. By the induction hypothesis, there exists a derivation $\sigma' \; {}_p\Rightarrow^* uav$ in G'.

[1] According to its definition, **SPCF** in this paper coincides with PCF in [13].

Since $P \subseteq P'$ and there are also unit productions $b \to b \in P'$, for every $b \in \Sigma$, clearly $uav \,_p\!\Rightarrow uxv$ in G', which completes the induction step. □

Claim 3.4. Let $\sigma' \,_p\!\Rightarrow^m w$ in G', where $w \in \Sigma^*$. Then $\sigma \,_s\!\Rightarrow^* w$ in G.

Proof. This claim is proved by induction on $m \geq 0$.

Basis. Let $m = 0$. Then $\sigma' \,_p\!\Rightarrow^0 \sigma'$ in G'. Since $\sigma' = \sigma$, we have that $\sigma \,_s\!\Rightarrow^* \sigma'$ in G and the basis holds.

Induction Hypothesis. Assume that the claim holds for all $0 \leq m \leq k$, where k is a non-negative integer.

Induction Step. Let $\sigma' \,_p\!\Rightarrow^{k+1} w$ in G', where $w \in \Sigma^*$. Express $\sigma' \,_p\!\Rightarrow^{k+1} w$ as $\sigma' \,_p\!\Rightarrow^k x \,_p\!\Rightarrow w$, where $x \in \Sigma^*$. Set $n = |x|$. Express x and w as $x = a_1a_2 \ldots a_n$ and $w = y_1y_2 \ldots y_n$, respectively, where $a_i \in \Sigma$, $y_i \in \Sigma^*$, and $a_i \to y_i \in P'$, $1 \leq i \leq n$. Observe that $a_i \to y_i \in P'$ and $a_i \neq y_i$ implies $a_i \to y_i \in P$, for all $1 \leq i \leq n$. Thus, $x \,_s\!\Rightarrow^* w$ in G. By the induction hypothesis, we have that $\sigma \,_s\!\Rightarrow^* x$ in G, which completes the induction step. □

By Claim 3.3 and Claim 3.4, $\sigma \,_s\!\Rightarrow^* w$ in G iff $\sigma' \,_p\!\Rightarrow^* w$ in G', that is $L(G, {}_s\!\Rightarrow) = L(G', {}_p\!\Rightarrow)$ and therefore **SPCF** $\subseteq$ **PPCF**. By Theorem 4.7 in [20], **0L** $\not\subseteq$ **CF**. Clearly, **0L** $\not\subseteq$ **SPCF**. Since **0L** $\subset$ **PPCF**, **PPCF** $\not\subseteq$ **SPCF** and hence **SPCF** $\subset$ **PPCF**. □

Corollary 3.1. **SPCF** $\subset$ **0L**.

Proof. Observe that G' from the proof of Theorem 3.2 is a correctly defined 0L system according to p. 304 in [20]. □

Theorem 3.5. **SPCF** $\subset$ **CF** $\cap$ **PPCF**.

Proof. **SPCF** $\subseteq$ **CF** $\cap$ **PPCF** is a consequence of recalled inclusion **SPCF** $\subset$ **CF** and Theorem 3.2. Let $\Sigma = \{a, b, c, d\}$ be an alphabet and $L = \{ab, ccdd\}$ be a language over Σ. Clearly, $L \in$ **CF** and also $L \in$ **PPCF** since there is a PCFG

$$G = (\Sigma, \{a \to cc, b \to dd, c \to c, d \to d\}, ab)$$

such that $L = L(G, {}_p\!\Rightarrow)$. We show by contradiction that there is no PCFG $G' = (\Sigma, P', \sigma)$ such that $L(G', {}_s\!\Rightarrow) = L$. Clearly, σ must be either ab or $ccdd$. If we take $ccdd$ as the axiom, there must be $c \to \varepsilon$ or $d \to \varepsilon$ in P' and hence cdd or ccd are contained in L, which is a contradiction. On the other hand, if we take ab, there is no possible way how to directly derive $ccdd$ from ab by using ${}_s\!\Rightarrow$. Hence $L \notin$ **SPCF**, which completes the proof. □

Corollary 3.2. **SPCF** $\subset$ **CF** $\cap$ **0L**.

Theorem 3.6. *For a unary alphabet,* **0L** = **PPCF** = **JPPCF**.

Proof. It follows directly from the definition of ${}_p\!\Rightarrow$ and ${}_{jp}\!\Rightarrow$ and from the definition of $\Rightarrow$ in 0L systems (see [20]). □

Theorem 3.7. *For a unary alphabet,* **SPCF** = **JSPCF**.

Proof. It follows directly from the definition of ${}_s\!\Rightarrow$ and ${}_j\!\Rightarrow$. □

We recall the following lemma from [20].

Lemma 3.2 (Rozenberg, Doucet). *Let G be a 0L system. Then there exists a number k such that for every string w in $L(G)$ there exists a derivation such that $|u| \leq k|w|$ for every string u in that derivation.*

Since Lemma 3.2 relies only on the lengths of derived strings, it is natural to extend the lemma also for PCFGs.

Lemma 3.3. *Let G be a PCFG. Let $h \in \{s, j, p, jp\}$. Then there exists a number k such that for every string w in $L(G, {}_h\!\Rightarrow)$ there exists a derivation such that $|u| \leq k|w|$ for every string u in that derivation.*

Lemma 3.4. **CS** $-$ **JPPCF** $\neq \emptyset$.

Proof. The language $X = \{a^p \mid p \text{ is a prime}\}$ over a unary alphabet $\{a\}$ is a well-known context-sensitive non-context-free language (see [7]). By contradiction, we show that $X \notin$ **JPPCF**. Assume that there is a PCFG $G = (\{a\}, P, \sigma)$ such that $L(G, {}_{jp}\!\Rightarrow) = X$. Obviously $a \to \varepsilon \notin P$ and $\sigma = a^2$ since 2 is the smallest prime. As 3 is also prime, $a^2 \,_{jp}\!\Rightarrow^* a^3$ and we have $a \to a \in P$ and $a \to a^2 \in P$. Thus, $a^2 \,_{jp}\!\Rightarrow^* a^4$. Since 4 is not a prime, we have a contradiction. □

Corollary 3.3. **CS** $-$ **JSPCF** $\neq \emptyset$.

Proof. From Lemma 3.4, we have that $X = \{a^p \mid p \text{ is a prime}\}$ is not contained in **JPPCF**. Since X is a unary language and for unary languages holds **JSPCF** = **SPCF** $\subset$ **PPCF** = **JPPCF** (see Theorems 3.2, 3.6 and 3.7), we have that $X \notin$ **JSPCF**. □

Theorem 3.8. **JPPCF** $\subset$ **CS**.

Proof. Let $G = (\Sigma, P, \sigma)$ be a PCFG. By the Church-Turing thesis there is an unrestricted grammar $H = (V, \Sigma, P', S)$ such that $L(H) = L(G, {}_{jp}\!\Rightarrow)$. More precisely, we are able to construct H in the way that H simulates G. In this case, Lemma 3.3 also holds for H. Observe that Lemma 3.3 is the workspace theorem, and every language from **JPPCF** must be then context-sensitive.

As **CS** $-$ **JPPCF** $\neq \emptyset$ by Lemma 3.4, we have **JPPCF** $\subset$ **CS**. □

Theorem 3.9. **JSPCF** $\subset$ **CS**.

