# ON BIAUTOMATA

Ondřej Klíma and Libor Polák*
Department of Mathematics and Statistics, Masaryk University
Kotlářská 2, 611 37 Brno, Czech Republic
Email: {klima,polak}@math.muni.cz

***Abstract***
*We initiate the theory and applications of biautomata. A biautomaton can read a word alternately from the left and from the right. We assign to each regular language $L$ its canonical biautomaton. This structure plays, among all biautomata recognizing the language $L$, the same role as the minimal deterministic automaton has among all deterministic automata recognizing the language $L$. We expect that from the graph structure of this automaton one could decide the membership of a given language to certain significant classes of languages. We present the first result of this kind: a language $L$ is piecewise testable if and only if the canonical biautomaton of $L$ is acyclic. From this result the famous Simon's characterization of piecewise testable languages easily follows.*

## 1. Introduction

Regular languages are recognized, among others, by deterministic automata. A regular language $L$ possesses, up to isomorphism, the unique minimal complete deterministic automaton. There is a canonical construction due to Brzozowski [1] where the states are constructed as left derivatives of $L$. A useful property is that each state $q$ is a language and it is exactly the set of all words transforming $q$ into a terminal state. A similar view can be also applied in the theory of universal automata, see Lombardy, Sakarovitch [5] and Polák [7]. Namely, the states of the universal automaton are exactly the finite intersections of left derivatives. This lead the authors to consider the so-called meet automata [3]. In this setting the universal automaton of $L$ can be viewed as the canonical meet automaton for $L$. Other useful structures for a language $L$ are the syntactic monoid and the syntactic semiring of $L$. The syntactic monoid is isomorphic to the transformation monoid of the minimal automaton of $L$. Similarly, the syntactic semiring of $L$ is isomorphic to the transformation semiring of the canonical meet automaton.

One of the major goals in regular language theory is to determine whether a given language is a member of certain significant classes of languages. All the above mentioned structures turned out to be appropriate tools for answering such a kind of questions. In this paper we introduce a new structure, called a biautomaton, and we claim that this structure can also clarify some aspects of these questions. Notice that the term "biautomaton" was used by other authors having quite different meanings. Moreover, our notion is not related to two-ways automata.

*Both authors were supported by the Ministry of Education of the Czech Republic under the project MSM 0021622409 and by the Grant 201/09/1313 of the Grant Agency of the Czech Republic

Motivated by Brzozowski's construction, we consider two-sided derivatives of $L$, to get the states of a new type of automaton. Now each letter determines two independent actions on states, namely the derivative from the left and the derivative from the right. In such a way we get the so-called canonical biautomaton and a natural generalization leads to an abstract notion of biautomata. The canonical biautomaton of the language $L$ plays, among all biautomata recognizing $L$, the same role as the minimal deterministic automaton has among all deterministic automata recognizing $L$.

As the first application of the theory of biautomata we give an effective characterization of piecewise testable languages via their canonical biautomata. The class of piecewise testable languages is a prominent one in the algebraic theory of regular languages. Simon [8, 9] showed that a language is piecewise testable if and only if its syntactic monoid is $\mathcal{J}$-trivial. This characterization is based on one of Green's relations, a basic concept of the semigroup theory. Similar classes of monoids related to other Green's relations are classes of $\mathcal{R}$-trivial monoids and $\mathcal{L}$-trivial monoids, two classes which are right-left dual. It is well known that a finite monoid is $\mathcal{J}$-trivial if and only if it is $\mathcal{R}$-trivial and $\mathcal{L}$-trivial at the same time. Apart from the combinatorial characterization of regular languages having $\mathcal{R}$-trivial syntactic monoids, it is known that these languages are those which have acyclic minimal automata (see Section 4.3 in [6] for more details). From this point of view, a language $L$ is piecewise testable if and only if both the minimal automaton of $L$ and the minimal automaton of $\overline{L}$ (the left-right dual language of $L$) are acyclic. Since both these automata can be found in the canonical biautomaton of $L$, this leads us to the claim that the canonical biautomaton of a piecewise testable language is acyclic as well. We show that this is true and that also the opposite implication is valid.

**Theorem 1.** *Let $L$ be a regular language. Then $L$ is piecewise testable if and only if the canonical biautomaton of $L$ is acyclic.*

It is possible to complete the previous arguments into a proof of the theorem as a consequence of known results. Instead of such a proof we show in Section 4 an elementary, direct proof of the theorem. On three pages we give a complete proof which is self-contained. This could demonstrate that there is a certain potential for finding further applications of biautomata in the algebraic theory of regular languages.

After this introductory section we collect necessary definitions and notation in Section 2. The next section is an introduction to the theory of biautomata. The last section characterizes piecewise testable languages in terms of their canonical biautomata. Finally, we derive the original Simon's theorem from our results.

## 2. Preliminaries

We fix a finite non-empty alphabet $A$ consisting of *letters*. Let $A^*$ be the free monoid over $A$ with the neutral element $\lambda$, i.e. the set of all *words* over $A$ equipped with the operation of concatenation. For $u = a_1a_2 \dots a_n \in A^*$ where $n$ is a positive integer and $a_1, a_2, \dots, a_n \in A$, we write $\overline{u} = a_n \dots a_2a_1$, $|u| = n$ and $\mathsf{c}(u) = \{a_1, \dots, a_n\}$ (i.e. the set of all letters occurring in $u$).

Moreover, we put $\overline{\lambda} = \lambda$, $|\lambda| = 0$ and $\mathsf{c}(\lambda) = \emptyset$. Also, for $L \subseteq A^*$, we write $\overline{L} = \{\, \overline{u} \mid u \in L \,\}$ and $L^c = A^* \setminus L$.

A *complete deterministic* finite automaton over $A$ is a fivetuple $\mathcal{A} = (Q, A, \cdot, i, F)$ where

- $Q$ is a nonempty set of *states*,
- $\cdot : Q \times A \to Q$, extended to $\cdot : Q \times A^* \to Q$ by $q \cdot \lambda = q,\ q \cdot (ua) = (q \cdot u) \cdot a$, where $q \in Q$, $u \in A^*$, $a \in A$,
- $i \in Q$ is the *initial* state,
- $T \subseteq Q$ is the set of *final* states.

