Pre-automata as Mathematical Models of Event Flows Recognisers

Boris Novikov, Ivan Perepelytsya, and Grygoriy Zholtkevych

V.N. Karazin Kharkiv National University,
4, Svobody Sqr., Kharkiv, 61077, Ukraine
novikov@univer.kharkov.ua, ivanperepelytsya@gmail.com,
zholtkevych@univer.kharkov.ua

Abstract. The new class of recognisers is introduced and studied in the paper. The models are based on the notion of partial action of a free finite generated monoid. Authors called such models by preautomata. Some properties of preautomata were established and proved in the paper. These properties allow to consider the pre-automata as mathematical models of recognizers of event flows in processes of the interaction of software systems.

Introduction

The experience of software development demonstrates that we have no the means for forecasting of progress of software projects (see [1]). So, in 2009, only 32% of the software development projects were successful. At the same time, the percentage of projects that ended with a significant budget overruns and the disruption of a schedule was 44%, and the projects that were interrupted in the form of loss control costs or timelines - 24% of total software development projects. As we can see, the implementation of different methods in software management, the use of increasingly sophisticated technologies in software development, have not led to significant improvement in the quality of software development processes. The reason for the complexity of the development processes of large software systems is the need to provide correct handling for all possible flows of system events. One of the authors of this article in 1990 noted [12]: "it is almost impossible to foresee the sequence of the information processing procedures for complex computer systems, and therefore impossible to plan the flow of control". Rejection of an identification of all possible control flows can provide scalability and flexibility of software product in a process of system design. Breaking down of monolithic architectures leads us to the concept of data-driven systems [11], in particular - to event-driven architecture (EDA) [4].

Modern applications development tools for EDA are based on using standard methodologies such as "Event Dispatcher - Event Listener". This methodology assumes that, each generated event can be listened by a number of handlers However, in case of an interaction of many systems the dispatchers have to listen flows of events, not only single events. The flow of events forms sensible messages, and in this case, there are no standardized software components even at the level of mathematical models, namely, events listeners are oriented on a recognition of event flows. This work describes one mathematical model of a machine for the event flows recognition. In this mathematical model each event is modelled by the symbol of some alphabet, and messages are modelled by certain words in this alphabet. The pre-automata notion has been introduced. This notion provides a possibility to analyze the event flows to highlight from them reasonable messages that are carried by these flows.

The aim of this paper is to study recognisers which are similar to automata-based recognisers. But we will suggest that a reconiser responses to finite sequences of events. This modification leads to study of partial actions of finite generated free monoids on a set as a recogniser's model.

The notion of a partial action was introduced for groups in [3] and for monoids in [10].

This paper is organised as follows.

In section 1, definitions of the terms are given and basic notation is introduced. Then the key example is considered therein.

In section 2, the relationship between preautomata and automata is studied. The Theorem about Universal Globalisation contains the main result of the section. It substantiates using of pre-automata as models for behaviour of systems in the case of restricted observability of system's states.

In section 3, a class of a languages, which are recognised by a preautomaton, is introduced. We call this class as a class of P-recognisable languages. Then we specify such languages in the terms of right congruences on a free monoid generated by a preautomaton's alphabet.

In section 4, the Eilenberg's Structural Theorem [2, see p. 83] is proved for P-recognisable languages.

In section 5, a capability of preautomata as recognizers is clarified by comparison of the class of P-recognisable languages with other known classes of languages.

In conclusion, the set of problems, which solution gives an answer to question of adequacy using preautomata for modelling behaviour of systems, is formulated.

1 Preliminaries

The notion of a partial action is adopted from [6] as follows.

