Computer Science Journal of Moldova, vol.**27**, no.3(81), 2019

On Operations over Language Families

Alexander Meduna, Radim Krčmář,
Adam Kövári, Zuzana Beníčková

Abstract

Let O and $\mathbf{F}$ be an operation and a language family, respectively. So far, in terms of closure properties, the classical language theory has only investigated whether $O(\mathbf{F}) \subseteq \mathbf{F}$, where $O(\mathbf{F})$ is the family resulting from O applied to all members of $\mathbf{F}$. If $O(\mathbf{F}) \subseteq \mathbf{F}$, $\mathbf{F}$ is closed under O; otherwise, it is not.

This paper proposes a finer and wider approach to this investigation. Indeed, it studies almost all possible set-based relations between $\mathbf{F}$ and $O(\mathbf{F})$, including $O(\mathbf{F}) = \emptyset$; $F \not\subset O(\mathbf{F})$, $O(\mathbf{F}) \not\subset \mathbf{F}$, $\mathbf{F} \cap O(\mathbf{F}) \neq \emptyset$; $\mathbf{F} \cap O(\mathbf{F}) = \emptyset$, $O(\mathbf{F}) \neq \emptyset$; $O(\mathbf{F}) = \mathbf{F}$; and $\mathbf{F} \subset O(\mathbf{F})$. Many operations are studied in this way. A sketch of application perspectives and open problems closes the paper.

Keywords: language operations; language families; closure properties; finer approach; new trend; set theory.

MSC 2010: 68Q45, 03D05.

1 Introduction

Over its entire history, formal language theory has primarily studied operations over language families in terms of closure properties by analogy with the investigation of these properties in discrete mathematics as a whole. To give an insight into this study, consider a languare family $\mathbf{F}$, a language operation O, and $O(\mathbf{F})$ as the language family resulting from the application of O to all languages in $\mathbf{F}$. In essence, so far, the language theory has restricted its attention only to the study whether or not $O(\mathbf{F}) \subseteq \mathbf{F}$. If so, $\mathbf{F}$ is closed with respect to O; otherwise, it is not.

The authors believe that formal language theory can approach the investigation of operations over language families in a much finer way than it has been done so far. In fact, there already exist many results that can be combined together to express some properties related to closure or non-closure results more precisely; unfortunately, formal language theory lacks a proper universal terminology or framework, which would allow it to express results of this kind in a uniform and general way. To illustrate this finer approach by a simple example, recall that the family of context-sensitive languages is not closed under homomorphism (see the Corollary on p. 279 in [8]). Apart from this non-closure result, it is well known that every recursively enumerable language L coincides with h(K), where h is a homomorphism and K is a context-sensitive language. Of course, if a language is not recursively enumerable, it cannot be expressed in this way. Putting these results together, we can naturally say that homomorphism expands the family of context-sensitive languages onto that of recursively enumerable languages. Therefore, the present paper proposes a new terminology for results like this and illustrates it by many examples, observations and results.

More specifically, the present paper introduces these notions— (1) if $\mathbf{F} - O(\mathbf{F}) \neq \emptyset$, O reduces $\mathbf{F}$; (2) if $O(\mathbf{F}) = \emptyset$, O eliminates $\mathbf{F}$; (3) if $O(\mathbf{F}) \subset \mathbf{F}$, O properly reduces $\mathbf{F}$; (4) if $\mathbf{F} \not\subset O(\mathbf{F}), O(\mathbf{F}) \not\subset \mathbf{F}$, and $\mathbf{F} \cap O(\mathbf{F}) \neq \emptyset$, then O incomparably reduces $\mathbf{F}$; (5) if $\mathbf{F} - O(\mathbf{F}) = \mathbf{F}$ and $O(\mathbf{F}) \neq \emptyset$, O expels $\mathbf{F}$; (6) if $O(\mathbf{F}) = \mathbf{F}$, O unchanges $\mathbf{F}$; and (7) if $\mathbf{F} \subset O(\mathbf{F})$, O expands $\mathbf{F}$. In terms of these notions, the paper discusses a broad variety of operations, ranging from classical operations, such as complement, up to newly introduced operations. It starts from utterly straightforward observations about simple operations and gradually proceeds towards more complicated operations and results concerning them. Sometimes, it applies these operations to well-known language families, such as the family of linear languages. Most often, however, as the main direction of this newly proposed investigation trend, the paper establishes general results concerning language families satisfying some prescribed properties. In its conclusion, in a greater detail, the paper suggests several special branches of study within this newly suggested trend as well as application perspectives.

2 Preliminaries

This paper assumes that the reader is familiar with discrete mathematics (see [4]). Most importantly, it assumes an in-depth knowledge of the language theory (see [2], [3], [5]). Let X and Y be two sets. X and Y are comparable if $X \subseteq Y$ or $Y \subseteq X$; otherwise, X and Y are incomparable. In other words, X and Y are incomparable if and only if $X \nsubseteq Y$ and $Y \nsubseteq X$ (notice that if X and Y are disjoint, then they are necessarily incomparable). For a set Q, $power(Q)$ denotes the power set of Q, and $card(Q)$ denotes its cardinality. For a total function f over Q, f^i denotes its ith power, for $i \geq 0$. $\mathbb{N}$ denotes the set of natural numbers.

For an alphabet V, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$; algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. Any member $w \in V^*$ is called a *word*. For any $w \in V^*, |w|$ and $reversal(w)$ denote the length of w and the reversal of w, respectively. For any $N \subseteq V, occur(w, N)$ denotes the number of symbols from N occurring in w. For every $i \in \{0, 1, \ldots, |w|\}$, $suffix(w, i)$ denotes the suffix of w of length i; analogously, $prefix(w, i)$ denotes the prefix of w of which length is i. A language L is any subset of V^*, $L \subseteq V^*$. Let **ALL** denote the set of all languages; in other words, **ALL** represents the *universal set* of languages throughout this paper. A *language family* $\mathbf{F}$ is any subset of **ALL**, $\mathbf{F} \subseteq \mathbf{ALL}$; notice that $\mathbf{F} \subseteq \mathbf{ALL}$ is synonymous with $\mathbf{F} \in power(\mathbf{ALL})$. Observe that both $\emptyset$ and $\{\emptyset\}$ are language families, but $\emptyset \neq \{\emptyset\}$; indeed, $card(\emptyset) = 0$ while $\text{card}(\{\emptyset\}) = 1$. Set $alph(L) = \{a \mid a \text{ occurs in a word in } L\}$, and $alph(\mathbf{F}) = \{a \mid a \in alph(L), L \in \mathbf{F}\}$.

A *grammar* is a quadruple, $G = (N, T, P, S)$, where N and T are nonterminal and terminal alphabets, respectively; $N \cap T = \emptyset$. N contains S—the start symbol of G. P is a finite non-empty set of productions of the form $x \to y$, where $x, y \in (N \cup T)^*$ so $N \cap alph(x) \neq \emptyset$. For every $p \in P$ of the form $x \to y$, x is the *left-hand side* of p, $lhs(p)$, and y is the *right-hand side* of p, $rhs(p)$. To express that $card(N) = n$, where $n \in \mathbb{N}$, we write ${}_nG$. If $x \to y \in P$, $v = uxz$, $w = uyz$ with $u, z \in (N \cup T)^*$, then v *directly derives* w in G, symbolically written as $v \Rightarrow w$ in G. In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The *language of* G, $L(G)$, is defined as $L(G) = \{w \in T^* : S \Rightarrow^* w\}$, and any derivation of the form $S \Rightarrow^* w$ in G with $w \in T^*$ is called a *successful derivation*.

Let $G = (N, T, P, S)$ be a grammar. G is referred to as *context-sensitive* if every production in P is of the form $uAv \to uyv$ with $A \in N$, $u, v \in (N \cup T)^*$, $y \in (N \cup T)^+$. G is referred to as *context-free* if every $x \to y \in P$ satisfies $x \in N, y \in (N \cup T)^*$. A context-free grammar is in the *Chomsky normal form* if every $x \to y \in P$ satisfies $y \in NN \cup T$. Let $j \in \mathbb{N}$; a context-free grammar G is referred to as j-linear if for any $x \to y \in P$, $S \notin alph(y)$ and either $x = S$ and $occur(y, N) \leq j$ or $x \neq S$ and $occur(y, N) \leq 1$. Instead of a 1-linear grammar, we often simply say a *linear grammar*. For $j \in \mathbb{N}$; ${}_j\mathbf{LIN}$ denotes the language family generated by j-linear grammars; recall that for every $k \in \mathbb{N}$; ${}_k\mathbf{LIN} \subset {}_{k+1}\mathbf{LIN}$ (see Theorem 8.3 in [5]).

REG, **LIN**, **CF**, **CS**, **DEC**, and **RE** denote the families of regular, linear, context-free, context-sensitive, decidable, and recursively enumerable languages, respectively. Recall (see [2])

$$\mathbf{REG} \subset \mathbf{LIN} \subset \mathbf{CF} \subset \mathbf{CS} \subset \mathbf{DEC} \subset \mathbf{RE} \subset \mathbf{ALL}$$

3 Definitions

Let $n \in \mathbb{N}$. An n-ary *language operation* on $\mathbf{ALL}$ is a total function O from the n-ary Cartesian product $\mathbf{ALL}^n$ into $\mathbf{ALL}$. An n-ary language-family operation over $power(\mathbf{ALL})$ is a total function O from the n-ary Cartesian product $(power(\mathbf{ALL}))^n$ into $power(\mathbf{ALL})$.