Proof. **JSPCF** $\subseteq$ **CS** can be proved analogously as **JPPCF** $\subseteq$ **CS** from Theorem 3.8. Together with Corollary 3.3, we have **JSPCF** $\subset$ **CS**. □

Mutual relations of SPCF, JSPCF, PPCF, JPPCF, CF, and CS

Now, we investigate 20 mutual relations between **SPCF**, **JSPCF**, **PPCF**, **JPPCF**, **CF**, and **CS**. We refer them as subfamilies A through T. Seven of these relations are stated as open problems.

THEOREM 3.10 (Subfamily A).

$$\mathbf{PPCF} - (\mathbf{CF} \cup \mathbf{JSPCF} \cup \mathbf{JPPCF}) \neq \emptyset$$

Proof. Let $\Sigma = \{a, b\}$ be an alphabet. Let $X = \{a^{2^n}b^{2^n} \mid n \geq 0\}$ be a language over Σ. Clearly, $X \in \mathbf{PPCF}$, since there exists a PCFG, $G = (\Sigma, \{a \to aa, b \to bb\}, ab)$, such that $L(G, {}_p\Rightarrow) = X$. $X \notin \mathbf{CF}$ and $X \notin \mathbf{JSPCF}$ is satisfied since X is not semilinear. By contradiction, we show that $X \notin \mathbf{JPPCF}$.

Consider that there is a PCFG, $G' = (\Sigma, P', \sigma')$, such that $L(G', {}_{jp}\Rightarrow) = X$. Observe that $ab \in L(G', {}_{jp}\Rightarrow)$. Let $a \to x$, $b \to y$ be productions from P', $x, y \in \Sigma^*$. Then, there exist two derivations, $ab \;{}_{jp}\Rightarrow xy$ and $ab \;{}_{jp}\Rightarrow yx$, in G'. Now, consider the following cases:

- $x = y = \varepsilon$. Then, $\varepsilon \in X$, which is a contradiction.
- $x = \varepsilon$ ($y = \varepsilon$). Then, $y \in X$ ($x \in X$) and either ab is the only string derivable in G' using ${}_{jp}\Rightarrow$ or there is a derivation $y \;{}_{jp}\Rightarrow^* z$ ($x \;{}_{jp}\Rightarrow^* z$) in G' such that $ba \in \mathrm{substr}(z)$, which is a contradiction.

In the following, we assume that $x \neq \varepsilon$ and $y \neq \varepsilon$.

- $x = bx'$ or $y = by'$, where $x', y' \in \Sigma^*$. Then, there is a derivation $ab \;{}_{jp}\Rightarrow bz$ in G', where $z \in \Sigma^*$, and thus $bz \in X$, which is a contradiction.
- $x = x'a$ or $y = y'a$, where $x', y' \in \Sigma^*$. Then, there is a derivation $ab \;{}_{jp}\Rightarrow za$ in G', where $z \in \Sigma^*$, and thus $za \in X$, which is a contradiction.
- $x = ax'b$ and $y = ay'b$, where $x', y' \in \Sigma^*$. Then, there is a derivation $ab \;{}_{jp}\Rightarrow z$ in G' such that $ba \in \mathrm{substr}(z)$, which is a contradiction.

No other cases are possible, which completes the proof. □

Several intersections of some language families are hard to investigate. Such an intersection is **PPCF** ∩ **JSPCF**. At this moment, we are not able to prove whether $\mathbf{PPCF} \cap \mathbf{JSPCF} \subseteq \mathbf{CF}$ or not. For this reason, we leave the subfamilies B and C as open problems.

OPEN PROBLEM 3.11 (Subfamily B). Is it true that

$$(\mathbf{PPCF} \cap \mathbf{JSPCF}) - (\mathbf{CF} \cup \mathbf{JPPCF}) \neq \emptyset?$$

OPEN PROBLEM 3.12 (Subfamily C). Is it true that

$$(\mathbf{PPCF} \cap \mathbf{JSPCF} \cap \mathbf{JPPCF}) - \mathbf{CF} \neq \emptyset?$$

THEOREM 3.13 (Subfamily D).

$$(\mathbf{PPCF} \cap \mathbf{JPPCF}) - (\mathbf{CF} \cup \mathbf{JSPCF}) \neq \emptyset$$

Proof. For unary alphabet, $\mathbf{0L} = \mathbf{PPCF} = \mathbf{JPPCF}$ (Theorem 3.6). Since **CF** and **JSPCF** are both semilinear, it is sufficient to find any non-semilinear language over unary alphabet which is also contained in **PPCF**. Such a language is indisputably $\{a^{2^n} \mid n \geq 0\}$. □

THEOREM 3.14 (Subfamily E).

$$\mathbf{SPCF} - (\mathbf{JSPCF} \cup \mathbf{JPPCF}) \neq \emptyset$$

Proof. Let $\Sigma = \{a, b, c\}$ be an alphabet. Let $X = \{a^n cb^n \mid n \geq 0\}$ be a language over Σ. Clearly, there exists a PCFG $G = (\Sigma, \{c \to acb\}, c)$ such that $L(G, {}_s\Rightarrow) = X$ and hence $X \in \mathbf{SPCF}$. We prove by contradiction that X is neither jumping sequential pure context-free nor jumping parallel pure context-free language.

$X \notin \mathbf{JSPCF}$. Assume that there is a PCFG $G' = (\Sigma, P', \sigma')$ such that

$$L(G', {}_j\Rightarrow) = X.$$

Clearly, $\sigma' = c$ must be the axiom since there must be no erasing productions in P' (observe that $ab, ac, cb \notin X$). Because $acb \in X$, we have that $c \to acb \in P'$. But $acb \;{}_j\Rightarrow abacb$ and $abacb \notin X$. A contradiction.

$X \notin \mathbf{JPPCF}$. Assume that there is a PCFG $H = (\Sigma, R, \omega)$ such that $L(H, {}_{jp}\Rightarrow) = X$. First, let $k \geq 1$ and assume that $\omega = a^k cb^k$ is an axiom. Since $\omega \;{}_{jp}\Rightarrow^* c$, there must be a productions $a \to \varepsilon$, $b \to \varepsilon$, and $c \to c$ contained in R. Now, assume that

- $\hat{d} \to dx \in R$, $\hat{d} \in \{a, b\}$, $d \in \Sigma$, $x \in \Sigma^*$; then, $\omega \;{}_{jp}\Rightarrow^* udxcv$ and $\omega \;{}_{jp}\Rightarrow^* ucdxv$ and obviously for $d = a$ holds $ucdxv \notin X$ and for $d = b$ holds $udxcv \notin X$, $u, v \in \Sigma^*$; $d = c$ is obvious;
- $\hat{d} \to xd \in R$, $\hat{d} \in \{a, b\}$, $d \in \Sigma$, $x \in \Sigma^*$; then, $\omega \;{}_{jp}\Rightarrow^* uxdcv$ and $\omega \;{}_{jp}\Rightarrow^* ucxdv$ and obviously for $d = a$ holds $ucxdv \notin X$ and for $d = b$ holds $uxdcv \notin X$, $u, v \in \Sigma^*$; $d = c$ is obvious.

Therefore, $a \to x, b \to y \in R$ implies $x = y = \varepsilon$. Hence, only productions of the form $c \to z$, where $z \in X$, can be considered. But the finiteness of R implies the finiteness of X, which is a contradiction.