Definition 1. *Suppose X is an arbitrary set, M is a monoid with unit 1, and $X \times M \dashrightarrow X \colon (x, m) \mapsto x \cdot m$ is a partial map. The triple $(X, M, \cdot)$ is called a partial M-action on X iff the following conditions are held*

$$x \cdot 1 = x \text{ for all } x \in X; \tag{1}$$

$$\begin{aligned} &\text{if } x \cdot m_1 \text{ and } (x \cdot m_1) \cdot m_2 \text{ are defined then } x \cdot (m_1 m_2) \text{ is defined} \\ &\text{and } (x \cdot m_1) \cdot m_2 = x \cdot (m_1 m_2); \end{aligned} \tag{2}$$

$$\begin{aligned}&\textit{if } x \cdot m_1 \textit{ and } x \cdot (m_1 m_2) \textit{ are defined then } (x \cdot m_1) \cdot m_2 \textit{ is defined}\\ &\textit{and } x \cdot (m_1 m_2) = (x \cdot m_1) \cdot m_2.\end{aligned} \tag{3}$$

We write $x \cdot m \neq \emptyset$ if $x \cdot m$ is defined, and $x \cdot m = \emptyset$ if $x \cdot m$ is undefined.

The case of a finite generated free monoid M will be considered in the article only. Therefore we need to reformulate Definition 1.

Definition 2. *Let Q be a set of states, Σ be a finite alphabet, and suppose a partial Σ^*-action on Q is defined then the triple $(Q, \Sigma, \cdot)$ is called a preautomaton.*

As usial for free monoid Σ^* we denote its unit by ϵ.

Example 1. Some class of examples of preautomata can be built in the following way.
Let X be a set, Q be a subset of X, Σ be a finite alphabet, and suppose a Σ^*-action on X is defined. We can build a partial Σ^*-action on Q with respect to the next formula

$$x \cdot w = \begin{cases} \emptyset, & \text{iff } x \cdot w \notin Q \\ x \cdot w, & \text{iff } x \cdot w \in Q \end{cases}$$

when $x \in Q$ and $w \in \Sigma^*$.
It is easy to prove that conditions 1, 2, and 3 of Definition 1 are held. Hence, $\mathcal{P} = (Q, \Sigma, \cdot)$ is a preautomaton.
We can consider the preautomaton $\mathcal{P}$ as a restriction of a deterministic automaton [5] $\mathcal{A} = (X, \Sigma, \cdot)$ on the set Q.

Example 1 describes a general case. It will be demonstrated in the next section.

The following definition makes it possible to consider the class of all preautomata as a category.

Definition 3. *Suppose $\mathcal{P}_1 = (Q_1, \Sigma, \cdot)$ and $\mathcal{P}_2 = (Q_2, \Sigma, \cdot)$ are preautomata, $\psi\colon Q_1 \longrightarrow Q_2$ is a map. The map ψ is called an equivariant map if for each $x \in Q_1$ and $w \in \Sigma^*$ such that $x \cdot w \neq \emptyset$ the following condition is held:*

$$\psi(x) \cdot w \neq \emptyset \textit{ and } \psi(x \cdot w) = \psi(x) \cdot w.$$

The class of all Σ-preautomata with equivariant maps as morphisms is a category [9]. The proof is trivial. We denote this category by $\Sigma\mathbf{PA}$, and by $\Sigma\mathbf{PA}(\mathcal{P}_1, \mathcal{P}_2)$ we denote a set of morphisms from $\mathcal{P}_1$ to $\mathcal{P}_2$ when $\mathcal{P}_1$ and $\mathcal{P}_2$ are preautomata.

As usual [9], we introduce notions of a monomorphism, an epimorphism, and an isomorphism. Note, that an equivariant map is a monomorphism iff it is injective; in the category $\Sigma\mathbf{PA}$ there are bimorphisms which are not isomorphisms.

Definition 4. *We shall say that a preautomaton $(Q, \Sigma, \cdot)$ is a finite preautomaton iff the set Q is finite.*

The class of all finite Σ-preautomata with equivariant maps form a subcategory of the category $\Sigma\mathbf{PA}$. We denote this subcategory by $\Sigma\mathbf{FPA}$.