The automaton $\mathcal{A}$ *accepts* the word $u \in A^*$ if $i \cdot u \in F$. The *right language* $\mathscr{L}(\mathcal{A}, q)$ of a state $q$ with respect to the automaton $\mathcal{A}$ is the set $\{\, w \in A^* \mid q \cdot w \in F \,\}$. The *language recognized* by $\mathcal{A}$ is the set $\mathscr{L}(\mathcal{A}) = \mathscr{L}(\mathcal{A}, i)$.

For a language $L \subseteq A^*$ and $u \in A^*$, we define $u^{-1}L = \{\, w \in A^* \mid uw \in L \,\}$. Moreover, we put $D_L = \{\, u^{-1}L \mid u \in A^* \,\}$. This set is finite for each regular language $L$. Further, let $\mathcal{D}_L = (D_L, A, \cdot, L, F)$, where $q \cdot a = a^{-1}q$, for each $q \in D_L$, $a \in A$, and $q \in F$ iff $\lambda \in q$. This automaton is called the *canonical* automaton for $L$ and it is well-known that it is a minimal complete deterministic automaton for $L$ – see [1].

For a language $L \subseteq A^*$, we define the relation $\sim_L$ on $A^*$ as follows: for $u, v \in A^*$ we have

$$u \sim_L v \text{ if and only if } (\ \forall\, p, r \in A^*\ )\ (\ pur \in L \iff pvr \in L\ ).$$

The relation $\sim_L$ is a congruence on $A^*$, it is called the *syntactic congruence* of $L$ and the quotient structure $\mathsf{M}(L) = A^*/\sim_L = \{\, [u]_{\sim_L} \mid u \in A^* \,\}$ is called the *syntactic monoid* of $L$. Moreover, the monoid $\mathsf{M}(L)$ is finite whenever $L$ is a regular language.

Recall, that a monoid $M$ is $\mathcal{J}$*-trivial* if and only if for all elements $a, b, c, d, e, f \in M$, the equalities $cad = b$, $ebf = a$ implies $a = b$.

## 3. Biautomata

### 3.1. General definition, congruences, quotient biautomaton, isomorphism

A *biautomaton* over a finite alphabet $A$ is a sixtuple $\mathcal{B} = (Q, A, \cdot, \circ, i, T)$ where

- $Q$ is a nonempty set of *states*,
- $\cdot : Q \times A \to Q$, extended to $\cdot : Q \times A^* \to Q$ by $q \cdot \lambda = q,\ q \cdot (ua) = (q \cdot u) \cdot a$, where $q \in Q$, $u \in A^*$, $a \in A$,
- $\circ : Q \times A \to Q$, extended to $\circ : Q \times A^* \to Q$ by $q \circ \lambda = q,\ q \circ (av) = (q \circ v) \circ a$, where $q \in Q$, $v \in A^*$, $a \in A$,
- $i \in Q$ is the *initial* state,

- $T \subseteq Q$ is the set of *final* states,
- for each $q \in Q$, $u \in A^*$, we have $q \cdot u \in T$ if and only if $q \circ u \in T$,
- for each $q \in Q$, $a, b \in A$, we have $(q \cdot a) \circ b = (q \circ b) \cdot a$.

Notice that from the last condition it follows:

$$\text{for each } q \in Q,\ u, v \in A^*,\ \text{ we have } (q \cdot u) \circ v = (q \circ v) \cdot u\,. \tag{$*$}$$

In contrast to deterministic automata, we could not take a finite set of vertices and define actions of letters arbitrarily.

The biautomaton $\mathcal{B}$ *accepts* a given word $u \in A^*$ if $i \cdot u \in T$. This is equivalent to $i \circ u \in T$. In the definition of acceptance we read $u$ from the left-hand side and transform states according to $\cdot$, in the equivalent condition we read $u$ from the right-hand side and transform states according to $\circ$. Moreover, it allows us an *impatient reading*: we can divide $u = u_1 \dots u_k v_k \dots v_1$ arbitrarily, where $u_1, \dots, u_k, v_k, \dots, v_1 \in A^*$, and we read $u_1$ first, then $v_1$, then $u_2$, and so on, i.e. we move from $i$ to the state

$$q = ((\dots((((i \cdot u_1) \circ v_1) \cdot u_2) \circ v_2) \dots) \cdot u_k) \circ v_k\,.$$

Indeed, we show that $q \in T$ if and only if $i \cdot u \in T$. Using $(*)$ repeatedly, we get

$$q = (\dots((((\dots((i \cdot u_1) \cdot u_2) \dots) \cdot u_k) \circ v_1) \circ v_2) \dots) \circ v_k = (i \cdot u_1 u_2 \dots u_k) \circ v_k \dots v_2 v_1\,.$$

Now $q \in T$ if and only if $(i \cdot u_1 u_2 \dots u_k) \cdot v_k \dots v_2 v_1 = i \cdot u \in T$.

The *right language* $\mathscr{L}(\mathcal{B}, q)$ of a state $q$ with respect to the biautomaton $\mathcal{B}$ is the set $\{\, w \in A^* \mid q \cdot w \in T \,\}$. The *language recognized* by $\mathcal{B}$ is the set $\mathscr{L}(\mathcal{B}) = \mathscr{L}(\mathcal{B}, i)$. The state $q \in Q$ of the biautomaton $\mathcal{B}$ is *reachable* if there exist $u, v \in A^*$ such that $q = (i \cdot u) \circ v$.

A relation $\sim$ is a *congruence relation* of the biautomaton $\mathcal{B} = (Q, A, \cdot, \circ, i, T)$ if

- $\sim$ is an equivalence relation on the set $Q$,
- for each $p, q \in Q$, $a \in A$, the fact $p \sim q$ implies that both $p \cdot a \sim q \cdot a$ and $p \circ a \sim q \circ a$,
- for each $p \in T$, $q \in Q$, the fact $p \sim q$ yields $q \in T$.