Throughout this paper, we discuss only unary and binary language operations, and we only consider unary language-family operations. For any $\mathbf{F} \subseteq \mathbf{ALL}$, $O(\mathbf{F})$ denotes the image of $\mathbf{F}$ over O, so $O(\mathbf{F}) = \{O(L) \mid L \in \mathbf{F}\}$. In what follows, we automatically assume that $\mathbf{F} \neq \emptyset$

(of course, $O(\mathbf{F}) = \emptyset$ is not ruled out). Notice that in the next main definition of this paper, 2 through 5 are, in effect, special cases of 1.

Definition 3.1. Let $\mathbf{F} \subseteq \mathbf{ALL}$.

1. If $\mathbf{F} - O(\mathbf{F}) \neq \emptyset$, O reduces $\mathbf{F}$.
2. If $O(\mathbf{F}) = \emptyset$, O eliminates $\mathbf{F}$.
3. If $O(\mathbf{F}) \subset \mathbf{F}$, O properly reduces $\mathbf{F}$.
4. If $\mathbf{F} \not\subset O(\mathbf{F}), O(\mathbf{F}) \not\subset \mathbf{F}$, and $\mathbf{F} \cap O(\mathbf{F}) \neq \emptyset$, then O incomparably reduces $\mathbf{F}$.
5. If $\mathbf{F} - O(\mathbf{F}) = \mathbf{F}$ and $O(\mathbf{F}) \neq \emptyset$, O expels $\mathbf{F}$.
6. If $O(\mathbf{F}) = \mathbf{F}$, O unchanges $\mathbf{F}$.
7. If $\mathbf{F} \subset O(\mathbf{F})$, O expands $\mathbf{F}$.

Suppose that O expands $\mathbf{F}$ and $O(\mathbf{F}) = \mathbf{W}$, where $\mathbf{W}$ is a well-known language family, such as any of the families listed in the conclusion of Section 2. Under these circumstances, we sometimes explicitly point out that O expands $\mathbf{F}$ onto $\mathbf{W}$ (see, for instance, Observation 4.13 and Theorem 4.28). We often make analogical statements in terms of the other parts of Definition 3.1 unless a confusion arises (see, for instance, Observation 4.6).

4 Results

Simply put, the present section illustrates Definition 3.1 by a large variety of language-family operations. Starting from part 1, it proceeds, in essence, towards part 7 of the definition. Consequently, it begins with a variety of reducing operations and ends with expanding operations. As far as the mathematical level is concerned, the section opens its discussion with utterly trivial operations and closes it with more complicated operations and results about them.

4.1 Reducing Operations

For all $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, define operations $\mathit{EmptySetConcatenation}(\mathbf{F}) = \{L\emptyset \mid L \in \mathbf{F}\}$, and $\mathit{EmptyStringConcatenation}(\mathbf{F}) = \{L\{\varepsilon\} \mid L \in \mathbf{F}\}$. The next observation is obvious.

Observation 4.1. *For all* $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, $\mathit{EmptySetConcatenation}(\mathbf{F}) = \emptyset$ *and* $\mathit{EmptyStringConcatenation}(\mathbf{F}) = \mathbf{F}$. *In words, the former eliminates* $\mathbf{F}$ *while the latter unchanges* $\mathbf{F}$.

Let Complement denote the well-known unary language operation of complement. For all $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, define $\mathit{Complement}(\mathbf{F}) = \{\mathit{Complement}(L) \mid L \in \mathbf{F}\}$.

Observation 4.2. $\mathit{Complement}(\mathbf{RE}) = \mathbf{DEC} \cup \mathit{Complement}(\mathbf{RE} - \mathbf{DEC})$, *where* $\mathit{Complement}(\mathbf{RE} - \mathbf{DEC}) \subset \mathbf{ALL} - \mathbf{RE}$. *Therefore, Complement incomparably reduces* $\mathbf{RE}$.

Proof. Let $L \in \mathbf{RE}$. If $L \in \mathbf{DEC}$, then $\mathit{Complement}(L) \in \mathbf{DEC}$ (see Theorem 18.3 in [6]). If $L \in \mathbf{RE} - \mathbf{DEC}$, $\mathit{Complement}(L) \notin \mathbf{RE}$ because $L \in \mathbf{DEC}$ if and only if $L \in \mathbf{RE}$ and $\mathit{Complement}(L) \in \mathbf{RE}$ (see Theorem 4.22 in [7]). Thus, $\mathit{Complement}(\mathbf{RE}) = \mathbf{DEC} \cup \mathit{Complement}(\mathbf{RE} - \mathbf{DEC})$ with $\mathit{Complement}(\mathbf{RE} - \mathbf{DEC})$ out of $\mathbf{RE}$, so $\mathit{Complement}(\mathbf{RE} - \mathbf{DEC}) \subset \mathbf{ALL} - \mathbf{RE}$. The rest of this observation follows from part 4 of Definition 3.1. □

Let $a \in \mathit{alph}(\mathbf{ALL})$. For all $L \in \mathbf{ALL}$, define operation $a\text{-}\mathit{End}(L) = L\{a\}$, and for all $\mathbf{F} \subseteq \mathbf{ALL}$, $a\text{-}\mathit{End}(\mathbf{F}) = \{a\text{-}\mathit{End}(L) \mid L \in \mathbf{F}\}$. As usual, $a\text{-}\mathit{End}^i$ denotes the ith power of $a\text{-}\mathit{End}$, $i \geq 1$. Consider these two disjoint language families

$$\mathbf{ODD}_a = \{L \mid L \subseteq \{a\}^*, |x| \text{ is odd for all } x \in L\}, \text{ and}$$

$$\mathbf{EVEN}_a = \{L \mid L \subseteq \{a\}^*, |x| \text{ is even for all } x \in L\}.$$

Observation 4.3. *Let* $i \geq 1$. *Then:*

a-End$^i(\mathbf{ODD}_a) = \mathbf{EVEN}_a$ *if* i *is odd, and*

a-End$^i(\mathbf{EVEN}_a) = \mathbf{ODD}_a$ *if* i *is even.*

Proof. Clear. $\square$

Corollary 4.4. *a-End*$(\mathbf{ODD}_a) = \mathbf{EVEN}_a$ *and a-End*$(\mathbf{EVEN}_a) = \mathbf{ODD}_a$. *In words, a-End expels* $\mathbf{ODD}_a$ *onto* $\mathbf{EVEN}_a$, *and it expels* $\mathbf{EVEN}_a$ *onto* $\mathbf{ODD}_a$.

Proof. Consider Observation 4.3 for $i = 1$ to see that this corollary holds true. $\square$

Define the homomorphism *a-Coding* as *a-Coding*$(b) = a$ for all $b \in alph(\mathbf{ALL})$. For all $\mathbf{F} \in power(\mathbf{ALL})$, let *a-Coding*$(\mathbf{F}) = \{a\text{-}Coding(L) \mid L \in \mathbf{F}\}$, where *a-Coding*$(L) = \{a\text{-}Coding(x) \mid x \in L\}$.

In the following lemma and observation, we narrow our attention to **CF** and **REG**.

Lemma 4.5. *Let* $L \in \mathbf{CF}$*; then, a-Coding*$(L) \in \mathbf{REG} \cap power(\{a\}^*)$.

Proof (sketch). Let $L \in \mathbf{CF}$. Let G be a context-free grammar in the Chomsky normal form such that $L(G) = L$. In G, change every production $A \to b$ to $A \to a\text{-}Coding(b)$, where b is a terminal; otherwise, keep G unchanged. Let H be the context-free grammar resulting from this simple change. Clearly, *a-Coding*$(L) = a\text{-}Coding(L(G))$ and *a-Coding*$(L(G)) \in power(\{a\}^*)$. Hence, *a-Coding*$(L(G)) \in \mathbf{REG}$ because every context-free language over $\{a\}$ is regular (see Theorem 6.3.1 on page 194 in [8]). Thus, *a-Coding*$(L) \in \mathbf{REG} \cap power(\{a\}^*)$, so this lemma, whose fully rigorous proof is left to the reader, holds true. $\square$

Observation 4.6. *a-Coding*$(\mathbf{CF}) = \mathbf{REG} \cap power(\{a\})^*$*, so a-Coding properly reduces* **CF** *onto* $\mathbf{REG} \cap power(\{a\}^*)$.

Proof. Take any $L \in \mathbf{REG} \cap power(\{a\}^*)$. Clearly, $L \in a\text{-}Coding(\mathbf{REG})$, so $L \in a\text{-}Coding(\mathbf{CF})$. Hence, $\mathbf{REG} \cap power(\{a\}^*) \subseteq a\text{-}Coding(\mathbf{CF})$. From Lemma 4.5, *a-Coding*$(\mathbf{CF}) \subseteq \mathbf{REG} \cap power(\{a\}^*)$. Thus, Observation 4.6 holds. $\square$

Let $n \in \mathbb{N}$. Recall that ${}_nG$ means that G has n nonterminals (see Section 2). For all $L \in \mathbf{ALL}$, define ${}_n\,\mathrm{GrammaticalDefinition}(L) = L$ if there exists a grammar ${}_nG$ such that $L({}_nG) = L$, and ${}_n\,\mathrm{GrammaticalDefinition}(L) = \emptyset$ otherwise.