Clearly, the axiom must be $\omega = c$, which implies that R contains productions of the form $c \to z$, where $z \in X$. Obviously, there must be also productions $a \to x, b \to y \in R$, $x, y \in \Sigma^*$. If $x = y = \varepsilon$, X must be finite. Thus, assume that $x \neq \varepsilon$ or $y \neq \varepsilon$. Then, like before, we can derive a string which is not contained in X. A contradiction. □

THEOREM 3.15 (Subfamily F).

$$(\mathbf{SPCF} \cap \mathbf{JSPCF}) - \mathbf{JPPCF} \neq \emptyset$$

Proof. Let $\Sigma = \{a, b, c\}$ be an alphabet and let $X = \{aa, aab, aac, aabc\}$ be a language over Σ. Consider a PCFG

$$G = (\Sigma, \{b \to \varepsilon, c \to \varepsilon\}, aabc).$$

Clearly, $L(G, {}_s\Rightarrow) = L(G, {}_j\Rightarrow) = X$ and hence $X \in$ **SPCF** $\cap$ **JSPCF**.

To show that $X \notin$ **JPPCF**, we use a contradiction. Assume that there exists a PCFG $G' = (\Sigma, P', \sigma)$ such that $L(G', {}_{jp}\Rightarrow) = X$. Since $\sigma \in X$ and $X \subseteq \{aa\}\{b\}^*\{c\}^*$, there must be a production $a \rightarrow x$ in P' with $x \in \Sigma^*$. But this implies that there must be a derivation $\sigma \ {}_{jp}\Rightarrow^* aa \ {}_{jp}\Rightarrow xx$ in G'. The only string from X that has a form xx is aa so $a \rightarrow a$ is the only production with a on its left-hand side so $a \rightarrow a \in P'$.

Next, we choose σ. Clearly, $\sigma \neq aa$. Furthermore, $\sigma \notin \{aab, aac\}$ since $\sigma \ {}_{jp}\Rightarrow aabc$ implies that $\sigma \ {}_{jp}\Rightarrow^* abca$, and $abca \notin X$. Thus, the only possibility is to choose $\sigma = aabc$. But $aabc \ {}_{jp}\Rightarrow aab$ means that $\{b \rightarrow b, c \rightarrow \varepsilon\} \subseteq P'$ or $\{b \rightarrow \varepsilon, c \rightarrow b\} \subseteq P'$. In both cases, $aabc \ {}_{jp}\Rightarrow aba$. As $aba \notin X$, there is no PCFG G' such that $L(G', {}_{jp}\Rightarrow) = X$. A contradiction. $\square$

Theorem 3.16 (Subfamily G).

$$\mathbf{SPCF} \cap \mathbf{JSPCF} \cap \mathbf{JPPCF} \neq \emptyset$$

Proof. Let $G = (\{a\}, \{a \rightarrow a, a \rightarrow aa\}, a)$ be a PCFG. It is easy to see that

$$L(G, {}_s\Rightarrow) = L(G, {}_j\Rightarrow) = L(G, {}_{jp}\Rightarrow) = \{a\}^+.$$

$\square$

Open Problem 3.17 (Subfamily H). Is it true that

$$(\mathbf{SPCF} \cap \mathbf{JPPCF}) - \mathbf{JSPCF} \neq \emptyset?$$

Theorem 3.18 (Subfamily I).

$$(\mathbf{PPCF} \cap \mathbf{CF}) - (\mathbf{SPCF} \cup \mathbf{JSPCF} \cup \mathbf{JPPCF}) \neq \emptyset$$

Proof. Let $X = \{aabb, ccdd\}$ be a language over an alphabet $\Sigma = \{a, b, c, d\}$. Clearly, $X \in$ **CF**. Since there exists a PCFG $G = (\Sigma, \{a \rightarrow c, b \rightarrow d\}, aabb)$ such that $L(G, {}_p\Rightarrow) = X$, $X \in$ **PPCF**. Furthermore, observe that derivations $aabb \ {}_s\Rightarrow ccdd$ ($aabb \ {}_j\Rightarrow ccdd$) or $ccdd \ {}_s\Rightarrow aabb$ ($ccdd \ {}_j\Rightarrow aabb$) cannot be performed due to the definition of ${}_s\Rightarrow$ (${}_j\Rightarrow$) and hence there is no PCFG G' such that $L(G', {}_s\Rightarrow) = X$ ($L(G', {}_j\Rightarrow) = X$). Thus, $X \notin$ **SPCF** and $X \notin$ **JSPCF**.

Now, suppose that there is a PCFG $H = (\Sigma, P, \sigma)$ such that $L(H, {}_{jp}\Rightarrow) = X$. For $\sigma = aabb$, we have $aabb \ {}_{jp}\Rightarrow ccdd$. If $a \rightarrow \varepsilon \in P$ or $b \rightarrow \varepsilon \in P$, then $aabb \ {}_{jp}\Rightarrow x$, where $x \notin X$. Thus, $a \rightarrow y$ and $b \rightarrow z$, where $y, z \in \{c, d\}$, are only possible productions in P. But $aabb \ {}_{jp}\Rightarrow cdcd$ and since $cdcd \notin X$, there is no PCFG H such that $L(H, {}_{jp}\Rightarrow) = X$. Analogously for $\sigma = ccdd$. We have a contradiction and therefore $X \notin$ **JPPCF**. $\square$

Open Problem 3.19 (Subfamily J). Is it true that

$$(\mathbf{PPCF} \cap \mathbf{CF} \cap \mathbf{JSPCF}) - (\mathbf{SPCF} \cup \mathbf{JPPCF}) \neq \emptyset?$$

Open Problem 3.20 (Subfamily K). Is it true that

$$(\mathbf{PPCF} \cap \mathbf{CF} \cap \mathbf{JSPCF} \cap \mathbf{JPPCF}) - \mathbf{SPCF} \neq \emptyset?$$

Theorem 3.21 (Subfamily L).

$$(\mathbf{PPCF} \cap \mathbf{CF} \cap \mathbf{JPPCF}) - (\mathbf{SPCF} \cup \mathbf{JSPCF}) \neq \emptyset$$

Proof. Consider a language $X = \{ab, cd, dc\}$ over an alphabet $\Sigma = \{a, b, c, d\}$. Clearly, X is neither classical sequential pure context-free nor jumping sequential pure context-free language since in some point during a derivation, we must rewrite two symbols simultaneously.

As X is a finite language, $X \in$ **CF**. As there exists a PCFG

$$G = (\Sigma, \{a \rightarrow c, b \rightarrow d, c \rightarrow d, d \rightarrow c\}, ab)$$

such that $L(G, {}_p\Rightarrow) = L(G, {}_{jp}\Rightarrow) = X$, $X \in$ **PPCF** $\cap$ **JPPCF**. $\square$

Theorem 3.22 (Subfamily M).

$$\mathbf{CF} - (\mathbf{PPCF} \cup \mathbf{JSPCF} \cup \mathbf{JPPCF}) \neq \emptyset$$

Proof. Let $\Sigma = \{a, b\}$ and let $X = \{a^n b^n \mid n \geq 1\}$ be a language over Σ. Indisputably, X is well-known context-free language. According to [20], $X \notin$ **0L**. Observe that every language Y that belongs to (**PPCF** $-$ **0L**) can be generated by PCFG $G = (\Sigma, P, \sigma)$ such that there exists $c \in \Sigma$ such that for every $x \in \Sigma^*$, $c \rightarrow x \notin P$. Thus, if $X \in$ (**PPCF** $-$ **0L**), then X must be a finite language (since either a or b blocks deriving of any string from axiom), which is a contradiction. Therefore, $X \notin$ (**PPCF** $-$ **0L**) and clearly $X \notin$ **PPCF**. Next, we demonstrate that $X \notin$ **JSPCF** and $X \notin$ **JPPCF**.

