

Realisation of “Black Boxes” Using Machines

Grygoriy Zholtkevych

Department of Theoretical and Applied Computer Science,
V.N. Karazin Kharkiv National University
4 Svobody Sqr, 61022, Kharkiv, Ukraine
g.zholtkevych@karazin.ua

Abstract. Modern engineering solutions attract attention of researchers to well-known problems in the field of system theory and cybernetics in general. The realisation problem of a “black box” is one among these problems. In this paper the non-anticipation property for a “black box” is generalised to the case of “black boxes”, whose behaviour admits deferred decisions. Furthermore, for such “black boxes” it is shown that they can be realised as pre-machines, which have been introduced by author jointly with his co-authors in series of earlier papers.

Keywords: “black box”, deferred responses, sequential processing, pre-machine, transfer function

Key Terms: Computation, Software Component, Specification Process, Mathematical Model

1 Introduction

Let us suppose that two finite alphabets X and Y are given. The first of them we identify as the alphabet of stimuli and the second one as the alphabet of responses. Following to the general cybernetic concept [1, Chapter 6] we can consider any mapping $M: X^\omega \rightarrow Y^\omega$ as the transfer function of some “black box”, whose inputs belong to the set X^ω of infinite sequences of stimuli and outputs belong to the set Y^ω of finite or infinite sequences of responses. The realisation problem of such a mapping using a machine is the principal problem that is solved by a system engineer. In other words a system engineer transforms a “black” box into a “white box” or “glass box”.

The realisation problem had been studied in detail (see, for example, [6]) for the mapping $M: X^\omega \rightarrow Y^\omega$ holding the following non-anticipation property¹

$$\begin{aligned} \text{if } M(\mathbf{u}\mathbf{x}') = \mathbf{y}', M(\mathbf{u}\mathbf{x}'') = \mathbf{y}'' \text{ for some finite sequence of stimuli} \\ \mathbf{u} = u_1u_2\dots u_n \text{ and } \mathbf{x}', \mathbf{x}'' \in X^\omega \text{ then } \mathbf{y}' = \mathbf{v}\mathbf{z}' \text{ and } \mathbf{y}'' = \mathbf{v}\mathbf{z}'' \text{ for } (1) \\ \text{some finite sequence of responses } \mathbf{v} = v_1v_2\dots v_n \text{ and } \mathbf{z}', \mathbf{z}'' \in Y^\omega. \end{aligned}$$

In this case there exists a Moore machine whose transfer function coincides with the mapping M [4, 6].

¹ This property informally means that a “black box” cannot use an information from the future.

We should note the following: the previous formulation for the non-anticipation property implicitly implies that the “black box” responds immediately on each stimulus. However there are systems having another reaction type. It is quite possible such a system behaviour that requires to defer a response for as long as the sufficient amount of the information will be received. For example, complex event processing systems (see [7]) have such a reaction type. Therefore processes of the specification and analysis for such systems require another models or at least models, which generalise already existing ones. This paper is an attempt to solve the realisation problem for “black boxes” with transfer function that satisfies the generalisation being defined below of the non-anticipation property.

2 Prerequisites and Notation

The aim of this section is to give brief survey of some matters and explain the basic notation used below.

At the paper we use the denotation \mathbb{N} for the natural series with 0.

For a set X (it is usually finite) we use the notation:

X^* denotes the set of all finite sequences (words) whose elements belong to X ;

ε denotes the empty word;

X^+ denotes the set $X^* \setminus \{\varepsilon\}$;

X^ω denotes the set of all (infinite) sequences whose elements belong to X ;

X^∞ denotes the union of the sets X^* and X^ω .

Further, we use the denotation $|\mathbf{u}|$ for the length of the word $\mathbf{u} \in X^*$ and assume that $|\mathbf{x}| = +\infty$ for any infinite sequence $\mathbf{x} \in X^\omega$.

To refer to the k -th member of a word $\mathbf{u} \in X^*$ (or a sequence $x \in X^\omega$) the denotation $u[k]$ (or $x[k]$ respectively) is used.

For a word $\mathbf{u} \in X^*$ whose length is equal or greater than n (or a sequence $\mathbf{x} \in X^\omega$) by $\mathbf{u}[1 : n]$ (or $\mathbf{x}[1 : n]$ respectively) we denote the word $u[1] \dots u[n]$ (or $x[1] \dots x[n]$).

Similarly, for a word $\mathbf{u} \in X^*$ whose length is greater than or equal to n (or a sequence $\mathbf{x} \in X^\omega$) by $\mathbf{u}[n :]$ (or $\mathbf{x}[n :]$ respectively) we denote the word $u[n] \dots u[|\mathbf{u}|]$ (or the sequence $x[n]x[n+1] \dots$).