For all $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, define operation

$${}_n\,\mathrm{GrammaticalDefinition}(\mathbf{F}) = \{{}_n\,\mathrm{GrammaticalDefinition}(L) \mid L \in \mathbf{F}\}.$$

Observation 4.7. ${}_2\,\mathrm{GrammaticalDefinition}(\mathbf{ALL}) = \mathbf{RE}$*, so* ${}_2\,\mathrm{GrammaticalDefinition}$ *properly reduces* **ALL** *onto* **RE***, for* $i = 2, 3, \ldots$.

Proof. To prove $\mathbf{RE} \subseteq {}_2\,\mathrm{GrammaticalDefinition}(\mathbf{ALL})$, take any $L \in \mathbf{RE}$. If $L = \emptyset$, then ${}_2\,\mathrm{GrammaticalDefinition}(L) = \emptyset$, and there obviously exists a grammar ${}_2G$ so $L({}_2G) = \emptyset$. Let $L \in \mathbf{RE}$ and $L \neq \emptyset$. By Church's thesis, there is a grammar ${}_nG$, $L({}_nG) = L$, for some $n \geq 2$. Let ${}_nG = (N, T, P, S), \{0, 1\} \subseteq N$, and $a \in T$. Introduce a homomorphism h from $N \cup T$ into $\{1\}\{a\}^+\{1\}$. Next, we construct a grammar ${}_2H$ so $L({}_nG) = L({}_2H)$. Set ${}_2H = (\{0, 1\}, T, R, 0)$ with

$$\begin{aligned} R = &\{0 \to 111h(S)1111, 1111111 \to \varepsilon\} \\ &\cup \{h(x) \to h(y) \mid x \to y \in P\} \\ &\cup \{111h(b) \to b111 \mid b \in T\}. \end{aligned}$$

A rigorous proof that $L({}_nG) = L({}_2H)$ is simple and left to the reader.

Thus, $\mathbf{RE} \subseteq {}_2\,\mathrm{GrammaticalDefinition}(\mathbf{ALL})$. By Church's thesis, a language is generated by a grammar if and only if it belongs to **RE**, so we definitely have ${}_2\,\mathrm{GrammaticalDefinition}(ALL) \in \mathbf{RE}$. Thus, the observation holds true. $\square$

Can Obseration 4.7 be established for $i = 1$? The answer is no as proved next.

Observation 4.8. ${}_1\,\mathrm{GrammaticalDefinition}(\mathbf{ALL}) \subset \mathbf{RE}$*, so* ${}_1\,\mathrm{GrammaticalDefinition}$ *properly reduces* **ALL** *into* **RE**.

Proof (by Contradiction). Set $L_{prime} = \{a^j \mid j \text{ is a prime}\}$. Of course, $L_{prime} \in \mathbf{RE}$. For the sake of contradiction, assume that a grammar ${}_1G = (\{S\}, \{a\}, P, S)$ satisfies $L_{prime} = L({}_1G)$. Having a single nonterminal, ${}_1G$ would have derivations in the form

$$S \Rightarrow^* uSw \Rightarrow^* uvSw \Rightarrow^* uvvSw \Rightarrow^* \cdots \Rightarrow^* uv^iSw \Rightarrow^* uv^iw,$$

where $u, w \in T^*, |uw| \geq 2, v \in T^+$, so $uv^iw \in L_{prime}$ for all $i \geq 0$. Of course, $uv^iw \in L_{prime}$. Set $|uw| = m$. Take uv^mw. Observe that $|uv^mw| = |uw| + m|v| = m + m|v| = m(1 + |v|)$, so $uv^mw \notin L_{prime}$—a contradiction. Thus, this observation holds. $\square$

For all $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, define operations

$$\mathit{Union}(\mathbf{F}) = \{L \cup K \mid K, L \in \mathbf{F}\}, \text{ and}$$
$$\mathit{DifferentUnion}(\mathbf{F}) = \{L \cup K \mid K, L \in \mathbf{F}, K \notin L\}.$$

As it is shown next, while the latter represents a reducing operation (see Observation 4.10), the former does not (see Observation 4.11).

Lemma 4.9. *Let $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, and let $L \in \mathbf{F}$ satisfy $\mathit{card}(L) \leq \mathit{card}(K)$, for all $K \in \mathbf{F}$. Then, $L \notin \mathit{DifferentUnion}(\mathbf{F})$.*

Proof. Let $L \in \mathbf{F}$ with $\mathit{card}(L) \leq \mathit{card}(K)$, for all $K \in \mathbf{F}$. If $L = \emptyset, \emptyset \cup \emptyset \notin \mathit{DifferentUnion}(\mathbf{F})$, so $\emptyset \notin \mathit{DifferentUnion}(\mathbf{F})$. Let $L \neq \emptyset$, $\mathit{card}(L) \leq \mathit{card}(K)$. By the definition of $\mathit{DifferentUnion}$, $L - \emptyset \notin \mathit{DifferentUnion}(\mathbf{F})$ and $L \cup L \notin \mathit{DifferentUnion}(\mathbf{F})$. Therefore, every $J \in \mathit{DifferentUnion}(\mathbf{F})$ satisfies $\mathit{card}(L) < \mathit{card}(J)$. Thus, $L \notin \mathit{DifferentUnion}(\mathbf{F})$. $\square$

Observation 4.10. *For all $\mathbf{F} \subseteq \mathbf{ALL}$, $\mathit{DifferentUnion}$ reduces $\mathbf{F}$.*

Proof. Recall that we always assume $\mathbf{F} \neq \emptyset$ (see Section 3) to see that Lemma 4.9 implies this observation. $\square$

4.2 Expanding Operation

Throughout the rest of this section, we discuss mostly operations that unchange or, more often, expand language families (see parts 6 and 7 in Definition 3.1).

Observation 4.11. *Let* $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$*. If there are* $K, L \in \mathbf{F}$ *such that* $L \cup K \notin \mathbf{F}$*, then Union expands* $\mathbf{F}$*; otherwise, Union unchanges* $\mathbf{F}$*.*

Proof. Let $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$. Notice that $\mathbf{F} = \{J \cup J \mid J \in \mathbf{F}\}$. If there are $K, L \in \mathbf{F}$ such that $L \cup K \notin \mathbf{F}$, then $\mathbf{F} \subset \mathit{Union}(\mathbf{F})$; otherwise, $\mathbf{F} = \mathit{Union}(\mathbf{F})$. □

For all $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, define $\mathit{Intersection}(\mathbf{F}) = \{L \cap K \mid K, L \in \mathbf{F}\}$.

Observation 4.12. *Let* $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$*. If there are* $K, L \in \mathbf{F}$ *such that* $L \cap K \notin \mathbf{F}$*, then Intersection expands* $\mathbf{F}$*; otherwise, Intersection unchanges* $\mathbf{F}$*.*

Proof. By analogy with the proof of Observation 4.11. □

Let *Homomorphism* denote the common language operation of homomorphism (*a-Coding*, discussed in Lemma 4.5 and Observation 4.6, represents its special case). For all $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, define:

$$\mathit{Homomorphism}(\mathbf{F}) = \{\mathit{Homomorphism}(L) \mid L \in \mathbf{F}\}.$$

Observation 4.13. *Homomorphism expands* **CS** *onto* **RE**.

Proof. By Theorem 9.10 in [5], $\mathbf{RE} \subseteq \mathit{Homomorphism}(\mathbf{CS})$. By Church's thesis, $\mathit{Homomorphism}(\mathbf{CS}) \subseteq \mathbf{RE}$. Thus, Observation 4.13 holds true. □

Throughout the rest of this section, we narrow our attention to operations applied only to **CF** or its subfamilies.

Observation 4.14. *Homomorphism unchanges* **CF**.

Proof. $Homomorphism(\mathbf{CF}) \subseteq \mathbf{CF}$ (see Theorem 8.12 in [2]). To prove that $\mathbf{CF} \subseteq Homomorphism(\mathbf{CF})$, take any $L \in CF$. Consider the homomorphism h over $alph(L)^*$ as the identity $h(a) = a$, for all $a \in alph(L)$. Clearly, $h(L) = L$, so $\mathbf{CF} \subseteq Homomorphism(\mathbf{CF})$. Thus, Observation 4.14 holds true. □

Next, we state some specific results concerning *Union* applied to **CF** and its proper subfamily of inherently ambiguous context-free languages, denoted by ${}_{amb}\mathbf{CF}$. Set ${}_{unamb}\mathbf{CF} = \mathbf{CF} - {}_{amb}\mathbf{CF}$.

Observation 4.15. *Union unchanges* **CF**.