We define the *quotient* automaton $\mathcal{B}/\!\sim\, = (Q/\!\sim, A, \cdot_\sim, \circ_\sim, i\sim, T/\!\sim)$ where $(q\sim) \cdot_\sim a = (q \cdot a)\!\sim$ and $(q\sim) \circ_\sim a = (q \circ a)\!\sim$. This structure is again a biautomaton. Moreover, it recognizes the same language as $\mathcal{B}$ does.

Two biautomata $\mathcal{B} = (Q, A, \cdot, \circ, i, T)$ and $\mathcal{B}' = (Q', A, \cdot', \circ', i', T')$ are *isomorphic* if there exists a bijection $\varphi : Q \to Q'$, called an *isomorphism*, such that

- for each $q \in Q$, $a \in A$, we have that $\varphi(q \cdot a) = \varphi(q) \cdot' a$ and $\varphi(q \circ a) = \varphi(q) \circ' a$,
- $\varphi(i) = i'$,
- for each $q \in Q$, we have that $q \in T$ if and only if $\varphi(q) \in T'$.

Clearly, isomorphic biautomata recognize the same languages.

### 3.2. Reverse biautomaton

The next construction shows how one can naturally convert a deterministic automaton into a biautomaton recognizing the same language.

Given a complete deterministic automaton $\mathcal{A} = (Q, A, \cdot, i, F)$, we define the structure $\mathcal{A}^{\mathsf{B}} = (Q^{\mathsf{B}}, A, \cdot^{\mathsf{B}}, \circ^{\mathsf{B}}, i^{\mathsf{B}}, F^{\mathsf{B}})$, where

- $Q^{\mathsf{B}} = \{\, (q, P) \mid q \in Q,\ P \subseteq Q \,\}$,
- for each $q \in Q,\ P \subseteq Q$, we have $(q, P) \cdot^{\mathsf{B}} a = (q \cdot a, P),\ (q, P) \circ^{\mathsf{B}} a = (\, q, \{p \in Q \mid p \cdot a \in P\}\,)$,
- $i^{\mathsf{B}} = (i, F)$,
- for each $q \in Q,\ P \subseteq Q$, we have $(q, P) \in F^{\mathsf{B}}$ iff $q \in P$.

**Lemma 1.** *For each complete deterministic automaton $\mathcal{A}$, the structure $\mathcal{A}^{\mathsf{B}}$ is a biautomaton recognizing the same language as $\mathcal{A}$ does.*

*Proof.* Let $q \in Q,\ P \subseteq Q,\ u, v \in A^*$. Then

$$(q, P) \cdot^{\mathsf{B}} u = (q \cdot u, P) \quad \text{and} \quad (q, P) \circ^{\mathsf{B}} u = (\, q, \{\, p \in Q \mid p \cdot u \in P \,\}\,) .$$

Each of the above states is terminal iff $q \cdot u \in P$.
Moreover, $((q, P) \cdot^{\mathsf{B}} u) \circ^{\mathsf{B}} v = (\, q \cdot u, \{\, p \in Q \mid p \cdot v \in P \,\}\,) = ((q, P) \circ^{\mathsf{B}} v) \cdot^{\mathsf{B}} u$.

Finally, we have $\mathscr{L}(\mathcal{A}^{\mathsf{B}}) = \{\, w \in A^* \mid (i, F) \cdot^{\mathsf{B}} w \in F^{\mathsf{B}} \,\} = \{\, w \in A^* \mid (i \cdot w, F) \in F^{\mathsf{B}} \,\} =$
$= \{\, w \in A^* \mid i \cdot w \in F \,\} = \mathscr{L}(\mathcal{A})$ . □

The biautomaton $\mathcal{A}^{\mathsf{B}}$ is called the *reverse biautomaton* of the automaton $\mathcal{A}$.

### 3.3. Product biautomaton

The following construction yields another model for a biautomaton accepting given language $L \subseteq A^*$. For $v \in A^*$, we define

$$Lv^{-1} = \{\, w \in A^* \mid wv \in L \,\}, \quad E_L = \{\, Lv^{-1} \mid v \in A^* \,\},\ P_L = D_L \times E_L .$$

Now we define $\mathcal{P}_L = (P_L, A, \cdot, \circ, (L, L), T)$, where

$$(s, t) \cdot a = (a^{-1}s, t),\ (s, t) \circ a = (s, ta^{-1}) \text{ and } (u^{-1}L, Lv^{-1}) \in T \text{ iff } uv \in L .$$

**Lemma 2.** *The above structure $\mathcal{P}_L$ is a biautomaton isomorphic to the biautomaton of all reachable states of $(\mathcal{D}_L)^{\mathsf{B}}$.*

*Proof.* In $\mathcal{P}_L$ we have $((L,L)\cdot u)\circ v = (u^{-1}L, Lv^{-1})$ and in $(\mathcal{D}_L)^{\mathsf{B}}$ we have $((L,F)\cdot u)\circ v = (\,u^{-1}L, \{w^{-1}L \mid w^{-1}L\cdot v\in F\}\,) = (\,u^{-1}L, \{w^{-1}L \mid (wv)^{-1}L \in F\}\,) = (\,u^{-1}L, \{w^{-1}L \mid wv \in L\}\,) = (\,u^{-1}L, \{w^{-1}L \mid w\in Lv^{-1}\}\,)$.

Therefore the mapping $(u^{-1}L, Lv^{-1}) \mapsto (\,u^{-1}L, \{w^{-1}L \mid w \in Lv^{-1}\}\,)$, $u,v\in A^*$, is correctly defined and it is the desired isomorphism. □

The biautomaton $\mathcal{P}_L$ is called the *product biautomaton* of the language $L$.