3 Non-anticipation Property

In Sec. 1 we have given the definition of the non-anticipation property for a transfer function from X^ω into Y^ω under condition that the corresponding “black box” reacts on each stimulus. Our nearest goal is to generalise the previous definition for the case when a “black box” is capable to decide whether the accumulated information is sufficient for the correct response and generates the response if the decision positive otherwise postpones the response generation.

Firstly, it is needed to say that in this case the class of studied transfer functions are being extended up to the class of mappings from X^ω into Y^∞ .

Further, we should specify that the identity of prefixes for streams of stimuli guarantees the identity of prefixes for the corresponding streams of responses. The sequential character of processing streams of stimuli by a “black box” requires that there exists a correspondence between word of stimuli \mathbf{u} (as prefix of the corresponding streams) and length $N(\mathbf{u})$ of the response word (see Fig. 1).



Fig. 1. A sequential “black box”

The following definition is our attempt to present these considerations as a formal specification.

Definition 1. We shall say that the non-anticipation property holds for a mapping $M: X^\omega \rightarrow Y^\infty$ if the following is true:

there exists a function $N: X^* \rightarrow \mathbb{N}$ such that

1. $N(\mathbf{u}) \leq |\mathbf{u}|$ for any $\mathbf{u} \in X^*$;
2. if $\mathbf{u}' \in X^*$ and $\mathbf{u}'' = \mathbf{u}'x$ for some $x \in X$ then

$$N(\mathbf{u}') \leq N(\mathbf{u}'') \leq N(\mathbf{u}') + 1;$$

3. if $\mathbf{x}', \mathbf{x}'' \in \mathbf{u} \cdot X^\omega$ for some $\mathbf{u} \in X^*$ then

$$(M\mathbf{x}') [1 : N(\mathbf{u})] = (M\mathbf{x}'') [1 : N(\mathbf{u})];$$

4. for any $\mathbf{u} \in X^*$ there exist $\mathbf{x}', \mathbf{x}'' \in \mathbf{u} \cdot X^\omega$ such that

$$(M\mathbf{x}') [1 : N(\mathbf{u}) + 1] \neq (M\mathbf{x}'') [1 : N(\mathbf{u}) + 1].$$

(2)

Remark 1. Informally, jump points for the function N introduced in Def. 1 determine response instants of the “black box”. Items 3) and 4) ensure this interpretation.

Remark 2. Item 2) ensures that the “black box” corresponding to a mapping that holds the non-anticipation property generates at most one response at a stimulus.

Remark 3. Item 3) and 4) of Def. 1 guarantee also that the existence of function N for the mapping $M: X^\omega \rightarrow Y^\infty$ implies the uniqueness of N .

Remark 4. One can easily see that if $N(\mathbf{u}) = |\mathbf{u}|$ then Def. 1 and the non-anticipation property given in Sec. 1 specify the same class of mappings.

Now let us consider the partial mapping $\mu: X^+ \dashrightarrow Y$ that is defined as follows

$$\begin{aligned} \mu(\mathbf{u}) \uparrow & \text{ iff } N(\mathbf{u}) = N(\mathbf{u}[1 : |\mathbf{u}| - 1]) \\ \mu(\mathbf{u}) \downarrow & = M(\mathbf{uz})[N(\mathbf{u})] \text{ iff } N(\mathbf{u}) > N(\mathbf{u}[1 : |\mathbf{u}| - 1]). \end{aligned}$$

Item 3) of Def. 1 ensures the uniqueness of determining $\mu(\mathbf{u})$.

Remark 5. Returning to Fig. 1, we note that $\mu(\mathbf{u}) = y[N(\mathbf{u})]$.

To determine the significance of the mapping μ let us consider the following algorithm and proposition.

Require: a sequence of stimuli $\mathbf{x} \in X^\omega$
Ensure: to print the corresponding sequence of responses

```

n = 1
while True :
    while  $\mu(\mathbf{x}[1 : n]) \uparrow$  : n += 1
        print( $\mu(\mathbf{x}[1 : n])$ )
        n += 1

```

Algorithm 1: “Black box” algorithm for a mapping $M: X^\omega \rightarrow Y^\infty$ that holds the non-anticipation property

Proposition 1. For any $\mathbf{x} \in X^\omega$ Algorithm 1 prints the sequence $M\mathbf{x}$.

Proof. Taking into account (2) one can easily see that new response is printed only if $N(\mathbf{x}[1 : n-1]) \neq N(\mathbf{x}[1 : n])$. In this case the printed symbol is $(M\mathbf{x})[n]$. \square

Definition 2. Let $M: X^\omega \rightarrow Y^\infty$ be a mapping that holds the non-anticipation property then the corresponding partial mapping $\mu: X^+ \dashrightarrow Y$ we shall call its reaction function.

