Protoautomata as Models of Systems with Data Accumulation

Irina Mikhailova[1] and Boris Novikov[2] and Grygoriy Zholtkevych[2]

[1] Luhansk Taras Shevchenko National University,
Institute of Information Technology, 2, Oboronna Str., 91011, Luhansk, Ukraine

mia_irina@rambler.ru

[2] V.N. Karazin Kharkiv National University,
School of Mathematics and Mechanics, 4, Svobody Sqr., 61022, Kharkiv, Ukraine

{bvnovikov46,g.zholtkevych}@gmail.com

Abstract. In the paper formal models of software systems and their components based on the notion of an abstract machine are discussed. Necessity to model systems with data accumulation sets the problem of study of generalizations of the notion of an abstract automaton. In the paper two generalizations, namely, preautomata and protoautomata, are considered. It is shown that passing from automata via preautomata to protoautomata can be naturally realized using the language and methods of category theory.

Keywords. system modelling, abstract automaton, category of automata, preautomata, category of preautomata, globalization, protoautomaton, category of protoautomaton, reflector, free protoautomaton

Key terms. MathematicalModel, SpecificationProcess, VerificationProcess

1 Introduction

Theory of abstract state machines or abstract automata is widely applied in different areas of Computer Science. While the early applications of automata theory were connected with theory of compilators design (see, for example, [1]), the more recent its applications are focused on the problems of specification and verification of behaviour of software components [3, 11]. Such changing of the object of the theory was marked by R. Milner in [11]: “In the classical theory, rather little attention is paid to the way in which two automata may interact, in the sense that an action by one entails a complementary action by another. This kind of interaction requires us to look at automata in new light; in particular, this interdependency of automata via their actions seems to demand a new approach to behavioural equivalence”.

But the practice of modelling system behaviour based on the automata approach has shown that the approach is inadequate if data accumulation for the correct response is necessary.

Using the concept of partial action of a semigroup on a set [7, 10], we have defined the notion of preautomaton and studied its properties [4, 12]. The further study has shown that preautomata can be used for modelling some aspects of behaviour of systems with a delayed response [13, 14].

In this paper, we consider a more general class of automaton-liked systems — the class of protoautomata. All necessary information from the theory of semigroups, automata theory, and category theory can be found in the monographs [5, 6, 8, 9].

We use the notation $\varphi : A \dashrightarrow B$ for the partial mapping of A to B (unlike the complete mapping $A \to B$). If $\varphi(a)$ is not defined for $a \in A$, we write $\varphi(a) = \varnothing$. The free monoid on the alphabet Σ is denoted by Σ^*, and its unit by ε. All actions and preactions used in the paper are right, as it is common in the automata theory.

2 Preliminaries

We will use the definition of the automaton in the following form (the condition of the finiteness is ignored):

Definition 1. *Given a set X and a free monoid Σ^* over the alphabet Σ, an* **automaton** *is a mapping $X \times \Sigma^* \to X : (x, a) \mapsto xa$ such that for all $x \in X$ and $u, v \in \Sigma^*$*

$$x\varepsilon = x, \tag{1}$$

$$x(uv) = (xu)v. \tag{2}$$

More general concept is the following

Definition 2 (see [4]). *A* **preautomaton** *is such a partial mapping of $X \times \Sigma^* \dashrightarrow X : (x, a) \mapsto xa$, that*

a) the condition (1) *is fulfilled;*
b) if $xu \neq \varnothing$ and $(xu)v \neq \varnothing$, then $x(uv) \neq \varnothing$ and equality (2) *is fulfilled;*
c) if $xu \neq \varnothing$ and $x(uv) \neq \varnothing$, then $(xu)v \neq \varnothing$ and equality (2) *is fulfilled.*

The preautomata over the monoid Σ^* form a category $\mathcal{PA}ut(\Sigma)$; its morphisms are such maps $\varphi : X \to Y$ that

$$(\forall\, a \in \Sigma^*)(\forall\, x \in X)(\; xa \neq \varnothing \Longrightarrow \varnothing \neq \varphi(x)a = \varphi(xa)). \tag{3}$$

The category $\mathcal{A}ut(\Sigma)$ of the automata over Σ is a full subcategory of $\mathcal{PA}ut(\Sigma)$.