$X \notin$ **JSPCF**. Suppose that $X \in$ **JSPCF**, so there exists a PCFG $G' = (\Sigma, P', \sigma')$ such that $L(G', {}_j\Rightarrow) = X$. As $a, b \notin X$, there are no erasing productions in P' and thus $\sigma' = ab$ must be the axiom. Now consider a derivation $ab \ {}_j\Rightarrow aabb$. There are exactly two possibilities how to get a string $aabb$ directly from the axiom ab—either expand a to aab ($a \rightarrow aab \in P'$) or expand b to abb ($b \rightarrow abb \in P'$). Due to the definition of ${}_j\Rightarrow$, $ab \ {}_j\Rightarrow baab$ in the first case, and $ab \ {}_j\Rightarrow abba$ in the second case. Since neither $baab$ nor $abba$ belongs to X, $X \notin$ **JSPCF**. A contradiction.

$X \notin$ **JPPCF**. Suppose that $X \in$ **JPPCF**, so there exists a PCFG $H = (\Sigma, R, \omega)$ such that $L(H, {}_{jp}\Rightarrow) = X$. As $a, b, \varepsilon \notin X$, there are no erasing productions in R and thus $\omega = ab$ must be the axiom. Clearly, $ab \ {}_{jp}\Rightarrow aabb$. There are exactly three ways how to get $aabb$ from ab:

- $a \rightarrow a \in R$, $b \rightarrow abb \in R$. In this case $ab \ {}_{jp}\Rightarrow aabb$ implies that $ab \ {}_{jp}\Rightarrow abba$, but $abba \notin X$.
- $a \rightarrow aa \in R$, $b \rightarrow bb \in R$. In this case $ab \ {}_{jp}\Rightarrow aabb$ implies that $ab \ {}_{jp}\Rightarrow bbaa$, but $bbaa \notin X$.
- $a \rightarrow aab \in R$, $b \rightarrow b \in R$. In this case $ab \ {}_{jp}\Rightarrow aabb$ implies that $ab \ {}_{jp}\Rightarrow baab$, but $baab \notin X$.

Thus, $X \notin$ **JPPCF**. A contradiction. $\square$

OPEN PROBLEM 3.23 (Subfamily N). Is it true that

$$(\mathbf{CF} \cap \mathbf{JSPCF}) - (\mathbf{PPCF} \cup \mathbf{JPPCF}) \neq \emptyset?$$

THEOREM 3.24 (Subfamily O).

$$(\mathbf{CF} \cap \mathbf{JSPCF} \cap \mathbf{JPPCF}) - \mathbf{PPCF} \neq \emptyset$$

Proof. Let $\Sigma = \{a, b\}$ be an alphabet and let

$$X = \{aabb, abab, abba, baab, baba, bbaa\}$$

be a language over Σ. Since X is finite, X is context-free. Given a PCFG

$$G = (\Sigma, \{a \to a, b \to b\}, aabb).$$

Clearly, $L(G, {}_j\!\Rightarrow) = L(G, {}_{jp}\!\Rightarrow) = X$. Hence, $X \in \mathbf{CF} \cap \mathbf{JSPCF} \cap \mathbf{JPPCF}$.

By contradiction, we show that $X \notin \mathbf{PPCF}$. Assume that there is a PCFG $H = (\Sigma, P, \sigma)$ such that $L(H, {}_p\!\Rightarrow) = X$. First, we show that P contains no erasing productions:

- If $a \to \varepsilon \in P$ and $b \to \varepsilon \in P$, we have $\varepsilon \in X$, which is a contradiction.
- If $a \to \varepsilon \in P$, then $b \to x \in P$ implies that $x \in \{aa, bb, ab, ba\}$, since for every $w \in X$, $|w| = 4$. Clearly, if $b \to aa \in P$ or $b \to bb \in P$, then $aaaa \in X$ or $bbbb \in X$, respectively, which is a contradiction. On the other hand, if $b \to ab \in P$ and $b \to ba \in P$, then $aabb \notin X$. A contradiction. Similarly for $b \to \varepsilon \in P$.

Since all strings in X have the same length and there are no erasing productions in P, only unit productions can be contained in P. Because $aaaa \notin X$ and $bbbb \notin X$, either $P = \{a \to a, b \to b\}$ or $P = \{a \to b, b \to a\}$. In both cases, we never get X. Thus, there is no PCFG H such that $L(H, {}_p\!\Rightarrow) = X$, and hence $X \notin \mathbf{PPCF}$. □

THEOREM 3.25 (Subfamily P).

$$(\mathbf{CF} \cap \mathbf{JPPCF}) - (\mathbf{PPCF} \cup \mathbf{JSPCF}) \neq \emptyset$$

Proof. Consider a language $Y = \{aabb, ccdd, cdcd, cddc, dccd, dcdc, ddcc\}$ over an alphabet $\Sigma = \{a, b, c, d\}$. Clearly, $Y \in \mathbf{CF}$ and also $Y \in \mathbf{JPPCF}$ because there is a PCFG

$$G = (\Sigma, \{a \to c, b \to d, c \to c, d \to d\}, aabb)$$

such that $L(G, {}_{jp}\!\Rightarrow) = Y$. The proof that $Y \notin \mathbf{PPCF}$ is almost identical to the proof that $X \notin \mathbf{PPCF}$ from Theorem 3.24, so it is omitted. Because it is not possible to rewrite two or more symbols simultaneously during direct derivation step by using ${}_j\!\Rightarrow$, we have $Y \notin \mathbf{JSPCF}$. □

OPEN PROBLEM 3.26 (Subfamily Q). Is it true that

$$\mathbf{JSPCF} - (\mathbf{CF} \cup \mathbf{PPCF} \cup \mathbf{JPPCF}) \neq \emptyset?$$

THEOREM 3.27 (Subfamily R).

$$(\mathbf{JSPCF} \cap \mathbf{JPPCF}) - (\mathbf{CF} \cup \mathbf{PPCF}) \neq \emptyset$$

Proof. Let $\Sigma = \{a, b, c\}$ be an alphabet and let $X = \{w \mid \#_a(w) - 1 = \#_b(w) = \#_c(w), w \in \Sigma^+\}$ be a language over Σ. $X \in \mathbf{JSPCF} \cap \mathbf{JPPCF}$ since there is a PCFG

$$G = (\Sigma, \{a \to abca, a \to a, b \to b, c \to c\}, a)$$

such that $L(G, {}_j\!\Rightarrow) = L(G, {}_{jp}\!\Rightarrow) = X$. By pumping lemma for context-free languages, $X \notin \mathbf{CF}$.