2 Universal Globalisation of Preautomata

The aim of this section is to prove that each preautomaton is a restriction of some automaton with same alphabet.

Definition 5. *An automaton $\mathcal{A} = (X, \Sigma, \cdot)$ is called a globalization of a preautomaton $\mathcal{P} = (Q, \Sigma, \cdot)$ if there is a monomorphism $\zeta \in \Sigma\mathbf{PA}(\mathcal{P}, \mathcal{A})$.*

At first, for each preautomaton $\mathcal{P} = (Q, \Sigma, \cdot)$ we build a set $Q_{\mathbf{gl}}$ and an injection $\iota\colon Q \longrightarrow Q_{\mathbf{gl}}$.
Put $\overline{Q} = Q \times \Sigma^*$.
For any $q_1, q_2 \in Q$ and $w_1, w_2 \in \Sigma^*$ we shall write $(q_1, w_1) \vdash (q_2, w_2)$ iff for some $u \in \Sigma^*$ is held the following condition: $w_1 = uw_2$ and $\emptyset \neq q_1 \cdot u = q_2$.
Denote by $\simeq$ the least equivalence on $\overline{Q}$ such that the condition $(q_1, w_1) \vdash (q_2, w_2) \Rightarrow (q_1, w_1) \simeq (q_2, w_2)$ is satisfied.
Now, by definition put $Q_{\mathbf{gl}} = \overline{Q}/\simeq$.
Denote by $[q, w]$ the $\simeq$-class of the $(q, w) \in \overline{Q}$.

Lemma 1. *The triple $\mathcal{P}_{\mathbf{gl}} = (Q_{\mathbf{gl}}, \Sigma, \cdot)$ is an automaton, where the action is defined by the formula $[q, w] \cdot a = [q, wa]$, when $q \in Q$, $w \in \Sigma^*$, and $a \in \Sigma$.*

Proof. One can establish this fact by direct checking of automaton's definition. □

Then, express explicitly the condition $(q_1, w_1) \simeq (q_2, w_2)$, where $q_1, q_2 \in Q$, $w_1, w_2 \in \Sigma^*$.

Definition 6. *Suppose $q \in Q$ and $w \in \Sigma^*$, we shall say that they form a canonical piar iff for each $u, v \in \Sigma^*$ such that $uv = w$ the following condition is held $q \cdot u \neq \emptyset \Rightarrow u = \epsilon$.*

We shall use the notation $q \times w$ if q and w form a canonical piar.

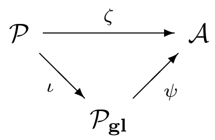
Lemma 2. *Suppose $q_1, q_2 \in Q$ and $w_1, w_2 \in \Sigma^*$ then $(q_1, w_1) \simeq (q_2, w_2)$ iff there exist u_1, u_2, and s in Σ^* such that $w_1 = u_1 s$, $w_2 = u_2 s$, $q_1 \cdot u_1 \neq \emptyset$, $q_2 \cdot u_2 \neq \emptyset$, $q_1 \cdot u_1 = q_2 \cdot u_2$, and for $q' = q_1 \cdot u_1 = q_2 \cdot u_2$ the condition $q' \times s$ is held.*

Proof. Evidently, the conclusion of the Lemma defines some equivalence, which we denote by $\sim$. The assertion $(q_1, w_1) \vdash (q_2, w_2) \Rightarrow (q_1, w_1) \sim (q_2, w_2)$ follows from the definition of $\sim$. Now, one can use the definition of $\simeq$ and check that the condition $(q_1, w_1) \sim (q_2, w_2) \Rightarrow (q_1, w_1) \simeq (q_2, w_2)$ is satisfied. From this assertion and the definition of $\simeq$ it follows that $\sim$ equals $\simeq$. □

Corollary 1. *In each $\simeq$-class there exists an unique canonical pair $(q, w) \in \overline{Q}$.*

By definition, put $\iota(q) = [q, \epsilon]$. Then from the Corallary 1 it follows that the map $\iota\colon Q \longrightarrow Q_{\mathbf{gl}}$ is injective.