Preautomata appear in the following situation. Let Y be an automaton and X an arbitrary nonempty subset of Y. Then a restriction of an action on X is a preautomaton.

Conversely, let $X \times M \dashrightarrow X$ be a preautomaton. The construction which is inverse to restriction is called globalization. More precisely:

Definition 3. *A* **globalization** *of the preautomaton X is an automaton Z with an injection $\iota : X \to Z$ such that for all $a \in \Sigma^*$, $x \in X$*

$$xa \neq \varnothing \implies \varnothing \neq \iota(x)a = \iota(xa),$$
$$\iota(x)a \in \iota(X) \implies xa \neq \varnothing \ \&\ \iota(xa) = \iota(x)a.$$

Obviously, ι is a morphism of $\mathcal{PA}ut(M)$. We also call it a globalization.

Definition 4. *A globalization $\iota : X \to Z$ is called* **universal** *if for any globalization $\iota' : X \to Z'$ there is an unique morphism $\varkappa : Z \to Z'$ such that $\iota' = \varkappa\iota$.*

The following construction gives an universal globalization (obviously unique up to isomorphism) for any preautomaton $X \times \Sigma^* \dashrightarrow X$. Define a relation $\vdash$ on the set $X \times \Sigma^*$:

$$(x, ab) \vdash (xa, b) \iff xa \neq \varnothing. \tag{4}$$

Let $\simeq$ be an equivalence relation generated by $\vdash$, and $X^U = (X \times \Sigma^*)/\simeq$. An equivalence class of $\simeq$ containing a pair (x, a) is denoted by $[x, a]$. For $[x, a] \in X^U$ and $b \in \Sigma^*$, we set $[x, a]b = [x, ab]$. Thus a complete action on X^U is defined.

Theorem 1. *The automaton X^U with a morphism $\iota^U : X \to X^U : x \mapsto [x, \varepsilon]$ is the universal globalization of the preautomaton X.*

Proof. See [4, Theorem 2] □

3 Protoautomata

The main object of this paper is a generalization of the notion of preautomaton:

Definition 5. *A* **protoautomaton** *is a partial mapping $X \times \Sigma^* \dashrightarrow X : (x, a) \mapsto xa$ such that*

a) the condition (1) is fulfilled;
b) if $xu \neq \varnothing$ and $(xu)v \neq \varnothing$, then $x(uv) \neq \varnothing$ and equality (2) is fulfilled.

We will also denote the protoautomaton from this definition simply by X, if it does not cause a confusion.

Example 1. Let S be a free subsemigroup of Σ^* and $\alpha : X \times S \to X$ an automaton. Define a partial mapping $X \times \Sigma^* \dashrightarrow X$ as an extension of α, putting $xu = \varnothing$ for $u \in \Sigma^* \setminus S$; so we get a protoautomaton over Σ^*. Note that in general it is not a preautomaton. In addition, this example shows that the automaton over an infinite alphabet can be represented as a protoautomaton over a two-letter alphabet.

Example 2. Let $X = \{x, y\}$ be a two-element set, L a subset of Σ^*. Define a protoautomaton $X \times \Sigma^* \dashrightarrow X$ putting for $a \neq \varepsilon$

$$xa = \begin{cases} y, & \text{if } a \in L, \\ \varnothing, & \text{if } a \notin L, \end{cases}$$

and $ya = \varnothing$. This example shows that protoautomata recognize all languages.

We denote the category of protoautomata with morphisms defined by the condition (3) by $\mathcal{P}t\mathcal{A}ut(\Sigma)$; clearly, $\mathcal{P}\mathcal{A}ut(\Sigma)$ is its subcategory.

It follows from the theory of partial action of semigroups [6, Theorem 5.7], that a protoautomaton which is not a preautomaton has no globalization. More precisely, for the protoautomaton X we can construct an automaton X^U as in Sec. 2, but in this case the morphism ι^U is not injective in general.