### 3.4. Canonical biautomaton

For a language $L \subseteq A^*$ and $u,v\in A^*$, we define

$$u^{-1}Lv^{-1} = \{\, w\in A^* \mid uwv \in L\,\},\quad C_L = \{\, u^{-1}Lv^{-1} \mid u,v\in A^*\,\}\,.$$

We define $\mathcal{C}_L = (C_L, A, \cdot, \circ, L, T)$, where

$$q\cdot a = a^{-1}q,\quad q\circ a = qa^{-1} \quad \text{ and } \quad u^{-1}Lv^{-1}\in T \text{ iff } \lambda \in u^{-1}Lv^{-1}\,.$$

**Lemma 3.** *For each regular language $L$ over $A$, the structure $\mathcal{C}_L$ is a biautomaton. Moreover, for each state $q$, the right language $\mathscr{L}(\mathcal{C}_L, q)$ is equal to $q$. In particular, the biautomaton $\mathcal{C}_L$ recognizes the language $L$.*

*Proof.* Let $u,v \in L$. Realize that each of the states $u^{-1}L\cdot v$ and $u^{-1}L\circ v$ is final iff $uv\in L$. Then $\mathscr{L}(\mathcal{C}_L, u^{-1}Lv^{-1}) = \{\, w\in A^* \mid u^{-1}Lv^{-1}\cdot w \in T\,\} = \{\, w\in A^* \mid \lambda \in (uw)^{-1}Lv^{-1}\,\} = \{\, w \in A^* \mid uwv \in L\,\} = u^{-1}Lv^{-1}$. □

The biautomaton $\mathcal{C}_L$ is called the *canonical biautomaton* of the language $L$.

**Example.** Let $L = \{a,b\}^*ca\{b,c\}^*$ be a language over the alphabet $A=\{a,b,c\}$. In Figure 1, the "reverse" actions by letters are drawn by dashed arrows. We omit here the empty set state and arrows leading there. The initial state $i$ is the language $L$ and the final states are $\{b,c\}^*$, $\{a,b\}^*$ and $1=\{\lambda\}$. The reader could try to read the word $acab$ from the state $i$ in various ways: $i\cdot acab$, $i\circ acab$ or $((i\cdot a)\circ ab)\cdot c$.

### 3.5. Minimalization of biautomata

The minimalization procedure for biautomata is similar to that for deterministic automata:

**Lemma 4.** *Let $\mathcal{B} = (Q, A, \cdot, \circ, i, T)$ be an arbitrary biautomaton where all states are reachable. Then the relation $\sim$ defined on $Q$ by*

$$p\sim q \quad \textit{if and only if} \quad \mathscr{L}(\mathcal{B},p) = \mathscr{L}(\mathcal{B},q)$$

*is a congruence relation on $\mathcal{B}$. Moreover, the mapping*

$$\varphi : ((i\cdot u)\circ v)_{\sim} \mapsto u^{-1}Lv^{-1}$$

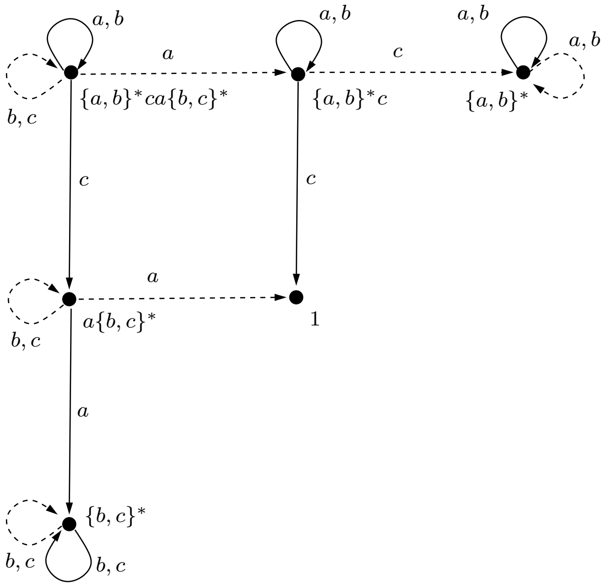

Figure 1: The canonical biautomaton of the language $L = \{a, b\}^*ca\{b, c\}^*$

*is an isomorphism of the quotient biautomaton $\mathcal{B}/\sim$ onto the canonical biautomaton for the language $L = \mathscr{L}(\mathcal{B})$.*

*Proof.* Let $L = \mathscr{L}(\mathcal{B})$. An arbitrary state $p \in Q$ is of the form $p = (i \cdot u) \circ v$, $u, v \in A^*$. Then

$$\mathscr{L}(\mathcal{B}, (i \cdot u) \circ v) = \{\, w \in A^* \mid ((i \cdot u) \circ v) \cdot w \in T \,\} = \{\, w \in A^* \mid ((i \cdot u) \cdot w) \circ v \in T \,\}$$
$$= \{\, w \in A^* \mid (i \cdot uw) \circ v \in T \,\} = \{\, w \in A^* \mid (i \cdot uw) \cdot v \in T \,\} = \{\, w \in A^* \mid i \cdot uwv \in T \,\}$$
$$= \{\, w \in A^* \mid uwv \in L \,\} = u^{-1}Lv^{-1}\,.$$

Thus for $u, v, u', v' \in A^*$, we have

$$p = (i \cdot u) \circ v \sim q = (i \cdot u') \circ v' \text{ if and only if } u^{-1}Lv^{-1} = (u')^{-1}L(v')^{-1}\,.$$

Now, for each $a \in A$, we have $p \cdot a = ((i \cdot u) \cdot a) \circ v = (i \cdot ua) \circ v$ and $p \circ a = (i \cdot u) \circ (av)$ and similarly for $q$. Thus $p \sim q$ yields both $p \cdot a \sim q \cdot a$ and $p \circ a \sim q \circ a$.

Further, $(i \cdot u) \circ v \in T$ iff $i \cdot uv = (i \cdot u) \cdot v \in T$ iff $uv \in L$ iff $\lambda \in u^{-1}Lv^{-1}$. Thus $p \in T$, $p \sim q$ implies $q \in T$.