Theorem 1 (about Universal Globalisation). *The map* $\iota\colon Q \longrightarrow Q_{\mathbf{gl}}$ *defines a globalisation* $\iota\colon \mathcal{P} \longrightarrow \mathcal{P}_{\mathbf{gl}}$*. It satisfies the following condition:*
for any globalisation $\zeta\colon \mathcal{P} \longrightarrow \mathcal{A}$ *there is a unique morphism* $\psi \in \Sigma\mathbf{PA}(\mathcal{P}_{\mathbf{gl}}, \mathcal{A})$ *such that the diagram*



is commutative.

Proof. First let us prove that ι is an equivariant map. In fact, suppose $q \cdot w \neq \emptyset$ when $q \in Q$ and $w \in \Sigma^*$. Using Lemma 2, we get

$$\iota(q \cdot w) = [q \cdot w, \epsilon] = [q, w] = [q, \epsilon] \cdot w = \iota(q) \cdot w.$$

Hence, ι is an injective morphism, i.e. a monomorphism, and $\iota\colon \mathcal{P} \longrightarrow \mathcal{P}_{\mathbf{gl}}$ is a globalisation.
Let $[q, w]$ be an element of $Q_{\mathbf{gl}}$. Without loss of generality, we can assume that $q \times w$. By definition, put

$$\psi([q, w]) = \zeta(q) \cdot w.$$

By construction, if q and w as above and $u \in \Sigma^*$ then

$$\psi([q, w] \cdot u) = \psi([q, wu]) = \psi([q \cdot wu_1, u_2]) = \\ \zeta(q \cdot wu_1) \cdot u_2 = \zeta(q) \cdot (wu_1u_2) = ((\zeta(q) \cdot w) \cdot u) = \psi([q, w]) \cdot u$$

when $u = u_1u_2$ and $q \times (wu_1)$.
Thus, $\psi \in \Sigma\mathbf{PA}(\mathcal{P}_{\mathbf{gl}}, \mathcal{A})$.
Finally, let q be an element of Q then we have

$$(\psi \circ \iota)(q) = \psi(\iota(q)) = \psi([q, \epsilon]) = \zeta(q) \cdot \epsilon = \zeta(q)$$

Evidently, ψ is unique. This completes the proof. □

Theorem 1 gives us the positive answer to the question "Is any preautomaton a restriction of some automaton?".

Problem 1. Let $\mathcal{P} = (Q, \Sigma, \cdot)$ be a finte preatomaton. Determine the necessary and sufficient existance conditions of a finite globalisation of $\mathcal{P}$.

3 Preacceptors and P-Recognisable Languages

Parsing of texts is the important class of tasks in computer science. Methods for solving these tasks aregrounded on the automata theory. The main concept in the context is the concept of a recognisable set [2, 5]. In this section we shall connect each preautomaton with some language. The class of such languges will be called as the class of P-recognisable language.

We begin with some notation.

Definition 7. *Let $\mathcal{P} = (Q, \Sigma, \cdot)$ be a finite preautomaton. Suppose some element $q_{in} \in Q$ (the initial state) and some subset $T \subset Q$ (the terminal subset) is marked out then a triple $(\mathcal{P}, q_{in}, T)$ is called a preacceptor.*

We shall denote the preacceptor $(\mathcal{P}, q_{in}, T)$ by $\mathcal{P}(q_{in}, T)$.
By definition, put

$$L[\mathcal{P}(q_{in}, T)] = \{w \in \Sigma^* \mid \emptyset \neq q_{in} \cdot w \in T\}, \tag{4}$$

where $\mathcal{P}(q_{in}, T)$ is a preacceptor.