By contradiction, we show that $X \notin \mathbf{PPCF}$. Assume that there is a PCFG $H = (\Sigma, P, \sigma)$ such that $L(H, {}_p\!\Rightarrow) = X$. First, we show that $\sigma = a$. Assume that $\sigma \neq a$. Then, $\sigma \ {}_p\!\Rightarrow^* a$ implies that $a \to \varepsilon \in P$ and we have that $\varepsilon \in X$, which is a contradiction. Thus, a must be the axiom. Next, we make the following observations:

(1) $a \to a \in P$; if $a \to a \notin P$, then the lengths of strings from X grow exponentially, since, in this case, for a direct derivation step $x \ {}_p\!\Rightarrow y$, $x, y \in \Sigma^*$, and for some $k \geq 2$ it holds that $\#_a(y) = k\#_a(x)$. Note that $a \to b, a \to c \notin P$ from the definition of X.

(2) If $a \to x \in P$, then $\#_a(x) - 1 = \#_b(x) = \#_c(x)$ as follows from the definition of X.

(3) Let $\bar{b}, \bar{c} \in \{b, c\}$, $\bar{b} \neq \bar{c}$. If $\bar{b}\bar{c} \ {}_p\!\Rightarrow x$, then $\#_a(x) = \#_b(x) - 1 = \#_c(x) - 1$; otherwise, we get a string which is not in X.

Let $l = 3 \max\{|\beta| \mid \alpha \to \beta \in P\}$. Let $\omega = a^{l+1}b^lc^l$. Clearly, $\omega \in X$. Express ω as

$$\omega = u_a u_{ab} u_b u_{bc} u_c,$$

where $u_a \in \{a\}^*$, $u_{ab} \in \{a\}^*\{b\}^*$, $u_b \in \{b\}^*$, $u_{bc} \in \{b\}^*\{c\}^*$, and $u_c \in \{c\}^*$. Due to l, there is no suitable production $a \to a^{n+1}b^nc^n$, $n \geq 1$ to get ω. Then, during the derivation $\theta \ {}_p\!\Rightarrow \omega$ the only used production with a on its left-hand side is $a \to a$, so θ can be expressed as $\theta = u_a\bar{x}$, $\bar{x} \in \{b, c\}^*$. Now, we inspect the ways how H derives $u_{ab}u_bu_{bc}u_c$ from $\bar{x}$. Let $\bar{b}, \bar{c}$ be from observation (3).

- If $u_{ab} \in \{a\}^*$, there exists an integer $n \geq 0$ such that $r\colon \bar{b} \to a^n \in P$. The presence of r in P implies that $\bar{c} \to b^{n+1}c^{n+1} \in P$. Due to l, $u_b \neq \varepsilon$ and $u_c \neq \varepsilon$, so there must be productions $d_1 \to b^i, d_2 \to c^j \in P$, $i, j \geq 1$, $d_1, d_2 \in \{b, c\}$. Since either $d_1 = \bar{b}$ or $d_1 = \bar{c}$, there is a derivation

  $$a \ {}_p\!\Rightarrow^* aa\bar{b}\bar{c} \ {}_p\!\Rightarrow^* aab^ib^{n+1}c^{n+1}$$

  or

  $$a \ {}_p\!\Rightarrow^* aa\bar{b}\bar{c} \ {}_p\!\Rightarrow^* aaa^nb^i,$$

  respectively. As $aab^ib^{n+1}c^{n+1}, aaa^nb^i \notin X$, we have a contradiction.

- If $u_{ab} \in \{b\}^*$, there exists an integer $n \geq 1$ such that $\bar{b} \to b^n, \bar{c} \to a^{n-1}c^n \in P$. In fact, $n = 1$ is the only possibility in this case, so either $\theta = \omega$ or $|\theta| = |\omega|$ and $\theta\ {}_p\!\Rightarrow \omega\ {}_p\!\Rightarrow \theta$. This means that one of θ, ω must be the axiom. A contradiction.
- If $u_{ab} \in \{a\}^+\{b\}^+$, there are two integers $n, m \geq 1$ such that $\bar{b} \to a^n b^m, \bar{c} \to b^{n-m+1}c^{n+1} \in P$. As before, there must be also productions $d_1 \to b^i, d_2 \to c^j \in P$, $i, j \geq 1$, $d_1, d_2 \in \{b, c\}$. Since $u_b \neq \varepsilon$ and $u_c \neq \varepsilon$, $a\ {}_p\!\Rightarrow^* x$, where $x \notin X$, which is a contradiction.

Therefore, $a^{l+1}b^l c^l \notin L(H, {}_p\!\Rightarrow)$, which implies that $X \notin \mathbf{PPCF}$. □

Theorem 3.28 (Subfamily S).

$$\mathbf{JPPCF} - (\mathbf{CF} \cup \mathbf{PPCF} \cup \mathbf{JSPCF}) \neq \emptyset$$

Proof. Let $\Sigma = \{a, b, c, \hat{a}, \hat{b}, \hat{c}\}$ be an alphabet and let

$$X = \{\hat{a}\hat{b}\hat{c}\} \cup \{x \mid \#_a(x) - 1 = \#_b(x) = \#_c(x), \\ x \in \{a, b, c\}^+\}$$

be a language over Σ. Following the pumping lemma for context-free languages, $X \notin \mathbf{CF}$. Since there is a PCFG $G = (\Sigma, \{\hat{a} \to a, \hat{b} \to \varepsilon, \hat{c} \to \varepsilon, a \to abca, a \to a, b \to b, c \to c\}, \hat{a}\hat{b}\hat{c})$ such that $L(G, {}_{jp}\!\Rightarrow) = X$, $X \in \mathbf{JPPCF}$. By contradiction, we show that $X \notin \mathbf{JSPCF}$ and $X \notin \mathbf{PPCF}$.

Suppose that $X \in \mathbf{JSPCF}$. Then, there is a PCFG $H = (\Sigma, P, \sigma)$ such that $L(H, {}_j\!\Rightarrow) = X$. First, we choose σ. From the definition of X, $a \in X$ and for every string $x \in X - \{a\}$ holds $|x| \geq 3$. Since we are able to erase only one symbol during direct derivation step by ${}_j\!\Rightarrow$ and there is no string of length 2 contained in X, we must choose $\sigma = a$ as the axiom. Because $abca \in X$ and $\hat{a}\hat{b}\hat{c} \in X$, there must be two derivations, $a\ {}_j\!\Rightarrow^* abca$ and $a\ {}_j\!\Rightarrow^* \hat{a}\hat{b}\hat{c}$, and this implies that there exists also a derivation $a\ {}_j\!\Rightarrow^* \hat{a}\hat{b}\hat{c}bca$. Since $\hat{a}\hat{b}\hat{c}bca \notin X$, we have a contradiction.