Conversely, we can consider a partial mapping $\mu: X^+ \dashrightarrow Y$ and use Algorithm 1 to define the mapping $M: X^\omega \rightarrow Y^\infty$.

Proposition 2. Let $\mu: X^+ \dashrightarrow Y$ be a partial mapping then the correspondence $\mathbf{x} \in X^\omega \mapsto \mathbf{y} \in Y^\infty$, when \mathbf{y} is the sequence printed by Algorithm 1 under handling \mathbf{x} , determines the mapping $M: X^\omega \rightarrow Y^\infty$ that holds the non-anticipation property.

Proof. The key idea of the proof consists in the following recursive construction of the function $N: X^* \rightarrow \mathbb{N}$:

base of recursion: $N(\varepsilon) = 0$;

step of recursion:

$$N(\mathbf{ux}) = \begin{cases} N(\mathbf{u}), & \text{if } \mu(\mathbf{x}) \uparrow \\ N(\mathbf{u}) + 1, & \text{if } \mu(\mathbf{x}) \downarrow \end{cases}.$$

Now, easily seen that the mapping M holds the non-anticipation property. \square

4 Automata and Pre-automata

In this section we remind the definition of automata as the simplest discrete systems that respond on external stimuli by changing their states. Automata are actions of free finitely generated monoids on the state sets from the mathematical standpoint. In [3] authors have introduced the notion of a pre-automaton using a generalisation of the notion of an action known as a partial action. Taking into account that these notions is used below and they are not widely used we include this section to give the information necessary for understanding of the further text.

4.1 Automata

We start our consideration reminding the definition of an automaton.

Definition 3. A triple $\mathcal{A}(X, S_{\mathcal{A}}, \delta_{\mathcal{A}})$ is called an automaton if X is a finite alphabet of stimuli, $S_{\mathcal{A}}$ is a set of states of the automaton, $\delta_{\mathcal{A}}: S_{\mathcal{A}} \times X \rightarrow S_{\mathcal{A}}$ is a mapping, which is called the transition function of the automaton.

An automaton behaviour is determined by a right action of the monoid X^* on the state set $S_{\mathcal{A}}$ [2].

Proposition 3. Let $\mathcal{A}(X, S_{\mathcal{A}}, \delta_{\mathcal{A}})$ be an automaton then the defined recursively defined mapping $\delta_{\mathcal{A}}^*: S_{\mathcal{A}} \times X^* \rightarrow S_{\mathcal{A}}$

$$\delta_{\mathcal{A}}^*(s, \varepsilon) = s \text{ for any } s \in S_{\mathcal{A}}; \quad (3)$$

$$\delta_{\mathcal{A}}^*(s, \mathbf{u}x) = \delta_{\mathcal{A}}(\delta_{\mathcal{A}}^*(s, \mathbf{u}), x) \text{ for } s \in S_{\mathcal{A}}, \mathbf{u} \in X^*, x \in X \quad (4)$$

is a right action of monoid X^* on the set $S_{\mathcal{A}}$.

Proof. To prove the proposition it is sufficient to check the equality

$$\delta_{\mathcal{A}}^*(s, \mathbf{u}'\mathbf{u}'') = \delta_{\mathcal{A}}^*(\delta_{\mathcal{A}}^*(s, \mathbf{u}'), \mathbf{u}'')$$

for any $s \in S_{\mathcal{A}}, \mathbf{u}', \mathbf{u}'' \in X^*$. Checking is a simple exercise in the application of mathematical induction on the length of \mathbf{u}'' . \square

4.2 Pre-automata

The notion of a pre-automaton had been introduced in [3] by replacing the action with the partial action in the definition.

Definition 4. A triple $\mathcal{P}(X, S_{\mathcal{P}}, \delta_{\mathcal{P}}^*)$ is called a pre-automaton if X is a finite alphabet of stimuli, $S_{\mathcal{P}}$ is a set of states of the pre-automaton, $\delta_{\mathcal{P}}^*$ is a right

partial action of the monoid X^* on the set $S_{\mathcal{P}}$, i.e. it is a partial mapping $\delta_{\mathcal{P}}^*: S_{\mathcal{P}} \times X \dashrightarrow S_{\mathcal{P}}$ such that

1. $\delta_{\mathcal{P}}^*(s, \varepsilon) \downarrow = s$ for all $s \in S_{\mathcal{P}}$;
2. if $\delta_{\mathcal{P}}^*(s, \mathbf{u}') \downarrow$ and $\delta_{\mathcal{P}}^*(\delta_{\mathcal{P}}^*(s, \mathbf{u}'), \mathbf{u}'') \downarrow$ then

$$\delta_{\mathcal{P}}^*(s, \mathbf{u}'\mathbf{u}'') \downarrow = \delta_{\mathcal{P}}^*(\delta_{\mathcal{P}}^*(s, \mathbf{u}'), \mathbf{u}''); \quad (5)$$