Proof. For any $K, L \in \mathbf{CF}, L \cup K \in \mathbf{CF}$, so this observation follows from Obsevation 4.11. □

Observation 4.16. *Union expands* ${}_{unamb}\mathbf{CF}$ *into* **CF**.

Proof. Of course, ${}_{unamb}\mathbf{CF} \subseteq Union({}_{unamb}CF) \subseteq \mathbf{CF}$. Take $L = \{a^n b^n c^m : n, m \geq 1\}$ and $K = \{a^m b^n c^n \mid n, m \geq 1\}$, both of which are in ${}_{unamb}\mathbf{CF}$. Recall that $L \cup K = \{a^i b^j c^k \mid i, j, k \geq 1, i = j \text{ or } j = k\}$ belongs to ${}_{amb}\mathbf{CF}$ (see Example 2.47 on page 205 in [9]). Thus, ${}_{unamb}CF \subset Union({}_{unamb}\mathbf{CF}) \subseteq \mathbf{CF}$, so this observation holds true. □

For every context-free grammar G, set

$$
{}_{CFG}Amb(L(G)) = \{x \in L(G) \mid x \text{ is the frontier of two or more distinct derivation trees for } G\}.
$$

Lemma 4.17. *Let* $L \in \mathbf{RE}$. *Then,* $L = {}_{CFG}Amb(L(G))$, *where* G *is a context-free grammar.*

Proof. Let $L \in RE$. Express L as $L = h(L(I) \cap L(J))$, where $L(I)$ and $L(J)$ are deterministic context-free languages, and I, J are unambiguous context-free grammars (see Theorem 10.3.1 on page 310 in [8] and Theorem 6.21 on page 250 in [10]). Let $I = (N_I, T, P_I, S_I)$ and $J = (N_J, T, P_J, S_J)$, $N_I \cap N_J = \emptyset$. Define the homomorphism

g over $(N_I \cup N_J \cup T)^*$ as $g(A) = A$ for every $A \in N_I \cup N_J$ and $g(a) = h(a)$ for every $a \in T$. Construct the context-free grammar $G = (N_I \cup N_J \cup \{X\}, T, P_G, S_G)$, where X is a new nonterminal and

$$P_G = \{A \to g(x) \mid A \to x \in P_I \cup P_J\} \cup \{S_G \to XS_I, S_G \to S_J X, X \to \varepsilon\}.$$

Observe that $L(G) = h(L(I) \cup L(J))$. Recall that I and J are unambiguous. Thus, $(h(L(J)) - h(L(I))) \cap {}_{CFG}Amb(L(G)) = \emptyset$ and $(h(L(I)) - h(L(J))) \cap {}_{CFG}Amb(L(G)) = \emptyset$, so ${}_{CFG}Amb\ (L(G)) \subseteq h(L(I) \cap L(J))$. Notice that $\{S_G \to XS_I, S_G \to S_J X, X \to \varepsilon\} \subseteq P_G$. Thus, $h(L(I) \cap L(J))$ is necessarily contained in ${}_{CFG}Amb(L(G))$, so $h(L(I) \cap L(J)) \subseteq {}_{CFG}Amb(L(G))$. Hence, ${}_{CFG}Amb(L(G)) = h(L(I) \cap L(J)) = L$. Therefore, Lemma 4.17 holds. □

Define language-family operation ${}_{CF}Amb$ as follows. For every $\mathbf{F} \subseteq \mathbf{CF}$:

$${}_{CF}Amb(\mathbf{F}) = \{{}_{CFG}Amb(L(G)) \mid G \text{ is a context-free grammar}\},$$

and for every $\mathbf{F} \subseteq \mathbf{ALL} - \mathbf{CF}, {}_{CF}Amb(\mathbf{F}) = \emptyset$.

Observation 4.18. $\mathbf{RE} = {}_{CF}Amb(\mathbf{CF})$, *so* ${}_{CF}Amb$ *expands* $\mathbf{CF}$ *onto* $\mathbf{RE}$.

Proof. By Lemma 4.17, $\mathbf{RE} \subseteq {}_{CF}Amb(\mathbf{CF})$. From Church's thesis, ${}_{CF}Amb(\mathbf{CF}) \subseteq \mathbf{RE}$. Thus, Observation 4.18 holds. □

Next, we introduce *i-Power* as a language-family operation and demonstrate that its application to **LIN** gives rise to an infinite hierarchy of language families.

Let $i \in \mathbb{N}$. For all $\mathbf{F} \subseteq \mathbf{ALL}$, recursively define operation i-$Power(\mathbf{F})$ as follows: (i) 1-$Power(\mathbf{F}) = \mathbf{F}$, (ii) for all $i \geq 1$, $(i+1)$-$Power(\mathbf{F}) = \{LK \mid L \in i\text{-}Power(\mathbf{F}), K \in \mathbf{F}\}$.

Theorem 4.19. *For all* $i \geq 1$, i-$Power(\mathbf{LIN}) \subset i+1$-$Power(\mathbf{LIN})$*; in words,* $i+1$-*Power properly expands* i-$Power(\mathbf{LIN})$.

Proof. Let $j \in \mathbb{N}$. Observe that j-$Power(\mathbf{LIN}) = {}_j\mathbf{LIN}$ (see Section 2 for ${}_j\mathbf{LIN}$); a proof of this observation is simple and left to the reader. Recall that for all $i \geq 1, {}_i\mathbf{LIN} \subset {}_{i+1}\mathbf{LIN}$ (see Theorem 8.3 in [5]). Thus, Theorem 4.19 holds true. □

In what follows, without any loss of generality, we assume that # represents a special delimited marker exclusively used as described in the following definitions of operations *Middle* and *SymmetricMiddle*.

For all $L \in \mathbf{ALL}$, define

$$Middle(L) = \{w \mid x\#w\#y \in L, x, w, y \in (alph(L) - \#)^*\}, \text{ and}$$

$$SymmetricMiddle(L) =$$

$$= \{w \mid x\#w\#y \in L, x, w, y \in (alph(L) - \#)^*, x = reversal(y)\}.$$

For all $\mathbf{F} \subseteq \mathbf{ALL}$, define

$$Middle(\mathbf{F}) = \{Middle(L) \mid L \in F\}, \text{ and}$$

$$SymmetricMiddle(\mathbf{F}) = \{SymmetricMiddle(L) \mid L \in \mathbf{F}\}.$$

At a glance, *Middle* and *SymmetricMiddle* resemble each other very much. However, while *Middle* unchanges ${}_j\mathbf{LIN}$ for any $j \geq 1$, *SymmetricMiddle* expands **LIN** onto **RE**.

Theorem 4.20. *Let $j \in \mathbb{N}$. $Middle({}_j\mathbf{LIN}) = {}_j\mathbf{LIN}$, so Middle unchanges ${}_j\mathbf{LIN}$.*

Proof. Let $j \in \mathbb{N}$. To prove $Middle({}_j\mathbf{LIN}) \subseteq {}_j\mathbf{LIN}$, take any $L \in Middle({}_j\mathbf{LIN})$. That is, $L = Middle(L(G))$, where $G = (N, T, P, S)$ is a j-linear grammar. Assume that in every rule of the form $S \to x \in P, x \in N^j$, and every nonterminal is terminating—that is, there is a derivation of a terminal word starting from it. A simple proof that any j-linear grammar can be turned to a j-linear grammar satisfying this assumption is simple and left to the reader.

Next, we construct a j-linear grammar H so $Middle(L(G)) = L(H)$. Set $H = (M, T, R, \langle S \rangle)$, whose components are constructed as follows. Set

$$M = \{\langle aAb\rangle \mid A \in N, a, b \in \{[,], \$, \varepsilon\}, ab \in \{\varepsilon, \$\$, \$], [\$, []\}\},$$

where [,], and \$ are new symbols not contained in $N \cup T$. Construct R by performing 1 through 8, given next. In this construction, we automatically assume that $u, v, w, x, y, z \in T^*$, and $A, B \in N - \{S\}$. Initially, set $R = \emptyset$. Perform

1. for all $S \rightarrow A_1 \dots A_h \dots A_j \in P$, where $h \in \{1, \dots, j\}$, add $\langle S\rangle \rightarrow \langle [A_h]\rangle$ to R;

2. for all $S \rightarrow A_1 \dots A_h A_{h+1} \dots A_{i-1} A_i \dots A_j \in P$, where $h \in \{1, \dots, j-1\}, i \in \{h+1, \dots, j\}$, add $\langle S\rangle \rightarrow$ $\langle [A_h\$\rangle\langle \$A_{h+1}\$\rangle \dots\dots \langle \$A_{i-1}\$\rangle\langle \$A_i]\rangle$ to R;

3. for all $A \rightarrow uBv \in P$, add $\langle A\rangle \rightarrow \langle B\rangle$ and $\langle \$A\$\rangle \rightarrow u\langle \$B\$\rangle v$ to R;

4. for all $A \rightarrow uBv\#y \in P$, add $\langle \$A]\rangle \rightarrow u\langle \$B\$v\rangle$ to R;

5. for all $A \rightarrow u\#vBy \in P$, add $\langle [A\$\rangle \rightarrow v\langle \$B\$\rangle y$ to R;

6. for all $A \rightarrow u\#vBx\#y \in P$, add $\langle [A]\rangle \rightarrow v\langle \$B\$\rangle x$ to R;

7. for all rules of the form $A \rightarrow vBu\#x\#y, A \rightarrow u\#x\#vBy$, and $A \rightarrow \#x\#P$, add $\langle [A]\rangle \rightarrow x$ to R;

8. for all rules of the form $A \rightarrow w \in P$, add $\langle \$A\$\rangle \rightarrow w$ to R.