In this situation, the concept of a reflector is useful. We recall its definition [9]:

Definition 6. *A subcategory* D *of a category* C *is called* **reflective** *if with each object* $C \in \mathsf{C}$ *an object* $R_{\mathsf{D}}(C) \in \mathsf{D}$ *is associated (called* D-**reflector** *of the object* C*) and a morphism* $\rho_{\mathsf{D}}(C) : C \to R_{\mathsf{D}}(C)$ *(***reflection morphism***) such that for each* $D \in \mathsf{D}$ *the diagram*

$$\begin{array}{l} C \overset{\rho_{\mathsf{D}}(C)}{\longrightarrow} R_{\mathsf{D}}(C) \\ \downarrow \\ D \end{array}$$

can be extended uniquely to a commutative diagram by some morphism out $\mathrm{Hom}_{\mathsf{D}}(R_{\mathsf{D}}(C), D)$.

It is convenient to use another description of the equivalence $\simeq$:

Lemma 1. *Define a relation* $\sharp$ *on the set* $X \times \Sigma^*$*:*

$$(x, a) \ \sharp \ (y, b) \iff (\exists\ a', b', p \in \Sigma^*)(a = a'p \ \&\ b = b'p \ \&\ xa' = yb' \neq \varnothing).$$

Let $\approx$ *be the equivalence relation generated by* $\sharp$. *Then* $\approx$ *coincides with* $\simeq$.

Proof. If $(x, a) \ \sharp \ (y, b)$ then

$$(x, a) = (x, a'p) \vdash (xa', p) = (yb', p) \dashv (y, b'p) = (y, b),$$

whence $\approx \ \subseteq \ \simeq$.

Conversely, if $(x, a) \vdash (y, b)$, then $a = cb, y = xc$ for some $c \in \Sigma^*$. Hence $\vdash \ \subseteq \ \sharp$. Consequently, $\approx \ \supseteq \ \simeq$ $\square$

Remark 1. Obviously, $\vdash$ is reflexive and transitive, while $\sharp$ is reflexive and symmetric.

Lemma 2. *Let* X *be a protoautomaton,* Y *be a preautomaton (both over* Σ^**),* $\alpha : X \to Y$ *be a morphism,* $x, y \in X$, $a \in \Sigma^*$. *Then* $[x, \varepsilon] = [y, a]$ *implies* $\alpha(x) = \alpha(y)a \neq \varnothing$..

Proof. It follows from the condition that

$$(x, \varepsilon) \ \sharp \ (z_1, b_1) \ \sharp \ \ldots \ \sharp \ (z_n, b_n) \ \sharp \ (y, a)$$

for some $z_1, \ldots, z_n \in X$, $b_1, \ldots, b_n \in \Sigma^*$.

Apply induction on n. Since $x = z_1 b_1 \neq \varnothing$ then $\alpha(x) = \alpha(z_1)b_1$. Suppose that $\alpha(x) = \alpha(z_n)b_n \neq \varnothing$. By definition of the relation $\sharp$

$$b_n = cp,\ a = dp,\ z_n c = yd \neq \varnothing$$

for some $c, d, p \in \Sigma^*$. Then $\alpha(z_n)c \neq \varnothing$ and by the induction $\alpha(z_n)(cp) \neq \varnothing$. Since Y is a preautomaton then

$$\alpha(x) = \alpha(z_n)(cp) = \alpha(z_n c)p = \alpha(yd)p = (\alpha(y)d)p = \alpha(y)a.$$

Proof has completed □

Similarly (and even easier) one can prove

Lemma 3. *Let X be a protoautomaton, Y be an automaton (both over Σ^*), $\alpha : X \to Y$ be a morphism, $x, y \in X$, $a, b \in \Sigma^*$. Then $[x, a] = [y, b]$ implies $\alpha(x)a = \alpha(y)b$.*

Proof is omitted □

We set $[X, \varepsilon] = \{[x, \varepsilon] \in X^U \mid x \in X\}$. Obviously, $[X, \varepsilon]$, being a subset of X^U, is a preautomaton, and in addition, $\iota^U(X) = [X, \varepsilon]$.