3. if $\delta_{\mathcal{P}}^*(s, \mathbf{u}') \downarrow$ and $\delta_{\mathcal{P}}^*(s, \mathbf{u}'\mathbf{u}'') \downarrow$ then

$$\delta_{\mathcal{P}}^*(\delta_{\mathcal{P}}^*(s, \mathbf{u}'), \mathbf{u}'') \downarrow = \delta_{\mathcal{P}}^*(s, \mathbf{u}'\mathbf{u}'').$$

4.3 Interrelations between Automata and Pre-automata

In this section we describe some method that allows us to construct a pre-automaton using an automaton.

Suppose that we have taken some automaton $\mathcal{A}(X, S_{\mathcal{A}}, \delta_{\mathcal{A}})$.

Let us define the pre-automaton $\mathcal{P}(X, S_{\mathcal{P}}, \delta_{\mathcal{P}}^*)$ in the following manner:

1. choose as $S_{\mathcal{P}}$ an arbitrary subset of $S_{\mathcal{A}}$;
2. define the partial mapping $\delta_{\mathcal{P}}^*: S_{\mathcal{P}} \times X^* \dashrightarrow S_{\mathcal{P}}$ as follows for $s \in S_{\mathcal{P}}$ and $\mathbf{u} \in X^*$ let assign that $\left. \begin{array}{l} \delta_{\mathcal{P}}^*(s, \mathbf{u}) \uparrow \text{ if } \delta_{\mathcal{A}}^*(s, \mathbf{u}) \notin S_{\mathcal{P}} \text{ and} \\ \delta_{\mathcal{P}}^*(s, \mathbf{u}) \downarrow = \delta_{\mathcal{A}}^*(s, \mathbf{u}) \text{ if } \delta_{\mathcal{A}}^*(s, \mathbf{u}) \in S_{\mathcal{P}}. \end{array} \right\} \quad (6)$

Proposition 4. *The triple defined by construction (6) is a pre-automaton.*

Proof. Indeed, item (3) ensures that item 1) of (5) is satisfied.

Further, Prop. 3 implies that items 2) and 3) of (5) are satisfied. \square

The assertion just proved demonstrates that the method to obtain pre-automata consists in hiding part of the states.

The converse assertion proved in [3] as Globalisation Theorem ensures that the method considered above is the most general method to obtain pre-automata.

5 Moore Machines and Pre-machines

In this section we discuss the question about how a Moore machine or its generalisation, which we call a Moore pre-machine, can realise a “black box”.

5.1 Moore Machines

Usually, automata associate with “black boxes” in the following manner.

Firstly, the class of Moore machines is defined.

Definition 5. *Let $\mathcal{A}(X, S_{\mathcal{A}}, \delta_{\mathcal{A}})$ be an automaton then the corresponding Moore machine is a pentacle $\mathcal{M}_{\mathcal{A}}(X, S_{\mathcal{A}}, \delta_{\mathcal{A}}, s_{\mathcal{M}}^0, \lambda_{\mathcal{M}})$ where $s_{\mathcal{M}}^0$ is some fixed state of \mathcal{A} called the initial state of the machine and $\lambda_{\mathcal{M}}: S_{\mathcal{A}} \rightarrow Y$ is a mapping called the output function of the machine.*

Then for a Moore machine is determined its reaction function.

Definition 6. Let $\mathcal{M}_{\mathcal{A}}(X, S_{\mathcal{A}}, \delta_{\mathcal{A}}, s_{\mathcal{M}}^0, \lambda_{\mathcal{M}})$ be a Moore machine then its reaction function $\mu_{\mathcal{M}}: X^* \rightarrow Y$ is determined by the formula

$$\mu_{\mathcal{M}}(\mathbf{u}) = \lambda_{\mathcal{M}}(\delta_{\mathcal{A}}(s_{\mathcal{M}}^0, \mathbf{u})). \quad (7)$$

Finally, we define the transfer function $M_{\mathcal{M}}: X^{\omega} \rightarrow Y^{\omega}$ for the machine $\mathcal{M}_{\mathcal{A}}$ using its reaction function $\mu_{\mathcal{M}}$ and Algorithm 1.

5.2 Moore Pre-machine

Here we repeat all constructions from the previous subsection substituting a pre-automaton for an automaton.

Firstly, the class of Moore pre-machines is defined.