Gist. *In essence, H uses [and] as boundary markers that delimit the corresponding subword w occurring in between the two #s generated in G. In this way, H determines every $w \in Middle(L(G))$ and generates it, so $L(H) = Middle(L(G))$. For instance, suppose that G makes*

$$
\begin{aligned}
S \Rightarrow & A_1 \dots A_h \dots A_j \\
& \Rightarrow^* u_1 A_h u_2 \\
& \Rightarrow u_1 u_2 \# u_3 A_2 u_4 u_2 \\
& \Rightarrow^* u_1 u_2 \# u_3 u_4 A_3 u_5 u_4 u_2 \\
& \Rightarrow u_1 u_2 \# u_3 u_4 u_6 A_4 u_7 \# u_8 u_5 u_4 u_2 \\
& \Rightarrow^* u_1 u_2 \# u_3 u_4 u_6 u_9 A_5 u_{10} u_7 \# u_8 u_5 u_4 u_2 \\
& \Rightarrow u_1 u_2 \# u_3 u_4 u_6 u_9 u_{11} u_{10} u_7 \# u_8 u_5 u_4 u_2.
\end{aligned}
$$

As a result, $u_3u_4u_6u_9u_{11}u_{10}u_7 \in Middle(\mathbf{LIN})$*. Then,* H *simulates the generation of the string* $u_3u_4u_6u_9u_{11}u_{10}u_7$ *in this way*

$$
\begin{aligned}
\langle S \rangle & \Rightarrow^* \langle [A_h] \rangle \\
& \Rightarrow u_3 \langle \$ A_2] \rangle \\
& \Rightarrow^* u_3 u_4 \langle \$ A_3] \rangle \\
& \Rightarrow u_3 u_4 u_6 \langle \$ A_4 \$ \rangle u_7 \\
& \Rightarrow^* u_3 u_4 u_6 u_9 \langle \$ A_4 \$ \rangle u_{10} u_7 \\
& \Rightarrow u_3 u_4 u_6 u_9 u_{11} u_{10} u_7.
\end{aligned}
$$

Consider all other possible forms of generating $x\#w\#y \in L(G)$ *such that* $w \in Middle(L(G))$*.* H *simulates them by analogy with the simulation sketched above, so* $Middle(L(G)) \subseteq L(H)$*. Similarly, we can establish* $L(H) \subseteq Middle(L(G))$*, so* $Middle(L(G)) = L(H)$*. A fully rigorous proof of this identity is simple, but lengthy and tedious, so we omit it; the reader can easily fill in all the details.*

Thus, this theorem holds true. □

Considering Theorem 4.20, we find it surprising that *SymmetricMiddle* properly expands **LIN** onto **RE** (see Theorem 4.28). Since proving this result is more complicated than the previous proof, we provide it in a greater detail. To start with, we need the notion of a queue grammar.

A *queue grammar* (see [1]) is a six tuple, $Q = (V, T, W, F, s, P)$, where V and W are alphabets satisfying $V \cap W = \emptyset, T \subseteq V, F \subseteq W, s \in (V - T)(W - F)$, and $P \subseteq (V \times (W - F)) \times (V^* \times W)$ is a finite relation such that for every $a \in V$, there exists an element $(a, b, x, c) \in P$. If $u, v \in V^*W$ such that $u = arb; v = rzc; a \in V; r, z \in V^*; b, c \in W$; and $(a, b, z, c) \in P$, then $u \Rightarrow v[(a, b, z, c)]$ in G or, simply, $u \Rightarrow v$. In the usual manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The language of Q, $L(Q)$, is defined as $L(Q) = \{w \in T^* : s \Rightarrow^* wf, \text{ where } f \in F\}$. Now, we slightly modify the notion of a queue grammar. A *left-extended queue grammar* is a sixtuple , $Q = (V, T, W, F, s, P)$, where V, T, W, F, and s have the same meaning as in a queue grammar. $P \subseteq (V \times (W - F)) \times (V \times W)$ is a finite relation (as opposed to an ordinary queue grammar, this definition does not require that for every $a \in V$, there exists an element $(a, b, x, c) \in P$). Furthermore, assume that $\# \notin V \cup W$. If $u, v \in V^*\{\#\}V^*W$ so that $u = w\#arb; v = wa\#rzc$; $a \in V; r, z, w \in V^*$; $b, c \in W$; and $(a, b, x, c) \in P$, then $u \rightarrow v[(a, b, z, c)]$ in G or, simply, $u \Rightarrow v$. In the usual manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$. The language of $Q, L(Q)$, is defined as $L(Q) = \{v \in T^* : \#s \Rightarrow^* w\#vf \text{ for some } w \in V^* \text{ and } f \in F\}$. Less formally, during every step of a derivation, a left-extended queue grammar shifts the rewritten symbol over #; in this way, it records the derivation history, which plays a crucial role in the proof of Lemma 4.22.

Lemma 4.21. *For every recursively enumerable language, L, there exists a left-extended queue grammar, Q, satisfying $L(Q) = L$.*

Proof. Recall that every recursively enumerable language is generated by a queue grammar (see [1], [3]). Clearly, for every queue grammar, there exists an equivalent left-extended queue grammar. Thus, this lemma holds. □

Lemma 4.22. *Let H be a left-extended queue grammar. Then, there exists a left-extended queue grammar, $Q = (V, T, W, F, s, R)$, such that $L(H) = L(Q)$ and every $(a, b, x, c) \in R$ satisfies $a \in V - T, b \in W - F$, and $x \in ((V - T)^* \cup T^*)$.*

Proof. Let $H = (\varsigma, T, \Omega, \phi, \sigma, \Pi)$ be any left-extended queue grammar. Set $\Omega' = \{q' : q' \in \Omega\}$, $\Omega'' = \{q'' : q'' \in \Omega\}$, and $\varsigma' = \{a' : a \in \varsigma\}$. Define the bijection α from Ω to Ω' as $\alpha(q) = q'$ for every $q \in \Omega$. Analogously, define the bijection β from Ω to Ω'' as $\beta(q) = q''$ for every $q \in \Omega$. Finally, define the bijection δ from ς to ς' as $\delta(a) = a'$ for every $a \in \varsigma$. In the standard manner, extend δ so it is defined from ς^* to $(\varsigma')^*$. Set

$$U = \{\langle y, p\rangle : y \in T^*, p \in \Omega, \text{ and } (a, q, xy, p) \in \Pi$$

$$\text{for some a } \in \varsigma, q \in \Omega, x \in \varsigma^*\}.$$

Without any loss of generality, assume that $(\delta(\varsigma) \cup T \cup \alpha(\Omega) \cup \beta(\Omega) \cup U) \cap \{1, f\} = \emptyset$. Set $V = \delta(\varsigma) \cup \{1\} \cup T, W = \alpha(\Omega) \cup \beta(\Omega) \cup \{f\} \cup U, F = \{f\}$, and $s = \delta(a)\alpha(q)$. Define the left-extended queue grammar

$$Q = (V, T, W, F, s, R)$$

with R constructed in the following way:

I if $(a, q, xy, p) \in \Pi$, where $a \in \varsigma$; $q \in \Omega - \Phi; x, y \in \varsigma^*$; and $p \in \Omega$, then add $(\delta(a), \alpha(q), \delta(x)\delta(y), \alpha(p))$ and $(\delta(a), \alpha(q), \delta(x)1\delta(y), \alpha(p))$ to R;

II if $(a, q, xy, p) \in \Pi$, where $a \in \varsigma, q \in \Omega - \Phi, x \in \varsigma^*, y \in T^*, p \in \Omega(\langle y, p\rangle \in U)$, then add $(\delta(a), \alpha(q), \delta(x), \langle y, p\rangle)$ and $(1, \langle y, p\rangle, y, \beta(p))$ to R;

III if $(a, q, x, p) \in \Pi$, where $a \in \varsigma, q \in \Omega - \Phi, x \in T^*$, and $p \in \Omega$, then add $(\delta(a), \beta(q), \delta(x), \beta(p))$ to R;

IV if $(a, q, x, p) \in \Pi$, where $a \in \varsigma$, $q \in \Omega - \Phi, x \in T^*$, and $p \in \Phi$, then add $(\delta(a), \beta(q), x, f)$ to R (recall that $F = \{f\}$).

Clearly, for every $(a, b, x, c) \in R$, $a \in V - T, b \in W - F$, and $x \in ((V - T)^* \cup T^*)$. Leaving a rigorous proof that $L(H) = L(Q)$ to the reader, we next give its sketch.