Definition 8. *Let $\mathcal{P}(q_{in}, T)$ be a preacceptor then we shall say that the language $L[\mathcal{P}(q_{in}, T)]$ is recognised by $\mathcal{P}(q_{in}, T)$.*

Now we can define the class of P-recognisable languages.

Definition 9. *Let L be a language over an alphabet Σ. We shall say that the language L is P-recognisable if there exists some preacceptor such that L is recognised by it.*

Our immediate aim is to find necessary and sufficient conditions for a language be a P-recognisable language. To achieve this aim, we need several definitions.

Recall [8] that an equivalence ρ on Σ^* is called a right congruence iff for any $u, v, w \in \Sigma^*$ from $u \,\rho\, v$ it follows $uw \,\rho\, vw$.

Theorem 2. *Let L be a language over an alphabet Σ. It is P-recognisable iff there exists a right congruence on the monoid Σ^* such that L is equal to some finite union of its classes.*

Proof. Suppose, that $\mathcal{P}(q_{in}, T)$ is a preacceptor that it recognises the language L. Denote by $\mathcal{P} = (Q, \Sigma, \cdot)$ the preautomaton such that $\mathcal{P}(q_{in}, T) = (\mathcal{P}, q_{in}, T)$. Let $\iota \colon \mathcal{P} \longrightarrow \mathcal{P}_{\mathbf{gl}}$ be the universal globalisation of $\mathcal{P}$, $Q_{\mathbf{gl}}$ be a set such that $\mathcal{P}_{\mathbf{gl}} = (Q_{\mathbf{gl}}, \Sigma, \cdot)$. By definition, put $T_{\mathbf{gl}} = \{[q, \epsilon] \in Q_{\mathbf{gl}} \mid q \in T\}$ and denote, by $\mathcal{P}_{\mathbf{gl}}([q_{in}, \epsilon], T_{\mathbf{gl}})$ the acceptor $(\mathcal{P}_{\mathbf{gl}}, [q_{in}, \epsilon], T_{\mathbf{gl}})$.
Put $u \,\rho\, v$ iff $[q_{in}, u] = [q_{in}, v]$.
The binary relation ρ on Σ^* is a right congruence. It follows from Lemma 1.
From Corollary 1 it follows that the acceptor $\mathcal{P}_{\mathbf{gl}}([q_{in}, \epsilon], T_{\mathbf{gl}})$ recognises the same language as the preacceptor $\mathcal{P}(q_{in}, T)$. Moreover,

$$[w]_\rho = \{w \in \Sigma^* \mid w = us, \emptyset \neq q_{in} \cdot u \times s, (q_{in}, w) \in [q_{in} \cdot u, s]\}.$$

Hence, $\emptyset \neq q_{in} \cdot w \in T$ iff $(q_{in}, w) \in [q, \epsilon]$ for some $q \in T$.
Summing the reasoning, we get $L = \bigcup_{q \in T} \{w \in \Sigma^* \mid (q_{in}, w) \in [q, \epsilon]\}$, i.e. L is equal to a finite union of ρ-classes.
Conversely, suppose $L = \bigcup_{i=1}^{n} [w_i]_\rho$, where ρ is some right congruence on Σ^*, $w_1, \ldots, w_n \in \Sigma^*$.
By definition, put $Q = \Sigma^*/\rho$, $[u]_\rho \cdot w = [uw]_\rho$.