Definition 7. Let $\mathcal{P}(X, S_{\mathcal{P}}, \delta_{\mathcal{P}}^*)$ be a pre-automaton then the corresponding Moore pre-machine is a pentacle $\mathcal{M}_{\mathcal{P}}(X, S_{\mathcal{P}}, \delta_{\mathcal{P}}^*, s_{\mathcal{M}}^0, \lambda_{\mathcal{M}})$ where $s_{\mathcal{M}}^0$ is some fixed state of \mathcal{P} called the initial state of the pre-machine and $\lambda_{\mathcal{M}}: S_{\mathcal{P}} \rightarrow Y$ is a mapping called the output function of the pre-machine.

Then for a Moore pre-machine is determined its reaction function.

Definition 8. Let $\mathcal{M}_{\mathcal{P}}(X, S_{\mathcal{P}}, \delta_{\mathcal{P}}^*, s_{\mathcal{M}}^0, \lambda_{\mathcal{M}})$ be a Moore pre-machine then its reaction function $\mu_{\mathcal{M}}: X^+ \dashrightarrow Y$ is determined in the following manner

$$\begin{aligned} 1. \mu_{\mathcal{M}}(\mathbf{u}) \uparrow \text{ if } \delta_{\mathcal{M}}^*(s_{\mathcal{M}}^0, \mathbf{u}) \uparrow \text{ and} \\ 2. \mu_{\mathcal{M}}(\mathbf{u}) \downarrow = \lambda_{\mathcal{M}}(\delta_{\mathcal{P}}^*(s_{\mathcal{M}}^0, \mathbf{u})) \text{ if } \delta_{\mathcal{M}}^*(s_{\mathcal{M}}^0, \mathbf{u}) \downarrow. \end{aligned} \quad (8)$$

Finally, we define the transfer function $M_{\mathcal{M}}: X^{\omega} \rightarrow Y^{\infty}$ for the pre-machine $\mathcal{M}_{\mathcal{P}}$ using its reaction function $\mu_{\mathcal{M}}$ and Algorithm 1.

5.3 Posing of Synthesis Problem

The preceding arguments show that machines and pre-machines can be considered as “glass boxes”. It is known that machines are “glass boxes” for a proper subclass of the class of all “black boxes” [4, 6]. Therefore we pose the following problem.

Problem 1 (Synthesis Problem). Suppose we have a mapping $M: X^{\omega} \rightarrow Y^{\infty}$ that holds the non-anticipation property.

It is required to describe the properties of the mapping that ensure the existence of a pre-machine $\mathcal{M}_{\mathcal{P}}$ such that $M_{\mathcal{M}} \cong M$.

6 Solving Synthesis Problem

Solving the problems posed at the end of the previous section is given in three stages: firstly, some solution of the problem is constructed, secondly, this solution is reduced, and, finally, the minimality of this reduced solution is proved.

Taking into account the fact that the hypothesis of the problem includes the non-anticipation property for the mapping M we can consider the reaction function μ of the “black box” instead its transfer function.

6.1 Existence of Solution

Thus we assume that two alphabets (the input alphabet X and the output alphabet Y) and a partial mapping $\mu: X^+ \dashrightarrow Y$ are given.

Let us choose

$$\left. \begin{aligned} S_{\mathcal{F}} &= X^*; \\ \delta_{\mathcal{F}}(\mathbf{u}, x) &= \mathbf{u}x \text{ for } x \in X \text{ and } \mathbf{u} \in X^*. \end{aligned} \right\} \quad (9)$$

Now consider the triple $\mathcal{F}(X, S_{\mathcal{F}}, \delta_{\mathcal{F}})$.

Lemma 1. *The triple $\mathcal{F}(X, S_{\mathcal{F}}, \delta_{\mathcal{F}})$ is an automaton such that the right action $\delta_{\mathcal{F}}^*: S_{\mathcal{F}} \times X^* \rightarrow S_{\mathcal{F}}$ associated with it satisfy the equation*

$$\delta_{\mathcal{F}}^*(\mathbf{u}, \mathbf{v}) = \mathbf{u}\mathbf{v} \quad (10)$$

for all $\mathbf{u}, \mathbf{v} \in X^*$.

Proof. Checking is reduced to a simple application of the mathematical induction. \square

Now let us choose $S_{\mathcal{F}}^{\mu} \subset S_{\mathcal{F}}$ in the following manner:

$$S_{\mathcal{F}}^{\mu} = \{\mathbf{u} \in X^* \mid \mu(\text{sequ}) \downarrow\} \cup \{\varepsilon\}.$$

Now applying construction (6) and we obtain the pre-automaton $\mathcal{F}^{\mu}(X, S_{\mathcal{F}}, \delta_{\mathcal{F}}^{\mu})$.