Next, suppose that $X \in \mathbf{PPCF}$, so there exists a PCFG $H' = (\Sigma, P', \sigma')$ such that $L(H', {}_p\!\Rightarrow) = X$. In this case, we must choose $\sigma' = \hat{a}\hat{b}\hat{c}$ as the axiom. If we choose a, then $a\ {}_p\!\Rightarrow^* abca$ and $a\ {}_p\!\Rightarrow^* \hat{a}\hat{b}\hat{c}$ implies that $a\ {}_p\!\Rightarrow^* u_1 a u_2 \hat{a} u_3$, $u_1, u_2, u_3 \in \Sigma^*$, and $u_1 a u_2 \hat{a} u_3 \notin X$. If we choose $abca$ or similar, then $abca\ {}_p\!\Rightarrow^* a$ implies that $a\ {}_p\!\Rightarrow^* \varepsilon$, and $\varepsilon \notin X$. Without loss of generality, assume that for every $\alpha \to \beta \in P'$, $\beta \in \{a, b, c\}^*$ (this can be assumed since $\hat{a}\hat{b}\hat{c}$ is the only string over $\{\hat{a}, \hat{b}, \hat{c}\}$ in X). As $a \in X$, $a \to \varepsilon$, $a \to b$, and $a \to c$ are not contained in P'. The observations (1) to (3) from the proof of Theorem 3.27 hold also for H'. The rest of proof is similar to the proof of Theorem 3.27. □

Theorem 3.29 (Subfamily T).

$$\mathbf{CS} - (\mathbf{CF} \cup \mathbf{JSPCF} \cup \mathbf{PPCF} \cup \mathbf{JPPCF}) \neq \emptyset$$

Proof. Let $X = \{a^p \mid p \text{ is a prime}\}$ be a language over unary alphabet $\{a\}$. $X \in \mathbf{CS}$ and $X \notin \mathbf{CF}$ are a well-known containments (see [7]). By Lemma 3.4 and Corollary 3.3, $X \notin \mathbf{JPPCF}$ and $X \notin \mathbf{JSPCF}$. As for unary languages $\mathbf{PPCF} = \mathbf{JPPCF}$, $X \notin \mathbf{PPCF}$. □

The summary of theorems 3.10 through 3.29 is visualized in Figure 1.

Absence of erasing productions

As stated in Lemma 3.1, it is natural that the family of languages generated by pure grammars without erasing productions is included in the family of languages generated by pure grammars in which the presence of erasing productions is allowed. As we show further, for PCFG, the inclusions stated in Lemma 3.1 are proper. The PG case is left as an open problem.

Theorem 3.30. *Let*

$$X \in \{\mathbf{SPCF}, \mathbf{JSPCF}, \mathbf{PPCF}, \mathbf{JPPCF}, \mathbf{0L}\}.$$

Then, $X^{-\varepsilon} \subset X$.

Proof. Let $K = \{a, ab\}$ and $Y = \{aa, aab\}$ be two languages over $\Sigma = \{a, b\}$. Furthermore, let $G = (\Sigma, \{a \to a, b \to \varepsilon\}, ab)$ and $G' = (\Sigma, \{a \to a, b \to \varepsilon\}, aab)$ be two PCFGs.

a) $\mathbf{SPCF}^{-\varepsilon} \subset \mathbf{SPCF}$. Since $K = L(G, {}_s\!\Rightarrow)$, $K \in \mathbf{SPCF}$. Assume that $K \in \mathbf{SPCF}^{-\varepsilon}$ and then there is a PCFG $H = (\Sigma, P, \sigma)$ with no erasing productions in P such that $L(H, {}_s\!\Rightarrow) = K$. Obviously, $\sigma = a$ and then $a \to ab \in P$. We have $a\ {}_s\!\Rightarrow^* abb$ and since $abb \notin K$, $K \notin \mathbf{SPCF}^{-\varepsilon}$.

b) $\mathbf{JSPCF}^{-\varepsilon} \subset \mathbf{JSPCF}$. $K \in \mathbf{JSPCF}$ and $K \notin \mathbf{JSPCF}^{-\varepsilon}$ are proved analogously as in a).

c) $\mathbf{PPCF}^{-\varepsilon} \subset \mathbf{PPCF}$. Since $Y = L(G', {}_p\!\Rightarrow)$, $Y \in \mathbf{PPCF}$. Assume that $Y \in \mathbf{PPCF}^{-\varepsilon}$ and then there is a PCFG $H = (\Sigma, P, \sigma)$ with no erasing productions in P such that $L(H, {}_p\!\Rightarrow) = Y$. Obviously, $\sigma = aa$ and then $a \to ab \in P$. We have $aa\ {}_p\!\Rightarrow^* abab$ and since $abab \notin Y$, $Y \notin \mathbf{PPCF}^{-\varepsilon}$.

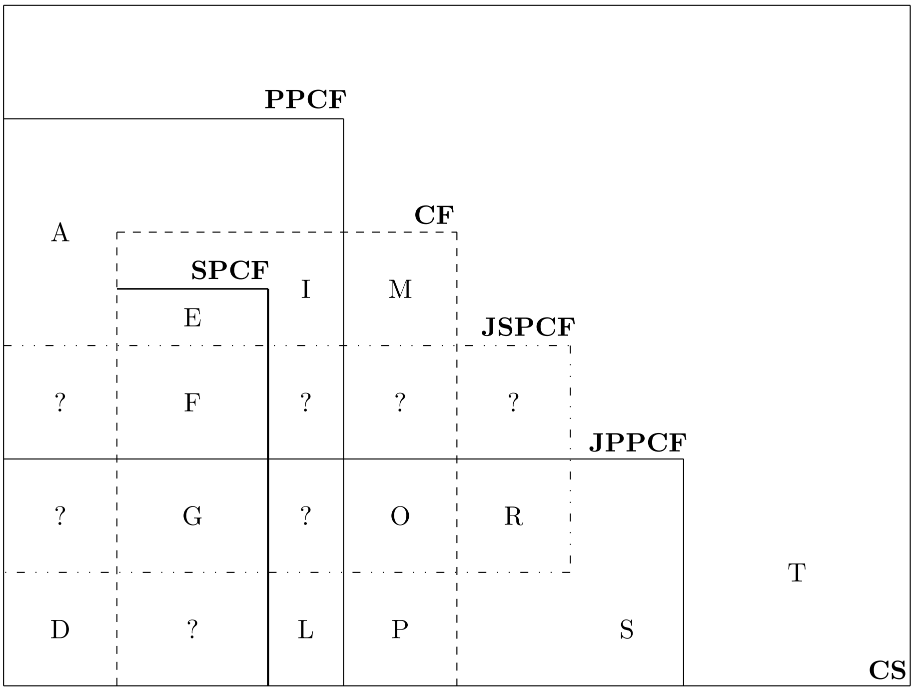
d) $\mathbf{JPPCF}^{-\varepsilon} \subset \mathbf{JPPCF}$. $Y \in \mathbf{JPPCF}$ and $Y \notin \mathbf{JPPCF}^{-\varepsilon}$ are proved analogously as in c).

e) $\mathbf{0L}^{-\varepsilon} \subset \mathbf{0L}$ follows from c).

□

Open Problem 3.31. Let $X \in \{\mathbf{SP}, \mathbf{JSP}, \mathbf{PP}, \mathbf{JPP}\}$. Is the inclusion $X^{-\varepsilon} \subseteq X$, in fact, proper?

From Figure 1 and from mentioned theorems, we are able to find out the most of relations between investigated language families (even for those which are generated by PCFGs without erasing productions—the most of languages used in Figure 1 have this property), but not all. Following theorems fill this gap.