Evidently, $\mathcal{A} = (\Sigma^*, \Sigma, \cdot)$ is an automaton. Therefore, we can define a preautomaton $\mathcal{P} = (Q, \Sigma, \cdot)$ as the restriction $\mathcal{A}$ on the set Q.
Now, consider the preacceptor $\mathcal{P}([\epsilon]_\rho, T)$, where $T = \{[w_1]_\rho, \ldots, [w_n]_\rho\}$.
If $w \in L$ then $w \, \rho \, w_i$ for some $1 \leq i \leq n$ by assumption, therefore $\emptyset \neq [\epsilon]_\rho \cdot w = [w_i]_\rho \in T$ and w is recognised by $\mathcal{P}([\epsilon]_\rho, T)$.
If w is recognised by $\mathcal{P}([\epsilon]_\rho, T)$ then $\emptyset \neq [\epsilon]_\rho \cdot w \in T$, i.e. $[w]_\rho = [w_i]_\rho$ for some $1 \leq i \leq n$. Hence, $w \in L$.
This completes the proof. □

Corollary 2. *Let L_1 and L_2 be P-recognasible languages over the same alphabet then $L_1 \bigcap L_2$ is a P-recognisable language too.*

Corollary 3. *The class of P-recognisable languges over a single-letter alphabet equals to the class of recognisable languages over the same alphabet.*

4 Structure of P-Recognisable Languages

In this section we shall prove that the structure of P-recognisable languages is similar to the structure of recognisable languages [2].

Lemma 3. *Let L be a P-languages then $L = \bigcup_{i=1}^{n} L_i$, where*

$$L_i \bigcap L_j = \emptyset \text{ for } 1 \leq i \neq j \leq n; \tag{5}$$

each L_i is recognised by a preacceptor that its terminal subset is an unit set. (6)

Proof. Let $\mathcal{P}(q_{in}, T)$ be a preacceptor that recognises the language L. Suppose $T = \{q_1, \ldots, q_n\}$ then denote by L_i the language recognised by $\mathcal{P}(q_{in}, \{q_i\})$, where $1 \leq i \neq j \leq n$. By construction, properties (5) and (6) are satisfied. □

Let us remember [2, 8]

1. let L be a subset of Σ^*, and u be a word over Σ^* then $u^{-1}L = \{w \in \Sigma^* \mid uw \in L\}$;
2. a language $L \subset \Sigma^*$ is unitary if for any $u_1, u_2 \in L$ it is held $u_1^{-1}L = u_2^{-1}L$;
3. a language $L \subset \Sigma^*$ is a prefix code iff for any $u, v \in \Sigma^*$ such that $u, uv \in L$ it follows $v = \epsilon$.

Note, if L is a prefix code then from $\epsilon \in L$ it follows $L = \{\epsilon\}$.

Lemma 4. *Let L be a language over an alphabet Σ then L is unitary iff L is recognised by a preacceptor such that its terminal subset is a unit set.*

Proof. Let L be a unitary language then there exists an acceptor $\mathcal{A}(q_{in}, \{q_{accept}\})$ which recognises the language L [2]. Denote by $\mathcal{P}(q_{in}, \{q_{accept}\})$ the restriction of $\mathcal{A}(q_{in}, \{q_{accept}\})$ on the set $\{q_{in}, q_{accept}\}$ then the preacceptor $\mathcal{P}(q_{in}, \{q_{accept}\})$

recognises the language L.
Conversely, suppose L is recognised by some preacceptor $\mathcal{P}(q_{in}, \{q_{accept}\})$, and $\mathcal{P}_{gl}([q_{in}, \epsilon], \{[q_{accept}, \epsilon]\})$ is its universal globalisation.
The acceptor $\mathcal{P}_{gl}([q_{in}, \epsilon], \{[q_{accept}, \epsilon]\})$ recognises the language L. Using results of [2, Prop. 1.1], one can get that L is an unitary language. □

Theorem 3 (about Structure of P-recognisable Languages). *Let L be a P-recognisable language then $L = \bigcup_{i=1}^{n} E_i B_i^*$, where E_i, B_i are prefix codes for $i = 1, \dots n$, and $E_i B_i^* \bigcap E_j B_j^* = \emptyset$ for $1 \le i \ne j \le n$.*

Proof. Indeed, from Lemma 3 and Lemma 4 follows that $L = \bigcup_{i=1}^{n} L_i$, where each L_i is a unitary language, and $L_i \bigcap L_j = \emptyset$ if $i \ne j$. In [2, Prop. 3.4] it has been proved that any unitary language has the representation EB^*, when E, B are prefix codes. This completes the proof. □

Problem 2. Describe the class of languages with structure as in Theorem 3 which are P-recognisable.