Theorem 2. *Let X be a protoautomaton over Σ^*, then*

1. *$[X, \varepsilon]$ is a reflector for X in the category $\mathcal{PA}ut(\Sigma)$,*
2. *X^U is a reflector for X in $\mathcal{A}ut(\Sigma)$,*
3. *X^U is a reflector for $[X, \varepsilon]$ in $\mathcal{A}ut(\Sigma)$.*

Proof. 1) Let Y be some preautomaton and $\alpha : X \to Y$ be a morphism of protoautomata. The required morphism $\beta : [X, \varepsilon] \to Y$ is uniquely determined from the equality $\alpha = \beta\iota^U$. Indeed, for $x \in X$ we have $\alpha(x) = \beta\iota^U(x) = \beta([x, \varepsilon])$. It follows from Lemma 2 that β is well-defined.
2) Similarly, using Lemma 3.
3) Follows from 1), 2), and the following well-known fact [9]:
If $\mathsf{A} \subset \mathsf{B} \subset \mathsf{C}$ are categories, A is reflective in B, and B is reflective in C, then A is reflective in C. Moreover, the reflection morphism from C to A is the product of the corresponding reflection morphisms from C to B and from B to A □

Corollary 1. *$\mathcal{A}ut(\Sigma)$ is a reflective subcategory of $\mathcal{PA}ut(\Sigma)$. Moreover, the universal globalization of a preautomaton is its reflector.*

Example 3. Let $X = \{x, y, z, t\}$, $p, u, v \in \Sigma^* \setminus \{\varepsilon\}$. We set $zu = zv = t$, $z(up) = x$, $z(vp) = y$ and $sw = \varnothing$ for all $s \in X$, $w \in \Sigma^* \setminus \{\varepsilon, p, u, v\}$. In such a manner X turns into a protoautomaton. Since $(x, \varepsilon) \sharp (z, up) \sharp (z, vp) \sharp (y, \varepsilon)$ then $[x, \varepsilon] = [y, \varepsilon]$ and the reflection morphism of X is non-injective.

A large class of protoautomata is contained in the following example.

Example 4. Consider a preautomaton $X \times \Sigma^* \dashrightarrow X$ as a directed weighted multigraph with states as vertices and with edges of the form (x, u, y), where $x, y \in X$, $u \in \Sigma^*$, and $y = xu$. Let U be an arbitrary subset of edges of X. Build a transitive closure U^t of the set U, extending it step by step by the rule: if the edges (x, u, y) and (y, v, z) are at some stage in the expansion, then on the next step we include the edge (x, uv, z). Then U^t is a protoautomaton.

Example 3 shows that there exists a protoautomaton such that it can not be embedded into some preautomaton (and thus into some automaton).

4 Free Protoautomata

It is well known [2] that free automata play a significant role in the theory of automata (for example, in the problem of constructing a minimal realization). Therefore, it is advisable to consider the question about the existence of free objects in the category of protoautomata.

Recall the necessary definitions:

Definition 7. *Let* C *and* D *be categories,* $F : \mathsf{C} \rightsquigarrow \mathsf{D}$ *be a functor,* C *be an object of* C. *An object* D *of* D *is called* **free on C with respect to the functor** F*, if there is a morphism* $\alpha : C \to FD$ *such that for any object* $D' \in \mathsf{D}$ *and any morphism* $\beta : C \to FD'$ *there exists the unique morphism* $\gamma : D \to D'$ *such that*

$$F(\gamma)\alpha = \beta. \tag{5}$$

We consider a category $\mathcal{R}el(\Sigma)$ whose objects are pairs (X, ρ), where X is a set ($X \in \mathcal{S}et$), $\rho \subset X \times \Sigma^*$ is a binary relation such that $X \times \{\varepsilon\} \subset \rho$. A morphism $\phi : (X, \rho) \to (Y, \sigma)$ of $\mathcal{R}el(\Sigma)$ is a map $\phi : X \to Y$ such that $(\phi x, u) \in \sigma$ for $(x, u) \in \rho$.

Next, let F be a forgetful functor $F : \mathcal{P}t\mathcal{A}ut(\Sigma) \rightsquigarrow \mathcal{R}el(\Sigma)$ mapping each protoautomaton $X \times \Sigma^* \dashrightarrow X$ to the pair (X, ρ) with $\rho = \{(x, u) \mid xu \neq \varnothing\}$.

Theorem 3. *For each object* $(X, \rho) \in \mathcal{R}el(\Sigma)$ *there is a protoautomaton that is free on it with respect to the forgetful functor* F.