$\mathrm{A} = \{a^{2^n} b^{2^n} \mid n \geq 0\}$

$\mathrm{D} = \{a^{2^n} \mid n \geq 0\}$

$\mathrm{E} = \{a^n c b^n \mid n \geq 0\}$

$\mathrm{F} = \{aa, aab, aac, aabc\}$

$\mathrm{G} = \{a\}^+$

$\mathrm{I} = \{aabb, ccdd\}$

$\mathrm{L} = \{ab, cd, dc\}$

$\mathrm{M} = \{a^n b^n \mid n \geq 1\}$

$\mathrm{O} = \{aabb, abab, abba, baab, baba, bbaa\}$

$\mathrm{P} = \{aabb, ccdd, cdcd, cddc, dccd, dcdc, ddcc\}$

$\mathrm{R} = \left\{ w \;\middle|\; \begin{array}{l} \#_a(w) - 1 = \#_b(w) = \#_c(w), \\ w \in \{a, b, c\}^+ \end{array} \right\}$

$\mathrm{S} = \{\hat{a}\hat{b}\hat{c}\} \cup \mathrm{R}$

$\mathrm{T} = \{a^p \mid p \text{ is a prime}\}$

FIGURE 1. Summary of hierarchy between **SPCF**, **JSPCF**, **PPCF**, **JPPCF**, **CF**, and **CS** language families (? stands for an open problem).

THEOREM 3.32. **SPCF** *and* $\mathbf{PPCF}^{-\varepsilon}$ *are incomparable, but not disjoint.*

Proof. Let $X = \{aa, aab\}$ be a language over alphabet $\Sigma = \{a, b\}$. Obviously, there is a PCFG $G = (\Sigma, \{a \to a, b \to \varepsilon\}, aab)$ such that $L(G, {}_s\!\Rightarrow) = X$, and then $X \in \mathbf{SPCF}$. By Theorem 3.30, $X \notin \mathbf{PPCF}^{-\varepsilon}$. Conversely, there is a language $Y = \{a^{2^n} \mid n \geq 0\}$ over $\{a\}$ such that $Y \notin \mathbf{SPCF}$ and $Y \in \mathbf{PPCF}^{-\varepsilon}$ (see D in Figure 1 and observe that to get Y we need no erasing productions). Finally, $\{a\}^+ \in \mathbf{SPCF} \cap \mathbf{PPCF}^{-\varepsilon}$. □

THEOREM 3.33. **SPCF** *and* $\mathbf{0L}^{-\varepsilon}$ *are incomparable, but not disjoint.*

Proof. Analogous to the proof of Theorem 3.32. □

The mutual relation between $\mathbf{JSPCF}^{-\varepsilon}$ and $\mathbf{JPPCF}^{-\varepsilon}$ is either incomparability or $\mathbf{JSPCF}^{-\varepsilon} \subset \mathbf{JPPCF}^{-\varepsilon}$, but we do not know the answer now. We also do not know either if $\mathbf{JSPCF}^{-\varepsilon}$ and **JPPCF** are incomparable or $\mathbf{JSPCF}^{-\varepsilon} \subset \mathbf{JPPCF}$.

OPEN PROBLEM 3.34. What is the relation between $\mathbf{JSPCF}^{-\varepsilon}$ and $\mathbf{JPPCF}^{-\varepsilon}$?

OPEN PROBLEM 3.35. What is the relation between $\mathbf{JSPCF}^{-\varepsilon}$ and **JPPCF**?

THEOREM 3.36. $\mathbf{PPCF}^{-\varepsilon}$ *and* **0L** *are incomparable, but not disjoint.*

Proof. Let $X = \{aa, aab\}$ and $Y = \{a, aab\}$ be two languages over $\{a, b\}$. $X \notin \mathbf{PPCF}^{-\varepsilon}$, $X \in \mathbf{0L}$, $Y \in \mathbf{PPCF}^{-\varepsilon}$, and $Y \notin \mathbf{0L}$ proves the incomparability, while $\{a\}^+ \in \mathbf{PPCF}^{-\varepsilon} \cap \mathbf{0L}$ proves the disjointness. □

Remark on unary alphabets

We close this section by showing how the mutual relations between investigated language families change if we consider only alphabets containing only one symbol. From Theorem 3.2, Theorem 3.6, and Theorem 3.7, we can conclude that for every unary

TABLE 1. Mutual relations between investigated language families. A denotes the language family from the first column, B the language family from the table header. If the relation in the cell given by A and B is $\star$, then $A \star B$. $A \| B$ means that A and B are incomparable, but not disjoint, ? stands for an open problem, and the meaning of $\subset$, $=$, and $\subset$ is as usual.

A \ B	**SPCF**	**SPCF**$^{-\varepsilon}$	**JSPCF**	**JSPCF**$^{-\varepsilon}$	**PPCF**	**PPCF**$^{-\varepsilon}$	**JPPCF**	**JPPCF**$^{-\varepsilon}$	**0L**	**0L**$^{-\varepsilon}$
SPCF	$=$	$\supset$	$\parallel$	$\parallel$	$\subset$	$\parallel$	$\parallel$	$\parallel$	$\subset$	$\parallel$
SPCF$^{-\varepsilon}$	$\subset$	$=$	$\parallel$	$\parallel$	$\subset$	$\subset$	$\parallel$	$\parallel$	$\subset$	$\subset$
JSPCF	$\parallel$	$\parallel$	$=$	$\supset$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$	$\parallel$
JSPCF$^{-\varepsilon}$	$\parallel$	$\parallel$	$\subset$	$=$	$\parallel$	$\parallel$	?	?	$\parallel$	$\parallel$
PPCF	$\supset$	$\supset$	$\parallel$	$\parallel$	$=$	$\supset$	$\parallel$	$\parallel$	$\supset$	$\supset$
PPCF$^{-\varepsilon}$	$\parallel$	$\supset$	$\parallel$	$\parallel$	$\subset$	$=$	$\parallel$	$\parallel$	$\parallel$	$\supset$
JPPCF	$\parallel$	$\parallel$	$\parallel$	?	$\parallel$	$\parallel$	$=$	$\supset$	$\parallel$	$\parallel$
JPPCF$^{-\varepsilon}$	$\parallel$	$\parallel$	$\parallel$	?	$\parallel$	$\parallel$	$\subset$	$=$	$\parallel$	$\parallel$
0L	$\supset$	$\supset$	$\parallel$	$\parallel$	$\subset$	$\parallel$	$\parallel$	$\parallel$	$=$	$\supset$
0L$^{-\varepsilon}$	$\parallel$	$\supset$	$\parallel$	$\parallel$	$\subset$	$\subset$	$\parallel$	$\parallel$	$\subset$	$=$

alphabet

$$\mathbf{SPCF} = \mathbf{JSPCF} \subset \mathbf{PPCF} = \mathbf{JPPCF} = \mathbf{0L}.$$

Trivially,

$$\mathbf{SPCF}^{-\varepsilon} = \mathbf{JSPCF}^{-\varepsilon} \subset \mathbf{PPCF}^{-\varepsilon} = \mathbf{JPPCF}^{-\varepsilon} = \mathbf{0L}^{-\varepsilon}.$$

As the following theorem demonstrates that **PPCF**$^{-\varepsilon}$ and **SPCF** are incomparable, but not disjoint, we can summarize the results for the unary alphabet by Figure 2.