5 Preautomata Recognition Capability

In this section we compare the class of P-recognisable language with other classes of languages [7]: the class of recognisable languages, the class of context free languages, the classes of recursive and recursively enumerable languages.

At first, compare the class of P-recognisable language with the class of recognisable languages.

Proposition 1. *Any recognisable language is P-recognisable.*

Proof. It is trivial. □

Others cases of a comparison are more complicated.

Example 2. As known [7], $E_1 = \{a^n b^n \mid n > 0\} \subset \{a, b\}^*$ is a context free language. It is evident that E_1 is a prefix code. From Lemma 4 it follows that E_1 is P-recognisable.

We need to improve Theorem 2.

Definition 10. *Let L be a language over an alphabet Σ, u, v be words over Σ. We shall use notation $u \, \rho_L \, v$ iff for any $w \in \Sigma^*$ it is satisfied $uw \in L \Leftrightarrow vw \in L$. In this case, we shall call ρ_L a right syntactic congruence induced by L.*

It is evident that ρ_L is a right congruence on Σ^*.

Proposition 2. *Let L be a language over an alphabet Σ. It is P-recognisable iff L is a finite union of ρ_L-classes.*

Proof. It follows from Theorem 2 and properties of right syntactic congruences [8, p. 27]. □

Example 3. Let L_1 be a language that is formed by all palindromes over the alphabet $\{a, b\}$. Note, that L_1 is a context free language [7]. But it is easy to see, that it is not held $a^m \, \rho_{L_1} \, a^n$ for $0 < m < n$. Therefore, $L_1 \supset \bigcup\limits_{n>0} [a^n]_{\rho_{L_1}}$ and L_1 is not P-recognisable.

Example 4. Let E_2 be a language over the alphabet $\{a, b, c\}$. Suppose $E_2 = \{a^n b^n c^n \mid n > 0\}$. It is evident, that E_2 is a prefix code, therefore it is P-recognisable. But well known [7], E_2 is not a context free language.

Example 5. Let L_2 be a language over the alphabet $\{a\}$. Suppose $L_2 = \{a^{n^2} \mid n > 0\}$. Evidently, L_2 is a recursive language. It is easy to see, that L_2 is not P-recognisable.

In contrast to recognisable languages, there exist a P-recognisable language which is not a recursively enumerable language. Unfortunally, our proof is not constructive.

Proposition 3. *There exists a P-recognisable language which is not recursively enumerable.*

Proof. The class of a recursively enumerable languages over some finite alphabet is countable. The cardinality of the class of all prefix codes over some finite alphabet equals to the cardinality of continuum. This completes the proof. □

Next proposition establishes that the class of P-recognisable languages is not closed under operations of a Kleene algebra.

Proposition 4. *Let Σ be a finite alphabet such that its power greater than 1, and $\Sigma\mathfrak{PR}$ be the class of P-recognisable languages over Σ then*

$$\text{there exist } L_1, L_2 \in \Sigma\mathfrak{PR} \text{ such that } L_1 \cup L_2 \notin \Sigma\mathfrak{PR} \tag{7}$$

$$\text{there exist } L_1, L_2 \in \Sigma\mathfrak{PR} \text{ such that } L_1 \cdot L_2 \notin \Sigma\mathfrak{PR} \tag{8}$$

$$\text{there exists } L \in \Sigma\mathfrak{PR} \text{ such that } L^* \notin \Sigma\mathfrak{PR} \tag{9}$$