Theorem 1 (Existence of Solutions for the Synthesis Problem). *Let us consider the Moore pre-machine $\mathcal{M}_{\mathcal{F}}^{\mu}(X, S_{\mathcal{F}}^{\mu}, \delta_{\mathcal{F}}^{\mu*} s_{\mathcal{F}}^0, \lambda_{\mathcal{F}}^{\mu})$, where*

$$\begin{aligned} s_{\mathcal{F}}^0 &= \varepsilon; \\ \lambda_{\mathcal{F}}^{\mu}(\mathbf{u}) &= \mu(\mathbf{u}) \text{ if } \mu(\mathbf{u}) \downarrow; \\ \lambda_{\mathcal{F}}^{\mu}(\varepsilon) &\text{ is defined arbitrary,} \end{aligned}$$

then $\mu_{\mathcal{M}_{\mathcal{F}}^{\mu}} \cong \mu$.

Proof. Really, $\mathcal{M}_{\mathcal{F}}^{\mu}$ is a Moore pre-machine.

Hence we need to prove that $\mu_{\mathcal{M}_{\mathcal{F}}^{\mu}}(\mathbf{u}) \downarrow$ if and only if $\mu(\mathbf{u}) \downarrow$ and the equality $\mu_{\mathcal{M}_{\mathcal{F}}^{\mu}}(\mathbf{u}) = \mu(\mathbf{u})$ holds on the common domain. But this follows immediately from Lemma 1 and the specification of the pre-machine $\mathcal{M}_{\mathcal{F}}^{\mu}$. \square

6.2 Indistinguishability and Syntactic Pre-Machine

The solution that is given in the previous subsection for the Synthesis Problem is too redundant because the state set of the corresponding pre-machine contains too many indistinguishable states. In this subsection we demonstrate the method to eliminate the lack of the construction.

Our consideration refers to some concepts of the theory of ordered sets. The necessary information can be found in [5, Chapter 1].

Definition 9. We shall say that $\mathbf{u}' \in X^*$ can not be distinct from $\mathbf{u}'' \in X^*$ using μ (this assertion is below written as $\mathbf{u}' \lesssim_\mu \mathbf{u}''$) if $\mu(\mathbf{u}'\mathbf{w}) \downarrow$ implies $\mu(\mathbf{u}''\mathbf{w}) \downarrow = \mu(\mathbf{u}'\mathbf{w})$ for any $\mathbf{w} \in X^*$.

Proposition 5. The relation “ \lesssim_μ ” is a quasi-order on X^* satisfying the following condition: if $\mathbf{u}' \lesssim_\mu \mathbf{u}''$ and $\mathbf{w} \in X^*$ then $\mathbf{u}'\mathbf{w} \lesssim_\mu \mathbf{u}''\mathbf{w}$.

Proof. Reflexivity and transitivity of the relation is evident. Now suppose that $\mathbf{u}' \in X^*$, $\mathbf{u}'' \in X^*$, $\mathbf{w} \in X^*$, and $\mathbf{u}' \lesssim_\mu \mathbf{u}''$. If $\mu((\mathbf{u}'\mathbf{w})\mathbf{v}) \downarrow$ for some $\mathbf{v} \in X^*$ then $\mathbf{u}' \lesssim_\mu \mathbf{u}''$ ensures $\mu(\mathbf{u}''(\mathbf{wv})) \downarrow = \mu(\mathbf{u}'(\mathbf{wv}))$ and, therefore, $\mu((\mathbf{u}''\mathbf{w})\mathbf{v}) \downarrow = \mu((\mathbf{u}'\mathbf{w})\mathbf{v})$. \square

The following simple property of “ \lesssim_μ ” is used below.

Proposition 6. The assertions $\mathbf{u}' \lesssim_\mu \mathbf{u}''$ and $\mu(\mathbf{u}') \downarrow$ imply $\mu(\mathbf{u}'') \downarrow = \mu(\mathbf{u}')$.

Proof. To verify the validity of the proposition it is sufficient to put $\mathbf{w} = \varepsilon$ in Def. 9. \square

Definition 10. Let $\mathbf{u}', \mathbf{u}'' \in X^*$ then we say that \mathbf{u}' and \mathbf{u}'' are μ -congruent (this assertion is below written as $\mathbf{u} \equiv_\mu \mathbf{u}''$) if both $\mathbf{u}' \lesssim_\mu \mathbf{u}''$ and $\mathbf{u}'' \lesssim_\mu \mathbf{u}'$ are true.

Proposition 7. The relation “ \equiv_μ ” is a right congruence on X^* that satisfies the following property: if $\mathbf{u}', \mathbf{u}'' \in X^*$ and $\mathbf{u}' \equiv_\mu \mathbf{u}''$ then $\mu(\mathbf{u}') \downarrow$ if and only if $\mu(\mathbf{u}'') \downarrow$ and in this case $\mu(\mathbf{u}') = \mu(\mathbf{u}'')$.