To see that $L(H) \subseteq L(Q)$, consider any $v \in L(H)$. As $v \in L(H)$,

$$\#\sigma \Rightarrow^* w\#vt$$

in $H, w \in \varsigma^*$, $v \in T^*$, and $t \in \Phi$. Express $\#\sigma \Rightarrow^* w\#vt$ in H as

$$\#\sigma \Rightarrow^* u\#zq \Rightarrow ua\#xyp \Rightarrow^* w\#vt,$$

where $a \in \varsigma, u, x \in \varsigma^*$, $y = \mathit{prefix}(v, |y|)$, $z = ax, w = uax$, and during $ua\#xyp \Rightarrow^* w\#vt$, only terminals are generated so that the resulting terminal string equals v. Q simulates $\#\sigma \Rightarrow^* u\#zq \Rightarrow ua\#xyp \rightarrow *w\#vt$ as follows. First, Q uses productions introduced in I to simulate $\#\sigma \Rightarrow^* u\#zq$. During this initial simulation, it once uses a production that generates 1 so that it can then simulate $u\#zq \Rightarrow ua\#xyp$ by making two derivation steps according to productions $(\Delta(a), \alpha(q), \Delta(x), \langle y, p\rangle)$ and $(1, \langle y, p\rangle, y, \beta(p))$ (see II). Notice that by using $(1, \langle y, p\rangle, y, \beta(p))$, Q produces y, which is a prefix of v. After the application of $(1, \langle y, p\rangle, y, \beta(p))$, Q simulates $ua\#xyp \rightarrow *w\#vt$ by using productions introduced in III followed by one application of a production constructed in IV, during which Q enters f and, thereby, completes the generation of v. Thus, $L(H) \subseteq L(Q)$.

To establish $L(Q) \subseteq L(H)$, consider any $v \in L(Q)$. Since $v \in L(Q)$,

$$\#s \Rightarrow^* w\#vf$$

in Q, where $w \in V^*$ and $v \in T^*$. Examine I through IV. Observe that Q passes through states of $\alpha(W), U, \beta(W)$, and $\{f\}$ in this order so that it occurs several times in states of $\alpha(W)$, once in a state of U, several times in $\beta(W)$, and once in f. As a result, Q uses productions introduced in I, and during this initial part of derivation it precisely once uses a production that generates 1 so that it can subsequently make two consecutive derivation steps according to $(\delta(a), \alpha(q), \delta(x), \langle y, p\rangle)$ and $(1, \langle y, p\rangle, y, \beta(p))$ (see II). By using the latter, Q produces y, which is a prefix of v. After the application of $(1, \langle y, p\rangle, y, \beta(p))$, Q applies productions introduced in III, which always use states of $\beta(\Omega)$. Finally, it once applies a production constructed in IV to enter f and, thereby, complete the generation of v. To summarize these observations, we can express $\#s \Rightarrow^* w\#vf$ in Q as

$$\#s \Rightarrow^* u\#zq \Rightarrow ua\#xyp \Rightarrow^* w\#vf,$$

where $a \in V, x \in V^*, y \in T^*, w = uax$ so that during $\#s \Rightarrow^* u\#zq$, Q uses productions introduced in I, then it applies $(1, \langle y, p\rangle, y, \beta(p))$ from II to make $u\#zq \Rightarrow ua\#xyp$, and finally, it performs $ua\#xyp \Rightarrow^* w\#vf$ by several applications of productions introduced in III and one application of a production constructed in IV. At this point, based on an examination of I through IV, we see that H makes

$$\#i\sigma \Rightarrow^* u\#zq \Rightarrow ua\#xyp \Rightarrow^* w\#vt$$

with $t \in \Phi$, so $v \in L(H)$. Therefore, $L(H) \subseteq L(Q)$.

As $L(H) \subseteq L(Q)$ and $L(Q) \subseteq L(H)$, $L(H) = L(Q)$. □

Lemma 4.23. *Let Q be a left-extended queue grammar. Then, there exists a linear grammar, $G = (N, T, P, S)$, such that $L(Q) = SymmetricMiddle(L(G))$.*

Proof. Let $Q = (V, T, W, F, s, R)$ be a left-extended queue grammar. Without any loss of generality, assume that Q satisfies the properties described in Lemma 4.22 and that $\{0, 1, \} \cap (V \cup W) = \emptyset$. For some positive integer n, define an injection ι from VW to $(\{0, 1\}^n - 1^n)$ so that ι is an injective homomorphism when its domain is extended to $(VW)^*$; after this extension, ι thus represents an injective homomorphism from $(VW)^*$ to $(\{0, 1\}^n - 1^n)^*$ (a proof that such an injection necessarily exists is simple and left to the reader). Based on ι, define the substitution ν from V to $(\{0, 1\}^n - 1^n)$ as $\nu(a) = \{\iota(aq) : q \in W\}$ for every $a \in V$. Extend the domain of ν to V^*. Furthermore, define the substitution μ from W to $(\{0, 1\}^n - 1^n)$ as $\mu(q) = \{reversal(\iota(aq)) : a \in V\}$ for every $q \in W$. Extend the domain of μ to W^*. Set $U = \{\langle p, i\rangle : p \in W - F \text{ and } i \in \{1, 2\}\} \cup \{S\}$.

Construction. *Introduce the linear grammar $G = (U, T \cup \{0, 1, \#\}, P, S)$ with P constructed in the following way. Initially, set $P = \emptyset$. To construct P, perform the following steps 1 through 5.*

1 *if* $a_0q_0 = s$, *where* $a \in V - T$ *and* $q \in W - F$, *then add* $S \to u\langle q, 1\rangle v$ *to* P, *for all* $u \in \nu(a_0)$ *and* $v \in \mu(q_0)$;

2 *if* $(a, q, y, p) \in R$, *where* $a \in V - T, p, q \in W - F$, *and* $y \in (V - T)^*$, *then add* $\langle q, 1\rangle \to u\langle p, 1\rangle v$ *to* P, *for all* $u \in \nu(y)$ *and* $v \in \mu(p)$;

3 *for every* $q \in W - F$, *add* $\langle q, 1\rangle \to \#\langle q, 2\rangle$ *to* P;

4 *if* $(a, q, y, p) \in R$, *where* $a \in V - T, p, q \in W - F, y \in T^*$, *then add* $\langle q, 2\rangle \to y\langle p, 2\rangle v$ *to* P, *for all* $v \in \mu(p)$;

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

Basic Idea. *G can generate every $y \in L(G)$ as $S \Rightarrow^* u_0\#y\#v_0$, where $u_0 \in \nu(a_0a_m)$ with $a_0, \ldots, a_m \in T, u_i \in suffix(n(a_0 \ldots a_m), |\nu(a_0 \ldots a_m)| -i)$ for $i = 1, \ldots, m-1$, $v_0 \in \nu(q_mq_0)$ with $q_0, \ldots, q_m \in Q, v_j \in prefix(\mu(q_m \ldots q_0), |\mu(q_m \ldots q_0))| - j)$ for $j = 1, \ldots, m-1$, $u_0 = reversal(v_0)$. Examine the construction of P to see that $S \Rightarrow^* u_0\#y\#v_0$ in G if and only if Q makes $a_0q_0 \Rightarrow^* a_0 \ldots a_myf$ according to (a_0, q_0, z_0, q_1) through (a_m, q_m, z_m, q_{m+1}), where $q_{m+1} \in F$. From this equivalence, $L(Q) = SymmetricMiddle(L(G))$.*

Formal Proof. *For brevity, the following rigorous proof omits some obvious details, which the reader can easily fill in. Claim 4.24, proved next, establishes a derivation form by which G can generate each member of $L(G)$. This claim fulfills a crucial role in the demonstration that $SymmetricMiddle(L(G)) \subseteq L(Q)$, given later in this proof (see Claim 4.26).*

Claim 4.24. *G can generate every $h \in L(G)$ in this way*

$$
\begin{aligned}
&S\\
&\Rightarrow X\langle q_0, 1\rangle t_0 \Rightarrow g_0\langle q_1, 1\rangle t_1 \Rightarrow \cdots \Rightarrow g_{k-1}\langle q_k, 1\rangle t_k\\
&\Rightarrow g_k\langle q_{k+1}, 1\rangle t_{k+1} \Rightarrow g_k\#\langle q_{k+1}, 2\rangle t_{k+1}\\
&\Rightarrow g_k\#y_1\langle q_{k+2}, 2\rangle t_{k+2} \Rightarrow g_k\#y_1y_2\langle q_{k+3}, 2\rangle t_{k+3} \Rightarrow \ldots\\
&\Rightarrow^* g_k\#y_1y_2\ldots y_{m-1}\langle q_{k+m}, 2\rangle t_{k+m}\\
&\Rightarrow g_k\#y_1y_2\ldots y_{m-1}y_m\#t_{k+m}
\end{aligned}
$$

in G, where $k, m \geq 1; q_0, q_1, \ldots, q_{k+m}W - F; y_1, \ldots, y_m \in T^*; X \in \nu(a_0)$, where $a_0 \in (V - T)$ and $s = a_0q_0$; $t_i \in \mu(q_i \ldots q_1q_0)$ for $i = 0, 1, \ldots, k{+}m; g_j \in \nu(d_0d_1 \ldots d_j)$ with $d_0 = a_0$ and $d_1, \ldots, d_j \in (V{-}T)^*$ for $j = 0, 1, \ldots, k$; $d_0d_1 \ldots d_k = a_0a_1 \ldots a_{k+m}$ with $a_1, \ldots, a_{k+m} \in V - T$ (that is, $g_k \in \nu(a_0a_1 \ldots a_{k+m})$); $g_k = reversal(t_{k+m})$; $h = y_1y_2 \cdots y_{m-1}y_m$.