The second part of our statement follows also from the considerations above. □

## 4. Biautomata for Piecewise Testable Languages

### 4.1. Proof of Theorem 1

A regular language $L$ over an alphabet $A$ is called *piecewise testable* if it is a finite Boolean combination of languages of the form $A^*a_1A^*a_2A^* \dots A^*a_\ell A^*$, where $a_1, \dots, a_\ell \in A$, $\ell \geq 0$. An effective characterization of piecewise testable languages was given by Simon [8, 9] who proved that a language $L$ is piecewise testable if and only if its syntactic monoid is $\mathcal{J}$-trivial. Here we give an alternative effective characterization of piecewise testable languages via biautomata.

For words $u, v \in A^*$ we write $u \triangleleft v$ if and only if $u$ is a subword of $v$, i.e. there are letters $a_1, \dots, a_\ell \in A$ and words $v_0, v_1, \dots, v_\ell \in A^*$ such that $u = a_1 \dots a_\ell$ and $v = v_0 a_1 v_1 \dots a_\ell v_\ell$. For $v \in A^*$, we denote $\mathsf{Sub}_k(v) = \{\, u \in A^+ \mid u \triangleleft v, |u| \leq k \,\}$. We define the equivalence relation $\sim_k$ on $A^*$ by the rule: $u \sim_k v$ if and only if $\mathsf{Sub}_k(u) = \mathsf{Sub}_k(v)$. Note that for $k = 1$ the set $\mathsf{Sub}_k(v)$ is equal to $\mathsf{c}(u)$. Further, for a given word $u \in A^*$ we denote by $L_u$ the language of all words which contain the word $u$ as a subword, i.e. $L_u = \{v \in A^* \mid u \triangleleft v\}$. If $u = a_1 a_2 \dots a_\ell$, where $a_1, a_2, \dots, a_\ell \in A$, then we can write $L_u = A^* a_1 A^* a_2 A^* \dots A^* a_\ell A^*$. An easy consequence of the definition of piecewise testable languages is the following lemma. The proof can be found in e.g. [8],[4]. In fact the proof is so easy that many authors skip it and even in some papers the condition from the lemma is taken as a definition condition for piecewise testable languages.

**Lemma 5.** *A language $L$ is piecewise testable if and only if there exists an index $k$ such that $L$ is a union of classes in the partition $A^*/\sim_k$.*

Our goal is to prove Theorem 1, i.e. the characterization that the piecewise testable languages are exactly languages with the acyclic canonical biautomata. We say that a biautomaton $\mathcal{B} = (Q, A, \cdot, \circ, i, T)$ contains a *cycle* if there exist $n \geq 2$, states $q_0, q_1, \dots, q_n \in Q$, where $q_n = q_0 \neq q_1$, and letters $a_1, \dots, a_n \in A$ such that for each $i = 1, \dots, n$ we have $q_{i-1} \cdot a_i = q_i$ or $q_{i-1} \circ a_i = q_i$. We call a biautomaton $\mathcal{B}$ *acyclic* if $\mathcal{B}$ does not contain any cycle.

**Example (continuation).** The biautomaton in Figure 1 is acyclic and therefore, by our main result, the language $L$ is piecewise testable. In fact, $L = A^*cA^*aA^* \cap (A^*cA^*aA^*aA^*)^{\mathsf{c}} \cap (A^*cA^*cA^*aA^*)^{\mathsf{c}} \cap (A^*cA^*bA^*aA^*)^{\mathsf{c}}$.

**Lemma 6.** *Let $\mathcal{B}$ be a acyclic biautomaton and let $\sim$ be a congruence relation on $\mathcal{B}$. Then the quotient automaton $\mathcal{B}/\sim$ is acyclic.*

*Proof.* Let $(q_0\sim, q_1\sim, \dots, q_n\sim)$ be a cycle in $\mathcal{B}/\sim$ with $q_0\sim, q_1\sim, \dots, q_{n-1}\sim$ pairwise different. We have $n \geq 2$, $q_n \sim q_0 \not\sim q_1$, and $q_0 *_1 a_1 = q_1$, $q_1 *_2 a_2 = q_2, \dots, q_{n-1} *_n a_n = q_n$ where each $*_i$ is $\cdot$ or $\circ$. We can continue $q_n *_1 a_1 = q_{n+1}, \dots, q_{2n-1} *_n a_n = q_{2n}$, $q_{2n} *_1 a_1 = q_{2n+1}, \dots$. Let $q_{kn+i-1}$, $k \geq 0$, $i \in \{1, \dots, n\}$ be the first one which equals to a state already considered. Then we have a cycle in $\mathcal{B}$ starting at $q_{kn}$. □

**Lemma 7.** *Let $L$ be a piecewise testable languages over an alphabet $A$. Then the canonical biautomaton $\mathcal{C}_L = (C_L, A, \cdot, \circ, L, T)$ of $L$ is acyclic.*

*Proof.* Since every piecewise testable language over the alphabet $A$ is a finite Boolean combination of languages $L_u$, it is enough to prove:
i) $\mathcal{C}_{L_u}$ and $\mathcal{C}_{L_u^{\mathsf{c}}}$ are acyclic for every $u \in A^*$;
ii) if $\mathcal{C}_K$ and $\mathcal{C}_L$ are acyclic then both $\mathcal{C}_{K\cap L}$ and $\mathcal{C}_{K\cup L}$ are also acyclic.

For every $u \in A^*$ the canonical biautomaton $\mathcal{C}_{L_u} = (C_{L_u}, A, \cdot, \circ, L, T)$ of the language $L_u$ has states of the form $L_v$, where $v \in A^*$ is a factor of $u$, i.e. $u = pvq$ for some $p, q \in A^*$, in which case we have $p^{-1}L_u q^{-1} = L_v$. Let $v, w \in A^*$ and $a \in A$ be such that $L_v \neq L_w \in C$ and $L_w = L_v \cdot a$ or $L_w = L_v \circ a$. Then $|w| > |v|$ and we can deduce that the biautomaton $\mathcal{C}_{L_u}$ is acyclic. If we consider a language $L_u^c$ instead of $L_u$ then the canonical biautomaton $\mathcal{C}_{L_u^c}$ is acyclic because it is, in fact, the canonical biautomaton $\mathcal{C}_{L_u}$ where just final states are changed.