Proof. To prove (7) put $L_1 = \{a^n \mid n > 0\}$, $L_2 = \{a^n b^n \mid n > 0\}$, and $L = L_1 \cup L_2$. Evidently, $L_1, L_2 \in \Sigma\mathfrak{PR}$ and for any $n > 0$ it is satisfied $[a^n]_{\rho_L} \subset L$. But it is not satisfied $a^m \, \rho_L \, a^n$ for $m \neq n$. From Proposition 2 it follows that $L \notin \Sigma\mathfrak{PR}$.
To prove (8) put $L = L_1 \cdot L_2$. Suppose that $1 < m < n$ then it is not satisfied $a^{m+1} b \, \rho_L \, a^{n+1} b$.
Indeed,

$$(a^{m+1} b) \cdot b^m = a^{m+1} b^{m+1} \notin L$$

$$(a^{n+1} b) \cdot b^m = a^{n+1} b^{m+1} = a^{n-m} a^{m+1} b^{m+1} \in L$$

As above, it is easy to see $L \notin \Sigma\mathfrak{PR}$.
To prove (9) put $L = \{a^n b^n \mid n > 0\} \cup \{a\}$. It is easy to see $[ab]_{\rho_L} = \{a^n b^n \mid n > 0\}$ and $[a]_{\rho_L} = \{a\}$, hence $L \in \Sigma\mathfrak{PR}$. As above, it is not satisfied $a^m \, \rho_{L^*} \, a^n$ for $m \neq n$. But $a^n \in L^*$, therefore $L^* \notin \Sigma\mathfrak{PR}$. □

Conclusion

We have introduced the new class of algebraic objects for systems behaviour modelling. Objects of this class are similar to deterministic finite automata. But presented models permit to describe hidden from observer behaviour of a system.

A model of this class can be obtained by a restriction some automaton on a finite subset of its states. An abstract concept to describe such models have been introduced. We call corresponding abstract objects by preautomata.

Theorem about universal globalisation for preautomata has been proved in the article. The theorem states that any preautomaton can be represented by a restriction of some automaton on a finite subset of its states.

Then we studied recognisers which based on preautomata and the corresponding class of languages.

Languages of this class have been called P-recognisable languages. The theorem about structure of P-recognisable languages have been proved.

Finally, the place of P-recognisable languages was determined among other classes of languages.

References

1. CHAOS Summary 2009. Standish Group, CHAOS Report (2009) http://www1.standishgroup.com/newsroom/chaos˙2009.php
2. Eilenberg, S.: Automata, Languages, and Machines. Volume A. Academic Press, New York and London (1974)
3. Exel, R.: Partial actions of groups and actions of semigroups. Proc. Amer. Math. Soc., **126** (1998) 3481–3494
4. Ferg, S.: Event-Driven Programming: Introduction, Tutorial, History. SourceForge (2006) http://eventdrivenprg.sourceforge.net
5. Holcombe, W. M. L.: Algebraic automata theory. Cambridge University Press, Cambridge (1982)
6. Hollings, C. Partial actions of monoids. Semigroup Forum, **75** (2007) 293–316
7. Hopcroft, G.E., Motwani, R., Ullman, J.D. Introduction to Automata Theory, Languages, and Computation (2nd Edition). Addison Wesley Publishing Co., Boston (2000)
8. Lallemant, G.: Semigroups and Combinatorial Applications. John Wiley & Sons, New York (1979)
9. Mac Lane, S.: Categories for the Working Mathematician. Springer, Berlin (1971)
10. Megrelishvili, M., Schröder, L.: Globalization of confluent partial actions on topological and metric spaces. Topology and its Appl., **145** (2004) 119–145
11. Microsoft Developer Framework. Microsoft Corporation (2010) http://msdn.microsoft.com/en-us/libriary/dd819894(VS.85).aspx
12. Zholtkevych, G: Design Principles of CAD for Engineering of a Reusable Technological Fitment [in Russian]. Central Research Institute of Information, Moscow (1990)