Proof of Claim 4.24. Examine the construction of P. Observe that every derivation begins with an application of a production having S on its left-hand side. Set 1-$U = \{\langle p, 1\rangle : p \in W\}$, 2-$U = \{\langle p, 2\rangle : p \in W\}$, 1-$P = \{p : p \in P \text{ and } lhs(p) \in 1\text{-}U\}$, and 2-$P = \{p : p \in P \text{ and } lhs(p) \in 2\text{-}U\}$. Observe that in every successful derivation, all applications of productions from 1-P precede the applications of productions from 2-P. Furthermore, notice that

$$
\begin{aligned}
F(G) - \{S\} \subseteq &\{\#, \varepsilon\}\{0, 1\}^*(1\text{-}U)\{0, 1\}^*\{\#, \varepsilon\}\\
&\cup\{\#, \varepsilon\}\{0, 1\}^*T^*(2\text{-}U)\{0, 1\}^*\{\#, \varepsilon\}.
\end{aligned}
$$

Thus, we can always express the derivation so that the generation of

$h \in L(G)$ can be expressed as

$$
\begin{aligned}
& S \\
\Rightarrow & X\langle q_0, 1\rangle t_0 \Rightarrow g_0\langle q_1, 1\rangle t_1 \Rightarrow \cdots \Rightarrow g_{k-1}\langle q_k, 1\rangle t_k \\
\Rightarrow & g_k\langle q_{k+1}, 1\rangle t_{k+1} \Rightarrow g_k\#\langle q_{k+1}, 2\rangle t_{k+1} \\
\Rightarrow & g_k\#y_1\langle q_{k+2}, 2\rangle t_{k+2} \Rightarrow g_k\#y_1y_2\langle q_{k+3}, 2\rangle t_{k+3} \\
\Rightarrow & \cdots \Rightarrow g_k\#y_1y_2 \dots y_{m-1}\langle q_{k+m}, 2\rangle t_{k+m} \\
\Rightarrow & g_k\#y_1y_2y_{m-1}y_m\#t_{k+m},
\end{aligned}
$$

where all the involved symbols have the meaning described in Claim 4.24. During the first $|g_k|$ steps, every sentential form has the form $\gamma\#y_1y_2y_{m-1}y_m\$m\#\tau$ with $\gamma, \tau \in \{0,1\}^*, 0 \leq |\gamma| = |\tau| \geq |g_k|$, and $\gamma = reversal(\tau)$. Thus, $g_k = reversal(t_{k+m})$; $h = y_1y_2y_{m-1}y_m$. As a result, Claim 4.24 holds. □

Claim 4.25. *Q generates every $h \in L(Q)$ in this way*

$$
\begin{array}{lr}
\quad \#a_0q_0 & \\
\Rightarrow a_0\#x_0q_1 & [(a0, q0, z0, q1)] \\
\Rightarrow a_0a_1\#x_1q_2 & [(a_1, q_1, z_1, q_2)] \\
\dots & \\
\Rightarrow a_0a_1 \dots a_k\#x_kq_{k+1} & [(a_k, q_k, z_k, q_{k+1})] \\
\Rightarrow a_0a_1 \dots a_ka_{k+1}\#x_{k+1}y_1q_{k+2} & [(a_{k+1}, q_{k+1}, y_1, q_{k+2})] \\
\dots & \\
\Rightarrow a_0a_1 \dots a_ka_{k+1} \cdots a_{k+m-1}\#x_{k+m-1}y_1 \cdots y_{m-1}q_{k+m} & [(a_{k+m-1}, q_{k+m-1}, y_{m-1}, q_{k+m})] \\
\Rightarrow a_0a_1 \dots a_ka_{k+1} \cdots a_{k+m}\#y_1 \cdots y_mq_{k+m+1} & [(a_{k+m}, q_{k+m}, y_m, q_{k+m+1})],
\end{array}
$$

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

Proof of claim 4.25. Recall that Q satisfies the properties given in Lemma 4.22. These properties imply that Claim 4.25 holds. □

Claim 4.26. *Let G generate $h \in L(G)$ in the way described in Claim 4.24; then, $h \in L(Q)$.*

Proof of Claim 4.26. Let $h \in L(G)$. Take the generation of h as described in Claim 4.24. Taking this into consideration, examine the construction of P to see that, R contains $(a_0, q_0, z_0, q_1), \ldots, (a_k, q_k, z_k, q_{k+1})$, $(a_{k+1}, q_{k+1}, y_1, q_{k+2}), \ldots, (a_{k+m-1}, q_{k+m-1}, y_{m-1}, q_{k+m})$, $(a_{k+m}, q_{k+m}, y_m, q_{k+m+1})$, where $z_1, \ldots, z_k \in (V - T)^*$, and $y_1, \ldots, y_m \in T^*$. Then, Q makes the generation of h in the way described in Claim 4.25. Thus, $h \in L(Q)$. □

Claim 4.27. *Let Q generates $h \in L(Q)$ in the way described in Claim 4.25; then, $h \in L(G)$.*

Proof of Claim 4.27. This is left to the reader. □

Claims 4.24 through 4.27 imply that $L(Q) = L(G)$, so this lemma holds.

□

Theorem 4.28. **RE** $=$ *SymmetricMiddle*(**LIN**)*, so SymmetricMiddle properly expands* **LIN** *onto* **RE***.*

Proof. From Lemmas 4.21,4.22, and 4.23,
RE $\subseteq$ *SymmetricMiddle*(**LIN**). From Church's thesis,
SymmetricMiddle(**LIN**) $\subseteq$ **RE**. Thus, Theorem 4.28 holds. □

Let $L \in$ **ALL** with $card(alph(L)) \geq 2$. For all $L \in$ **ALL** and $F \subseteq$ **ALL**, define *BinarySymmetricMiddle*$(L) = \{w \mid x\#w\#y \in L, \# \notin alph(w), x, w, y \in (alph(L) - \{\#\})^*, x = reversal(y), card(alph(\{x, y\})) = 2\}$. For all $F \subseteq$ **ALL**, define *BinarySymmetricMiddle*(**F**) $=$ $\{$*BinarySymmetricMiddle*$(L) \mid L \in$ **F**$\}$.

Corollary 4.29. **RE** $=$ *BinarySymmetricMiddle*(**LIN**)*, so BinarySymmetricMiddle expands* **LIN** *onto* **RE***.*

Proof. This corollary follows from the demonstration of Theorem 4.28. The details are left to the reader. □

5 Concluding Remarks

This section closes the study by suggesting and illustrating five new investigation trends concerning the subject of this paper.

1 We can simplify some proofs above if we restrict our attention only to special cases of the results that are demonstrated. To illustrate, reconsider Theorem 4.20 and its proof. Next, we rephrase the result just for **LIN** and prove it in a simpler way than the proof of Theorem 4.20. Indeed, observe that in the following proof, the construction of H is shorter. The resulting H is also more succinct and economical with respect to the number of nonterminals.

Theorem 5.1. *Middle*(**LIN**) = **LIN**, *so Middle unchanges* **LIN**.

Proof. Let $j \in \mathbb{N}$. To prove $Middle(\mathbf{LIN}) \subseteq \mathbf{LIN}$, take any $L \in Middle(\mathbf{LIN})$. That is, $L = Middle(L(G))$, where $G = (N, T, P, S)$ is a linear grammar. Next, we construct a linear grammar H so $Middle(L(G)) = L(H)$. Set $H = (M, T, R, \langle S \rangle)$, whose components are constructed as follows. Set

$$M = \{\langle aAb \rangle : A \in N, a, b \in \{\#, \varepsilon\}\}$$

Initially, set $R = \emptyset$. Construct R by performing (1) through (5), given next, where $u, v, w, x, y, z \in T^*$, and $A, B \in N$.

1. for all $A \to uBv \in P$, add $\langle A \rangle \to \langle B \rangle$, $\langle \#A\# \rangle \to u\langle \#B\# \rangle v$, $\langle A\# \rangle \to \langle B\# \rangle v$, $\langle \#A \rangle \to u\langle \#B \rangle$, $\langle \#\#A \rangle \to \langle \#\#B \rangle$, and $\langle A\#\# \rangle \to \langle B\#\# \rangle$ to R;

2. for all $A \to uBv\#y \in P$, add $\langle A \rangle \to \langle B\# \rangle v$, $\langle \#A \rangle \to u\langle \#B\# \rangle v$, and $\langle A\# \rangle \to \langle B\#\# \rangle y$ to R;

3. for all $A \to u\#vBy \in P$, add $\langle A \rangle \to v\langle \#B \rangle$, $\langle A\# \rangle \to v\langle \#B\# \rangle y$, and $\langle \#A \rangle \to \langle \#\#B \rangle u$ to R;

4. for all $A \to u\#vBx\#y \in P$, add $\langle A \rangle \to v\langle \#B\# \rangle x$ to R;

5. for all $A \to uBv\#x\#y \in P$, add $\langle A \rangle \to \langle B\#\# \rangle x$ to R;

6. for all $A \to u\#v\#xBy \in P$, add $\langle A \rangle \to v\langle \#\#B \rangle x$ to R;
7. for all $A \to w \in P$, add $\langle \#A\# \rangle \to w$, $\langle \#\#A \rangle \to \epsilon$, and $\langle \#\#A \rangle \to \epsilon$ to R.