THEOREM 3.37. *In the case of unary alphabets,* **SPCF** *and* **PPCF**$^{-\varepsilon}$ *are incomparable, but not disjoint.*

Proof. Clearly, the language $\{a\}^+$ is contained in both **SPCF** and **PPCF**$^{-\varepsilon}$. Since the language $\{\varepsilon, a\}$ from **SPCF** is not contained in **PPCF**$^{-\varepsilon}$, **SPCF** $\not\subseteq$ **PPCF**$^{-\varepsilon}$. Conversely, **PPCF**$^{-\varepsilon}$ $\not\subseteq$ **SPCF** since **PPCF**$^{-\varepsilon}$ is not semilinear. $\square$

$$\begin{tikzcd}
\mathbf{PPCF}^{-\varepsilon} \arrow[r, dash] \arrow[d] & \mathbf{JPPCF}^{-\varepsilon} \arrow[r, dash] \arrow[d] & \mathbf{0L}^{-\varepsilon} \arrow[d] \\
\mathbf{PPCF} \arrow[r, dash] & \mathbf{JPPCF} \arrow[r, dash] & \mathbf{0L} \\
\mathbf{SPCF} \arrow[r, dash] \arrow[u] & \mathbf{JSPCF} \arrow[u] & \\
\mathbf{SPCF}^{-\varepsilon} \arrow[r, dash] \arrow[u] & \mathbf{JSPCF}^{-\varepsilon} \arrow[u] &
\end{tikzcd}$$

FIGURE 2. A mutual relations between investigated language families in the case of unary alphabets. The straight line between two families means that these families are identical. The arrow from family A to family B denotes that $A \subset B$.

4. CONCLUSION

Consider **SPCF**, **JSPCF**, **PPCF**, **JPPCF**, **0L**, **SPCF**$^{-\varepsilon}$, **JSPCF**$^{-\varepsilon}$, **PPCF**$^{-\varepsilon}$, **JPPCF**$^{-\varepsilon}$, and **0L**$^{-\varepsilon}$ (see Section 2). The present paper has investigated mutual relations between these language families, which are summarized in Table 1 and Figure 1. As a special case, this paper has also performed an analogical study in terms of unary alphabets (see Figure 2).

Although we have already pointed out several open problems earlier in the paper (see Open Problems 3.11, 3.12, 3.17, 3.19, 3.20, 3.23, 3.26, 3.31, 3.34, and 3.35), we repeat the questions of a particular significance next.

- Is it true that (**PPCF** $\cap$ **JSPCF**) $-$ (**CF** $\cup$ **JPPCF**) $\neq \emptyset$?
- Is it true that (**PPCF** $\cap$ **JSPCF** $\cap$ **JPPCF**) $-$ **CF** $\neq \emptyset$?
- Is it true that (**SPCF** $\cap$ **JPPCF**) $-$ **JSPCF** $\neq \emptyset$?
- Is it true that (**PPCF** $\cap$ **CF** $\cap$ **JSPCF**) $-$ (**SPCF** $\cup$ **JPPCF**) $\neq \emptyset$?
- Is it true that (**PPCF** $\cap$ **CF** $\cap$ **JSPCF** $\cap$ **JPPCF**) $-$ **SPCF** $\neq \emptyset$?
- Is it true that (**CF** $\cap$ **JSPCF**) $-$ (**PPCF** $\cup$ **JPPCF**) $\neq \emptyset$?
- Is it true that **JSPCF** $-$ (**CF** $\cup$ **PPCF** $\cup$ **JPPCF**) $\neq \emptyset$?
- Let $X \in \{\mathbf{SP}, \mathbf{JSP}, \mathbf{PP}, \mathbf{JPP}\}$. Is the inclusion $X^{-\varepsilon} \subseteq X$, in fact, proper?
- What is the relation between **JSPCF**$^{-\varepsilon}$ and **JPPCF**$^{-\varepsilon}$?

- What is the relation between $\mathbf{JSPCF}^{-\varepsilon}$ and **JPPCF**?

Recall that the present study has only considered pure grammars based upon context-free productions. Of course, from a broader perspective, we might reconsider all the study in terms of grammars that allow non-context-free productions as well.

ACKNOWLEDGEMENTS

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II), project IT4Innovations excellence in science – LQ1602.

REFERENCES

[1] H. Bordihn, H. Fernau, and M. Holzer, *Accepting pure grammars*, Publ. Math. 60 (2002), pp. 483–510.

[2] H. Chigahara, S.Z. Fazekas, and A. Yamamura, *One-way jumping finite automata*, Int. J. of Found. of Comput. Sci. 27 (2016), pp. 391–405.

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

[4] H. Fernau, M. Paramasivan, and M.L. Schmid, *Jumping finite automata: Characterizations and complexity*, in *Implementation and Application of Automata - 20th International Conference, CIAA*, F. Drewes, ed., LNCS Vol. 9223, Springer, 2015, pp. 89–101.

[5] H. Fernau, M. Paramasivan, M.L. Schmid, and V. Vorel, *Characterization and complexity results on jumping finite automata*, Theoret. Comput. Sci. (2016), in press.

[6] A. Gabrielian, *Pure grammars and pure languages*, Rep. C.S.R.R. 2027, Department of Computer Research, University of Waterloo, Waterloo, Ontario, Canada, 1970.

[7] J.E. Hopcroft, R. Motwani, and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., Addison-Wesley, 2001.

[8] R. Kocman, Z. Křivka, and A. Meduna, *On double-jumping finite automata and their closure properties*, Theor. Inf. Appl. (2017), in press.

[9] R. Kocman and A. Meduna, *On parallel versions of jumping finite automata*, in *Proceedings of the 2015 Federated Conference on Software Development and Object Technologies*, Advances in Intelligent Systems and Computing 511 (2016), Springer International Publishing, pp. 142–149.

[10] Z. Křivka and A. Meduna, *Jumping grammars*, Int. J. of Found. of Comput. Sci. 26 (2015), pp. 709–731.

[11] Z. Křivka, C. Martín-Vide, A. Meduna, and K.G. Subramanian, *A variant of pure two-dimensional context-free grammars generating picture languages*, in *Proceedings of Combinatorial Image Analysis - 16th International Workshop (IWCIA 2014)*, Brno, Czech Republic, 28–30 May 2014, pp. 123–133.

[12] M. Langer and A. Kelemenová, *Positioned agents in eco-grammar systems with border markers and pure regulated grammars*, Kybernetika 48 (2012), pp. 502–517.

[13] H.A. Maurer, A. Salomaa, and D. Wood, *Pure grammars*, Inf. and Control 44 (1980), pp. 47–72.

[14] A. Meduna, *Automata and Languages: Theory and Applications*, Springer, London, 2000.

[15] A. Meduna and O. Soukup, *Jumping scattered context grammars*, Fund. Inform. (2017), in press.

[16] A. Meduna and P. Zemek, *Jumping finite automata*, Int. J. of Found. of Comput. Sci. 23 (2012), pp. 1555–1578.

[17] B. Nagy, *A class of 2-head finite automata for linear languages*, Triangle 8 (Languages: Mathematical Approaches) (2012), pp. 89–99.

[18] M. Novotný, *Construction of pure grammars*, Fundam. Inform. 52 (2002), pp. 345–360.

[19] R.J. Parikh, *On context-free languages*, J. of the Assoc. for Comput. Mach. 13 (1966), pp. 570–581.

[20] G. Rozenberg and P.G. Doucet, *On 0L-languages*, Inf. and Control 19 (1971), pp. 302–318.

[21] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages, Vol. 1: Word, Language, Grammar*, Springer, New York, 1997.

[22] A. Salomaa, *Formal Languages*, Academic Press, London, 1973.