Proof. For $(X, \rho) \in \mathcal{R}el(\Sigma)$ construct a protoautomaton $M = (\rho \times \Sigma^* \dashrightarrow \rho)$, defining the action by the rule

$$(x, u)v = \begin{cases} (x, uv), & \text{if } (x, uv) \in \rho \\ \varnothing, & \text{if } (x, uv) \notin \rho. \end{cases}$$

Then $FM = (\rho, \hat{\rho})$, where $\hat{\rho} = \{((x, u), v) \mid (x, u)v = (x, uv)\} \subset \rho \times \Sigma^*$. Define the morphism $\alpha : (X, \rho) \to (\rho, \hat{\rho})$ by the formula $\alpha(x) = (x, \varepsilon)$.

Let us show that M is a free protoautomaton on (X, ρ).
Let $N = (Y \times \Sigma^* \dashrightarrow Y)$ be some protoautomaton and $FY = (Y, \sigma)$. For the required morphism $\gamma : M \to N$ of (5) we have:

$$\gamma(x, \varepsilon) = F(\gamma)(x, \varepsilon) = F(\gamma)\alpha(x) = \beta(x).$$

Then for any $u \in \Sigma^*$ one can obtain

$$\gamma(x,u) = \gamma(x,\varepsilon)u = \beta(x)u,$$

i.e. γ is uniquely determined □

5 Conclusion

It seems that the class of protoautomata, which has been introduced in the paper, gives the most abstract models for systems with discrete behaviour. This class of abstract machines includes not only machines reacting on the received data immediately, as automata, but it also includes machines whose reactions depend on the accumulated information.

The machines of this class having a greedy behaviour are united into a subclass whose instances are called preautomata. Machines of the subclass are used for modelling behaviour systems for complex event processing as it was shown earlier [13, 14]. This class of machines, in contrast to the class of automata, is closed under structural decomposition, and hence, is more suitable for specifying complex systems. But the condition c) in the definition of a preautomaton (see Definition 2) seems unnatural. This condition also impedes definition of a nondeterministic preautomaton.

Therefore, by eliminating the condition c) we provide a possibility to study nondeterministic models. In our opinion, the models derived in this way (protoautomata) are interesting objects that can be used for specification and verification of complex systems.

References

1. Aho, A.V., Ullman, J.D.: Theory of Parsing, Translation, and Compiling. Prentice-Hall, New York (1972)
2. Arbib, M.A., Manes, E.G.: Machines in a category: an expository introduction. SIAM Rev. 16, 163–192 (1974).
3. Börger, E., Stärk, R.: Abstract State Machines: A Method for High-Level System Design and Analysis. Springer-Verlag, Berlin Heidelberg (2003)
4. Dokuchaev, M., Novikov, B., Zholtkevych, G.: Partial actions and automata. Alg. and Discr. Math. Vol. 11, 2, 51–63 (2011)
5. Eilenberg, S.: Automata, Languages, and Machines, vol. B. Academic Press, New York (1976)
6. Holcombe, W.M.L.: Algebraic Automata Theory. Cambridge Univ. Press (1982)
7. Hollings, C.: Partial actions of monoids. Semigroup Forum. 75, 293–316 (2007)
8. Lallement, G.: Semigroups and combinatorial applications. John Wiley, New York (1979)
9. MacLane, S.: Categories for the Working Mathematician. Springer, Berlin (1971)
10. Megrelishvili, M., Schröder, L.: Globalization of confluent partial actions on topological and metric spaces. Topol. and Appl. 145, 119–145 (2004)
11. Milner, R.: Communicating and Mobile Systems: The Pi Calculus. Cambridge University Press, Cambridge (1999)

12. Novikov, B., Perepelytsya, I., Zholtkevych, G. Pre-automata as mathematical models of event flows recognisers. In: V. Ermolayev et al. (eds.) Proc. 7-th Int. Conf. ICTERI 2011, 41–50 (2011)
13. Perepelytsya, I., Zholtkevych, G.: On some class of mathematical models for static analysis of critical-mission asynchronous systems. Syst. ozbr. ta viysk. tehn. Vol. 27, 3, 60–63 (2011)
14. Perepelytsya, I., Zholtkevych, G.: Hierarchic Decomposition of Pre-machines as Models of Software System Components. Syst. upravl. navig. i zv'iazku. Vol. 20, 4, 233–238 (2011)