Basic Idea. *Suppose that G makes*

$$\begin{aligned} S \Rightarrow^* & u_1 A_1 u_2 \\ \Rightarrow & u_1 u_2 \# u_3 A_2 u_4 u_2 \\ \Rightarrow_* & u_1 u_2 \# u_3 u_4 A_3 u_5 u_4 u_2 \\ \Rightarrow & u_1 u_2 \# u_3 u_4 u_6 A_4 u_7 \# u_8 u_5 u_4 u_2 \\ \Rightarrow^* & u_1 u_2 \# u_3 u_4 u_6 u_9 A_5 u_{10} u_7 \# u_8 u_5 u_4 u_2 \\ \Rightarrow & u_1 u_2 \# u_3 u_4 u_6 u_9 u_{11} u_{10} u_7 \# u_8 u_5 u_4 u_2, \end{aligned}$$

where As and us are nonterminals and terminal strings, respectively. As a result, $u_3u_4u_6u_9u_{11}u_{10}u_7 \in Middle(\mathbf{LIN})$. Then, H simulates the generation of $u_3u_4u_6u_9u_{11}u_{10}u_7$ in this way

$$\begin{aligned} \langle S \rangle \Rightarrow^* & \langle A_1 \rangle \\ \Rightarrow & u_3 \langle \#A_2 \rangle \\ \Rightarrow^* & u_3 u_4 \langle \#A_3 \rangle \\ \Rightarrow & u_3 u_4 u_6 \langle \#A_4\# \rangle u_7 \\ \Rightarrow^* & u_3 u_4 u_6 u_9 \langle \#A_4\# \rangle u_{10} u_7 \\ \Rightarrow & u_3 u_4 u_6 u_9 u_{11} u_{10} u_7. \end{aligned}$$

Consider all the other possible generations of $x\#w\#y \in L(G)$ such that $w \in Middle(L(G))$. H simulates these generations by analogy with the simulation sketched above, so $Middle(L(G)) \subseteq L(H)$. Similarly, we can establish $L(H) \subseteq Middle(L(G))$. Thus, $Middle(L(G)) = L(H)$. A fully rigorous proof of this identity is simple, but lengthy and tedious, so we omit it because the reader can easily fill in all the details.

Thus, $Middle(\mathbf{LIN}) = \mathbf{LIN}$. □

2 Most classical books about formal languages contain many results concerning closure properties. Reconsider and reformulate them in terms of the notions introduced in the present paper. For instance, for all $\mathbf{F} \in \mathit{power}(\mathbf{ALL})$, define operation

$$\mathit{Iteration}(\mathbf{F}) = \{L^* \mid L \in \mathbf{F}\}.$$

Take $\mathit{Iteration}(\mathbf{REG})$. Clearly, $\{\varepsilon\}^* = \emptyset^* = \{\varepsilon\}$. For any $L \in \mathbf{REG} - \{\emptyset, \{\varepsilon\}\}, L^* \in {}_{inf}\mathbf{REG}$, where ${}_{inf}\mathbf{REG}$ denotes the family of infinite regular languages. As obvious, ${}_{inf}\mathbf{REG} - \mathit{Iteration}(\mathbf{REG}) \neq \emptyset$; in words, there exist (infinitely many) regular languages K satisfying $K \neq L^*$, for all $L \in \mathbf{REG}$. For instance, $K = \{a\} \cup \{b\}^*$ is a regular language that does not represent the iteration of any regular language. Thus, in terms of Definition 3.1, *Iteration* properly reduces $\mathbf{REG}$ into ${}_{inf}\mathbf{REG} \cup \{\varepsilon\}$.

3 In this paper, we restricted our attention to unary and binary language operations, and we only considered unary language-family operations. Drop this restriction. Study n-ary language operation as well as n-ary language-family operation in general, for any $n \geq 1$. To illustrate, define binary language-family operation *Intersection* from $(\mathit{power}(\mathbf{ALL}))^2$ into $\mathit{power}(\mathbf{ALL})$ so for all $\mathbf{E}, \mathbf{F} \in \mathit{power}(\mathbf{ALL})$,

$$Intersection(\mathbf{E}, \mathbf{F}) = \{K \cap L \mid K \in \mathbf{E}, L \in \mathbf{F}\}.$$

Set $\mathbf{UNARY} = \{L \mid L \in \mathbf{ALL}, \mathit{card}(\mathit{alph}(L))\} = 1\}$. Recall that $\mathbf{UNARY} \cap \mathbf{CF} \subseteq \mathbf{REG}$ (see Theorem 6.3.1 on page 194 in [8]). Thus, $\mathit{Intersection}(\mathbf{UNARY}, \mathbf{CF}) = \mathit{Intersection}(\mathbf{UNARY}, \mathbf{REG})$, so $\mathit{Intersection}(\mathbf{UNARY}, \mathbf{CF}) \subseteq \mathbf{REG}$ and $\mathit{Intersection}(\mathbf{UNARY}, \mathbf{CF} - \mathbf{REG}) = \emptyset$. Of course, further investigation in this direction would necessitate a proper generalization of Definition 3.1, restricted to unary language-family operations in this paper.

4 Apart from operations over language families, we can study operations over other mathematical notions, including notions used in formal language theory. To illustrate a study of this kind by an example closely related to the subject of the present paper, consider families of grammars. Let **GRAMMARS** denote the family of all grammars, and let ${}_n$**GRAMMARS** denote the family of all n-nonterminal grammars, where $n \in \mathbb{N}$ (see Section 2). Is there a total function f from **GRAMMARS** into ${}_n$**GRAMMARS** so $L(f(G)) = L(G)$ for every $G \in$ **GRAMMARS**? If so, what is the smallest $n \in \mathbb{N}$ for which such a function exists? Reconsider the proofs of Theorems 4.7 and 4.8 to see that we can always find such a function f from **GRAMMARS** into ${}_n$**GRAMMARS**, where $n = 2$ is the smallest number. That is, for $n = 1$, no function like this exists.

5 Apart from a theoretical viewpoint, results concerning operations over language families are important from a practical standpoint, too. For instance, take multilingual translators that contain parsers, whose techniques are restricted to a language family **F**. If prior to parsing, the translators modify some languages in **F** so they are expelled from this family, these techniques cannot parse them; consequently, any possible expulsion like this has to be ruled out. On the other hand, assume **F** $\subset$ **E**. If the translators can reduce **E** into **F** so this reduction makes the parsing techniques applicable, then the reduction is obviously highly desirable in practice. To illustrate this practical standpoint even more specifically in terms of multi-natural-language translation, take **F** as the languages offically used in the EU states, and consider **E** as the same family extended by other languages used in these states together with their major dialects. For instance, apart from French as the official language, many French people speak other languages, such as a broad variety of Gallo-Romance languages, including several Oïl and Occitan languages. As obvious, from this viewpoint, the reduction sketched above together with its application-related advantages have its practical importance.

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; the TAČR grant TE01020415; and the BUT grant FIT-S-17-3964. I also thank Jakub Martiško for his help concerning this paper.

References

[1] H. C. M. Kleijn and G. Rozenberg, "On the Generative Power of Regular Pattern Grammars," *Acta Informatica*, vol. 20, no. 4, pp. 391–411, 1983.

[2] A. Meduna, *Formal Languages and Computation*, London: CRC Press, 2014.

[3] *Handbook of Formal Languages, Volumes 1 through 3*, G. Rozenberg and A. Salomaa, Eds. Berlin: Springer, 1997.

[4] J. G. Pace, *Mathematics of Discrete Structures for Computer Science*, London: Springer, 2012.

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

[6] J. Martin, *Intro to Lan and the The of Com, 3rd Edition*, Boston: McGraw-Hill, 2003.

[7] M. Sipster, *Introduction to the Theory of Computation,* 3rd ed., Boston: Cengage Learning, 2012.

[8] M.A. Harrison, *Introduction to Formal Language Theory*, Berkeley: Addison-Wesley, 1978.

[9] A.V. Aho and J.D. Ullman, *The theory of parsing, translation, and compiling, Volume I*, Upper Saddle River, NJ: Prentice-Hall, Inc., 1972.

[10] J.E. Hopcroft, R. Motwani, and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3rd ed., Boston: Pearson, 2006.

A. Meduna[1], R. Krčmář[2], A. Kövári[3], Z. Beníčková[4] Received October 31, 2019

Faculty of Information Technology, Brno University of Technology
Brno, Czech Republic
E–mails: [1]`meduna@fit.vutbr.cz`, [2]`ikrcmar@fit.vutbr.cz`, [3]`ikovari@fit.vutbr.cz`, [4]`ibenickova@fit.vut.cz`