Now if $K, L$ are languages such that $\mathcal{C}_K$ and $\mathcal{C}_L$ are acyclic then one can consider the direct product of biautomata $\mathcal{C}_K$ and $\mathcal{C}_L$ which is acyclic. In this structure we can choose, in usual way, reachable states and also final states $T_{K\cap L}$ and $T_{K\cup L}$ respectively, namely $(p, q) \in T_{K\cap L}$ iff both $p$ and $q$ are final states in the biautomata $\mathcal{C}_K$ and $\mathcal{C}_L$ and $(p, q) \in T_{K\cup L}$ iff at least one of the states $p, q$ is final. In this way we obtain a certain acyclic biautomaton which recognized the language $K \cap L$ (and $K \cup L$ respectively). To finish the proof we can use Lemmas 4 and 6. □

Now we prove the difficult part of Theorem 1. The basic idea, namely reading one word from left and the other from right, is inspired by our recent combinatorial proof [2] of Simon's result.

**Lemma 8.** *Let L be a regular language such that the canonical biautomaton $\mathcal{C}_L$ of L is acyclic. Then L is a piecewise testable language.*

*Proof.* With respect to Lemma 5 we need to find an appropriate index $k$ such that $L$ is a union of some classes in the partition $A^*/{\sim_k}$. Such $k$ will be 2 times the size of the canonical biautomaton $\mathcal{C}_L = (C_L, A, \cdot, \circ, L, T)$ and the proof will be given by the induction with respect to this $k$.

**Claim.** Let $\mathcal{B} = (B, A, \cdot, \circ, i, T)$ be an arbitrary acyclic biautomaton such that $|B| = \ell$. For every $u, v \in A^*$ such that $\mathsf{Sub}_{2\ell}(u) = \mathsf{Sub}_{2\ell}(v)$ and every $q \in B$, we have $q \cdot u \in T$ iff $q \cdot v \in T$.

*Proof of the claim* : For $\ell = 1$ the statement is trivial. Let $\ell > 1$ be an arbitrary and assume that the statement holds for all smaller numbers. Let $q \in B$ be arbitrary and $u, v \in A^*$ be such that $\mathsf{Sub}_{2\ell}(u) = \mathsf{Sub}_{2\ell}(v)$. We will assume that $q \cdot u \in T$ and $q \cdot v \notin T$ and we show that this assumption leads to a contradiction. Recall that $q \cdot v \notin T$ is equivalent to $q \circ v \notin T$. In the state $q$ we read $u$ from left and $v$ from right and we are interested in the position in the words, where we leave the state $q$. First assume that $q \cdot u = q \in T$, i.e. we do not leave the state $q$. Then $\mathsf{Sub}_{2\ell}(u) = \mathsf{Sub}_{2\ell}(v)$ implies $\mathsf{c}(u) = \mathsf{c}(v)$ and we have $q \cdot v = q \in T$ – a contradiction. Thus $q \cdot u \neq q$ and in the same way we can show that $q \circ v \neq q$. Hence we really leave the state $q$ and there are $u', u'' \in A^*$, $a \in A$ such that $u = u'au''$, for every $c \in \mathsf{c}(u')$ we have $q \cdot c = q$, and $q \cdot a \neq q$. In particular $a \notin \mathsf{c}(u')$. Similarly, let $v', v'' \in A^*$, $b \in A$ be such that $v = v'bv''$, for every $c \in \mathsf{c}(v'')$ we have $q \circ c = q$, and $q \circ b \neq q$. Since $\ell > 1$ we have $\mathsf{c}(u) = \mathsf{c}(v)$ and we can look for the first occurrence of $a$ in the word $v$ and the last occurrence of $b$ in the word $u$. We distinguish three cases depending on relative positions of these occurrences of $a$ and $b$ in $u$. In general, note that for $x, y \in A$, $w \in A^*$ $xy \in \mathsf{Sub}_2(w)$ if and only if the first occurrence of $x$ in $w$ is before the last occurrence of $y$ in $w$. This will be a useful property with respect $\mathsf{Sub}_2(u) = \mathsf{Sub}_2(v)$ which follows from the assumption $\mathsf{Sub}_{2\ell}(u) = \mathsf{Sub}_{2\ell}(v)$.

Case I : *The first occurrence of $a$ in $u$ is before the last occurrence of $b$ in $u$.* Since $\mathsf{Sub}_2(u) = \mathsf{Sub}_2(v)$ the same is true for $v$ and we can consider the following decompositions of $u$ and $v$: $u = u_0 a u_1 b u_2$, $v = v_0 a v_1 b v_2$ where $u_0 = u', u_1, u_2, v_0, v_1, v_2 = v'' \in A^*$ are such that $a \notin \mathsf{c}(u_0)$, $a \notin \mathsf{c}(v_0)$, $b \notin \mathsf{c}(u_2)$, $b \notin \mathsf{c}(v_2)$. If we consider an arbitrary $w \in \mathsf{Sub}_{2\ell-1}(u_1 b u_2)$, then $aw \in \mathsf{Sub}_{2\ell}(u) = \mathsf{Sub}_{2\ell}(v)$ from which $w \in \mathsf{Sub}_{2\ell-1}(v_1 b v_2)$ follows. This means $\mathsf{Sub}_{2\ell-1}(u_1 b u_2) \subseteq \mathsf{Sub}_{2\ell-1}(v_1 b v_2)$ and the opposite inclusion can be proved in the same way. Thus we have $\mathsf{Sub}_{2\ell-1}(u_1 b u_2) = \mathsf{Sub}_{2\ell-1}(v_1 b v_2)$ and similarly $\mathsf{Sub}_{2\ell-1}(u_0 a u_1) = \mathsf{Sub}_{2\ell-1}(v_0 a v_1)$ and $\mathsf{Sub}_{2\ell-2}(u_1) = \mathsf{Sub}_{2\ell-2}(v_1)$.