Proof. The fact that “ \equiv_μ ” is an equivalence relation follows from the properties of a quasi-order [5, Sec. 1.3]. Its stability relative to the right multiplication follows immediately from the similar property for the relation “ \lesssim_μ ”. Finally, the last assertion of the proposition follows from Prop. 6. \square

Let us define

$$\left. \begin{aligned} S_A^\mu &= X^* / \equiv_\mu; \\ \delta_A^\mu([\mathbf{u}]_\mu, x) &= [\mathbf{u}x]_\mu, \end{aligned} \right\} \quad (11)$$

where $[\cdot]_\mu$ denotes a class of the μ -congruence, $\mathbf{u} \in X^*$, and $x \in X$. Note that the property to be a right congruence for the equivalence “ \equiv_μ ” ensures the correctness of the definition of δ_A^μ . Now consider the triple $\mathcal{A}^\mu(X, S_A^\mu, \delta_A^\mu)$.

Lemma 2. The triple \mathcal{A}^μ is an automaton such that the right action associated with it $\delta_A^{\mu*} : S_A^\mu \times X^* \rightarrow S_A^\mu$ satisfies the equation

$$\delta_A^{\mu*}([\mathbf{u}]_\mu, \mathbf{v}) = [\mathbf{u}\mathbf{v}]_\mu \quad (12)$$

for all $\mathbf{u}, \mathbf{v} \in X^*$.

Proof. The lemma is easy proved by induction on the length of \mathbf{v} . \square

Now we can note that Prop. 7 ensures one of the alternatives:

$$\text{either } [\mathbf{u}]_\mu \subset \{\mathbf{v} \in X^* \mid \mu(\mathbf{v}) \downarrow\} \text{ or } [\mathbf{u}]_\mu \cap \{\mathbf{v} \in X^* \mid \mu(\mathbf{v}) \downarrow\} = \emptyset.$$

This remark allows us to choose $S_S^\mu \subset S_A^\mu$ in the following manner:

$$S_S^\mu = \{[\mathbf{u}]_\mu \mid \mathbf{u} \in X^* \text{ and } \mu(\mathbf{u}) \downarrow\} \cup \{[\varepsilon]_\mu\}.$$

Hence we can again apply construction (6) and obtain the pre-automaton $\mathcal{S}^\mu(X, S_S^\mu, \delta_S^{\mu*})$.

Theorem 2 (about Syntactic Pre-machine). *Let us consider the Moore pre-machine $\mathcal{M}_S^\mu(X, S_S^\mu, \delta_S^{\mu*}, s_S^0, \lambda_S^\mu)$, where*

$$\begin{aligned} s_S^0 &= [\varepsilon]_\mu; \\ \lambda_S^\mu([\mathbf{u}]_\mu) &= \mu(\mathbf{u}) \text{ if } \mu(\mathbf{u}) \downarrow; \\ \lambda_S^\mu([\varepsilon]_\mu) &\text{ is defined arbitrary,} \end{aligned}$$

then $\mu_{\mathcal{M}_S^\mu} \cong \mu$.

Proof. Let us note that Prop. 7 ensures the correctness for the definition of the mapping λ_S^μ . Further, Lemma 2 guarantees the validity of $\mu_{\mathcal{M}_S^\mu} \cong \mu$. \square

Remark 6. We shall call the Moore pre-machine built in the theorem the syntactic pre-machine.

6.3 Syntactic Pre-machine as Minimal Solution of Synthesis Problem

To complete the program indicated above, we need to establish the minimality of the pre-machine \mathcal{M}_S^μ in any sense.

First of all, we note that the pre-machine \mathcal{M}_S^μ holds evidently the following property called the reachability: one can obtain any state of the pre-machine applying its partial action to the initial state.

Now let us formulate the main result.

Theorem 3 (about Minimality of Syntactic Pre-machine). *For any reachable Moore pre-machine $\mathcal{M}_P(X, S_P, \delta_P^*, s_{\mathcal{M}}^0, \lambda_{\mathcal{M}})$ such that $\mu_{\mathcal{M}} \cong \mu$ there exists a mapping $\psi: S_P \rightarrow S_S^\mu$ satisfying the following conditions*

1. *for any $s \in S_P$ and $\mathbf{u} \in X^*$ the assertion $\delta_P^*(s, \mathbf{u}) \downarrow$ implies $\delta_S^{\mu*}(\psi(s), \mathbf{u}) \downarrow = \psi(\delta_P(s, \mathbf{u}))$;*
2. *$\psi(s_{\mathcal{M}}^0) = s_S^0$;*
3. *$\mu \cong \mu_{S^\mu} \circ \psi$.*

Proof. The key item of the proof is the construction of the mapping ψ .