We denote $q_u = q \cdot u_0 a \neq q$ and we can consider the biautomaton consisting of all states reachable from $q_u$. This is an acyclic biautomaton with at most $\ell - 1$ states, because it is a subset of B and it does not contain the state $q$. By induction assumption $q_u \cdot u_1 b u_2 \in T$ iff $q_u \cdot v_1 b v_2 \in T$. The first condition is satisfied because $q_u \cdot u_1 b u_2 = q \cdot u_0 a u_1 b u_2 = q \cdot u$. Hence $q_u \cdot v_1 b v_2 \in T$ and also $(q_u \circ b v_2) \cdot v_1 = (q_u \cdot v_1) \circ b v_2 \in T$. We denote the state $q_u \circ b v_2$ as $p$.

Analogically, we denote $q_v = q \circ b v_2 = (q \circ v_2) \circ b \neq q$ and we consider the acyclic biautomaton consisting of all states reachable from $q_v$. We have $q_v \circ v_0 a v_1 = q \circ v \notin T$ hence $q_v \circ u_0 a u_1 \notin T$ follows from the induction assumption. Since we work with the biautomaton we deduce that $q_v \cdot u_0 a u_1 = (q_v \cdot u_0 a) \cdot u_1 \notin T$. Now we can see that $q_v \cdot u_0 a = (q \circ b v_2) \cdot u_0 a = (q \cdot u_0 a) \circ b v_2 = q_u \circ b v_2 = p$. We have observed $p \cdot v_1 \in T$ in the previous paragraph and $p \cdot u_1 \notin T$ here. It is clear that $p \neq q$ and we can consider the biautomaton consisting of all states reachable from $p$ which has at most $\ell - 1$ states. Since $\mathsf{Sub}_{2\ell-2}(u_1) = \mathsf{Sub}_{2\ell-2}(v_1)$ we see that both $p \cdot v_1 \in T$ and $p \cdot u_1 \notin T$ cannot be true at the same moment. We obtain a contradiction.

Case II : *The first occurrence of $a$ in $u$ is the last occurrence of $b$ in $u$ at the same time.* In other words, $a = b$ and the first occurrence of $a$ is the unique occurrence of this letter in $u$. Hence $a \in \mathsf{c}(u) = \mathsf{c}(v)$, $aa \notin \mathsf{Sub}_2(u) = \mathsf{Sub}_2(v)$ and $a$ has the unique occurrence in $v$ too. In the same manner as in Case I we can deduce that $\mathsf{Sub}_{2\ell-1}(u') = \mathsf{Sub}_{2\ell-1}(v')$ and $\mathsf{Sub}_{2\ell-1}(u'') = \mathsf{Sub}_{2\ell-1}(v'')$. In particular $\mathsf{c}(u') = \mathsf{c}(v')$ and $\mathsf{c}(u'') = \mathsf{c}(v'')$ which give $q \cdot v' = q$ and $q \circ u'' = q$. Now $q \cdot u = (q \cdot u'a) \cdot u'' \in T$ implies $(q \cdot u'a) \circ u'' \in T$, and thus $(q \cdot u'a) \circ u'' = (q \circ u'') \cdot u'a = q \cdot u'a = q \cdot a$. In the same way $q \circ v = (q \circ av'') \circ v' \notin T$ implies $(q \circ av'') \cdot v' \notin T$, and thus $(q \circ av'') \cdot v' = (q \cdot v') \circ av'' = q \circ av'' = (q \circ v'') \circ a = q \circ a$. We get $q \cdot a \in T$ and $q \circ a \notin T$ which is not possible in biautomata – a contradiction.

Case III : *The first occurrence of $a$ in $u$ is after the last occurrence of $b$ in $u$.* This means that $ab \notin \mathsf{Sub}_2(u) = \mathsf{Sub}_2(v)$ and the first occurrence of $a$ in $v$ is after the last occurrence of $b$ in $v$. We can consider the following decompositions of $u$ and $v$: $u = u_0 b u_1 a u_2$, $v = v_0 b v_1 a v_2$ where $u_0, u_1, u_2, v_0, v_1, v_2 \in A^*$ are such that $u_0 b u_1 = u'$, $v_1 a v_2 = v''$. Again we can deduce that $\mathsf{Sub}_{2\ell-1}(u_0) = \mathsf{Sub}_{2\ell-1}(v_0)$ and $\mathsf{Sub}_{2\ell-1}(u_2) = \mathsf{Sub}_{2\ell-1}(v_2)$, in particular $\mathsf{c}(u_0) = \mathsf{c}(v_0)$ and $\mathsf{c}(u_2) = \mathsf{c}(v_2)$. Hence for every $c \in \mathsf{c}(u') = \mathsf{c}(u_0 b u_1)$ we have $q \cdot c = q$, in particular $q \cdot b = q$ and $q \cdot c = q$ for every $c \in \mathsf{c}(u_0) = \mathsf{c}(v_0)$. Now we see that $q \circ v = q \circ v_0 b v_1 a v_2 = (q \circ v_1 a v_2) \circ v_0 b = q \circ v_0 b \notin T$. Hence $q \cdot v_0 b \notin T$, $q \cdot v_0 b = q$ and we deduce $q \notin T$. On the other hand, for every $c \in \mathsf{c}(v'') = \mathsf{c}(v_1 a v_2)$ we have $q \circ c = q$, in particular $q \circ a = q$ and $q \circ c = q$ for every $c \in \mathsf{c}(v_2) = \mathsf{c}(u_2)$. Now $q \cdot u = (q \cdot u_0 b u_1) \cdot a u_2 = q \cdot a u_2 \in T$. Hence $q \circ a u_2 \in T$, $q \circ a u_2 = q$.