Let $s \in S_P$ and $\mathbf{u}', \mathbf{u}'' \in X^*$ such that $\delta_P^*(s_{\mathcal{M}}^0, \mathbf{u}') \downarrow = s$ and $\delta_P(s_{\mathcal{M}}^0, \mathbf{u}'') \downarrow = s$ then we can show that $\mathbf{u}' \lesssim_\mu \mathbf{u}''$.

Indeed, suppose that $\mu(\mathbf{u}'\mathbf{w}) \downarrow$ for some $\mathbf{w} \in X^*$. Taking into account that

$\mu_{\mathcal{M}} \cong \mu$ we can write $\mu(\mathbf{u}'\mathbf{w}) = \lambda_{\mathcal{M}}(\delta_{\mathcal{M}}^*(s_{\mathcal{M}}^0, \mathbf{u}'\mathbf{w}))$.
 Note that the previous equality ensures $\delta_{\mathcal{M}}^*(s_{\mathcal{M}}^0, \mathbf{u}'\mathbf{w}) \downarrow$ and therefore (5) leads to the conclusion that $\delta_{\mathcal{P}}^*(\delta_{\mathcal{P}}^*(s_{\mathcal{M}}^0, \mathbf{u}'), \mathbf{w}) \downarrow$.
 Using the supposition $\delta_{\mathcal{P}}^*(s_{\mathcal{M}}^0, \mathbf{u}') \downarrow = s$ and (5) we obtain

$$\mu(\mathbf{u}'\mathbf{w}) = \lambda_{\mathcal{M}}(\delta_{\mathcal{P}}^*(\delta_{\mathcal{P}}^*(s_{\mathcal{M}}^0, \mathbf{u}'), \mathbf{w})) = \lambda_{\mathcal{M}}(\delta_{\mathcal{P}}^*(s, \mathbf{w})).$$

This equation ensures $\delta_{\mathcal{P}}^*(s, \mathbf{w}) \downarrow$.
 Hence the supposition $\delta_{\mathcal{P}}^*(s_{\mathcal{M}}^0, \mathbf{u}'') \downarrow = s$ implies $\delta_{\mathcal{P}}^*(\delta_{\mathcal{P}}^*(s_{\mathcal{M}}^0, \mathbf{u}''), \mathbf{w}) \downarrow = \delta_{\mathcal{P}}^*(s, \mathbf{w})$.
 Thus $\mu(\mathbf{u}''\mathbf{w}) \downarrow = \mu(\mathbf{u}'\mathbf{w})$ and, therefore, $\mathbf{u}'' \lesssim_{\mu} \mathbf{u}'$.
 Similar reasoning gives $\mathbf{u}' \lesssim_{\mu} \mathbf{u}''$ and, therefore, $\mathbf{u}' \equiv_{\mu} \mathbf{u}''$.
 Now we can define ψ in the following manner:

$$\psi(s) = [\mathbf{u}_s]_{\mu} \quad \text{where } s = \delta_{\mathcal{P}}^*(s_{\mathcal{M}}^0, \mathbf{u}_s).$$

Checking the validity of items 1)–3) for the constructed mapping ψ is a simple exercise now. \square

7 Conclusion

Thus, we can summarize that the paper gives the algebraic analysis for the problem of realisation “black boxes” by machines. The main results of the analysis are

- the generalisation of the non-anticipation property for “black boxes” that accumulate information for decision;
- the complete solution of the synthesis problem for such “black boxes”.

The machines that realise the corresponding transfer functions are based on pre-automata. The class of such algebraic structures had been introduced by author jointly with Prof. M. Dokuchaev and Prof. B. Novikov in earlier papers.

It should be emphasized that issues dealing with computational properties of pre-machines has not considered in the paper. The coverage of these issues requires a separate study.

References

1. Ashby, W.R.: An introduction to cybernetics. Chapman & Hall, London (1956)
2. Clifford, A.H., Preston, G.B.: The Algebraic Theory of Semigroups, Volume 1. AMS (1961)
3. Dokuchaev, M., Novikov, B., Zholtkevych, G.: Partial actions and automata. Alg. and Discr. Math. 11, 51–63 (2011)
4. Glushkov, V.M.: Some problems in the synthesis of digital automata. USSR Computational Mathematics and Mathematical Physics. 1(3), 399–446 (1962)
5. Harzheim, E.: Ordered Sets. Springer Science+Business Media Inc., New York (2005)

6. Trakhtenbrot, B.A., Barzdin, J.M.: Finite automata: behaviour and synthesis. North-Holland Publishing Company, US (1973)
7. Zholtkevych, G., Novikov, B., Dorozhinsky, V.: Pre-automata and Complex Event Processing. In: Ermolayev, V. et al (eds) ICT in Education, Research, and Industrial Applications. CCIS, vol. 469, pp. 100–116. Springer International Publishing (2014)