But this means $q \in T$ which is a contradiction with the previous conclusion $q \notin T$.

We have proved the claim which completes the proof of the lemma. □

Now Theorem 1 is a consequence of Lemmas 7 and 8.

### 4.2. Simon theorem as a consequence of Theorem 1

In 1972, Simon gave the following effective characterization of piecewise testable languages.

**Theorem 2.** *Let $L$ be a language over a finite alphabet $A$. Then $L$ is piecewise testable if and only if the syntactic monoid $\mathsf{M}(L)$ is $\mathcal{J}$-trivial.*

The statement consists of two implications where one of them is easy to prove. Namely, one can easily show that for each word $u \in A^*$ the syntactic monoid of the language $L_u$ is $\mathcal{J}$-trivial. This implies that every piecewise testable language has a $\mathcal{J}$-trivial syntactic monoid. Here we want to show the difficult implication in Simon theorem as a consequence of Theorem 1.

**Lemma 9.** *Let $L$ be a language over a finite alphabet $A$ such that the syntactic monoid $\mathsf{M}(L)$ is $\mathcal{J}$-trivial. Then the canonical biautomaton $\mathcal{C}_L$ of $L$ is acyclic. Therefore $L$ is piecewise testable.*

*Proof.* Let $L$ be a regular language such that $M = \mathsf{M}(L)$ is $\mathcal{J}$-trivial. Recall that each element of $M$ is of the form $[u]_{\sim_L}$, which is denoted simply by $[u]$. We also denote the subset of $M$ corresponding to words from the language $L$ by $F$, i.e. $F = \{\, [u] \in M \mid u \in L \,\}$.

For the monoid $M$ we construct the following biautomaton $\mathcal{B} = (Q, A, \cdot, \circ, i, T)$. We put $Q = M \times M$, for every $a \in A$ and $p, r \in M$ we set $(p, r) \cdot a = (p[a], r)$ and $(p, r) \circ a = (p, [a]r)$. Furthermore, $i = ([\lambda], [\lambda])$ and $T = \{\, (p, r) \mid pr \in F \,\}$. Now one can check that $\mathcal{B}$ is a biautomaton. Moreover, we see that

$$u \in \mathscr{L}(\mathcal{B}) \quad \text{iff} \quad ([\lambda], [\lambda]) \cdot u \in T \quad \text{iff} \quad ([u], [\lambda]) \in T \quad \text{iff} \quad [u] \in F \quad \text{iff} \quad u \in L\,.$$

Hence the constructed biautomaton $\mathcal{B}$ recognizes $L$. We claim that the biautomaton $\mathcal{B}$ is acyclic. Indeed, assume that $\mathcal{B} = (Q, A, \cdot, \circ, i, T)$ contains a *cycle*, i.e we consider states $q_0, q_1, \ldots, q_n \in Q$, where $q_n = q_0 \neq q_1$, and letters $a_1, \ldots, a_n \in A$ such that for each $i = 1, \ldots, n$ we have $q_{i-1} \cdot a_i = q_i$ or $q_{i-1} \circ a_i = q_i$. Assume additionally that $q_1 = q_0 \cdot a_1$. (The case $q_1 = q_0 \circ a_1$ can be done dually.) Then we have $q_0 = (p_0, r_0)$ and $q_1 = (p_0[a_1], r_0)$. Thus $p_0[a_1] \neq p_0$. Now $q_n = q_0$ implies that there are $u, v \in A^*$ such that $(q_1 \cdot u) \circ v = q_0$. Hence $p_0[a_1][u] = p_0$. This is a contradiction to the fact that $p_0[a_1] \neq p_0$ and to the assumption that $M$ is $\mathcal{J}$-trivial.

Finally, $\mathcal{C}_L$ can be obtained as a quotient biautomaton of the biautomaton $\mathcal{B}$ by Lemma 4. Hence $\mathcal{C}_L$ is acyclic by Lemma 6. □

## 5. Conclusion

We initiated the theory and applications of biautomata. We assigned to each regular language $L$ its canonical biautomaton. This structure plays, among all biautomata recognizing the language $L$, the same role as the minimal deterministic automaton of $L$ has among all deterministic automata recognizing $L$. We expect that from the graph structure of this automaton one could decide the membership of a given language to certain significant classes of languages. We presented the first result of this kind: a language $L$ is piecewise testable if and only if the canonical biautomaton of $L$ is acyclic. From this result the famous Simon's characterization of piecewise testable languages easily follows.

**Acknowledgment** The authors would like to express their gratitude to Michal Kunc who actively participated in several discussions on the subject.

## References

[1] BRZOZOWSKI J., Derivatives of regular expressions. J. ACM 11(4) (1964), 481–494

[2] KLÍMA O., Piecewise testable languages via combinatorics on words, in WORDS 2009, journal version Discrete Mathematics to appear, see also `www.math.muni.cz/~klima/Math/publications.html`

[3] KLÍMA O., POLÁK L., On varieties of meet automata, Theoretical Computer Science 407 (2008), 278–289

[4] KLÍMA O., POLÁK L., Hierarchies of piecewise testable languages, International Journal of Foundations of Computer Science, Vol. 21, No. 4 (2010), 517–533

[5] LOMBARDY S., SAKAROVITCH, J.: The universal automaton. In Flum, J., Grödel, E., Wilke, T., eds.: Logic and Automata: History and Perspectives. Volume 2 of Texts in Logic and Games. Amsterdam University Press (2007), 457–504

[6] PIN J. E., Varieties of Formal Languages, North Oxford, London and Plenum, New-York, (1986)

[7] POLÁK L, Syntactic semiring and universal automata, in Proc. of the Seventh International Conference on Developments in Language Theory, Szeged 2003, LNCS Vol. 2710 (2003), 411–422

[8] SIMON I., Hierarchies of events of dot-depth one, Ph.D. thesis, University of Waterloo, 1972

[9] SIMON I., Piecewise testable events, in *Proc. ICALP 1975*, LNCS Vol. 33 (1975), 214–222