Coalgebraic Understanding of Random Systems with Output

Artem Panchenko, Grygoriy Zholtkevych

¹School of Math and Computer Science, V.N. Karazin Kharkiv National University, 4 Svobody Sqr, Kharkiv, 61022, Ukraine

Abstract

The safety concept for clock constraints has already been coalgebraically studied for fully deterministic cyber-physical systems (CPS). However, the corresponding control systems, as a rule, cannot be implemented in practice. There are many reasons for this, and the main one is the inability to collect accurate information about the state of the system in real time. In this article, the theory of deterministic systems is generalised to the case of stochastic systems. The key tool for this generalisation is the use of the distribution monad. The advantage of the proposed approach is that it ensures that the previously proposed structure can be implemented in real-time systems, the behaviour of which could not be otherwise specified based on a purely deterministic approach.

Keywords

discrete system, system with output, random system, coalgebraic approach

1. Introduction

It is known that rapid growth of CPS capacity is not possible today, primarily due to the limitations of technological progress and the atomic nature of the substances from which transistors, which provide computing power, are made, [1]. Alternative approaches are therefore being designed to address this problem, the main one being the design of distributed systems.

Herein distributed system is a collection of independent entities that cooperate to solve a problem that cannot be individually solved [2]. We intentionally did not use the expression "computing machines" in the definition but replaced it with the word entities, because we mean that the nodes of such a system can be both subtracting machines and different kinds of controllers and sensors, as far as modern technical systems are complexes of interacting physical and cybernetical components.

Hence, we can say that the mainstream of developing modern technical systems is developing cyber-physical systems (CPS). However, due to the fact that cyber-physical systems, as a rule, are deeply integrated with control elements, unforeseen situations can emerge which could lead to system failure. The consequences of such emergencies can be catastrophic [3]

To identify emergencies in real-time mode, we need to have a precise specification of the system behaviour that must consider both desired and actual system states. The solving of this

problem requires the development of formal methods for modelling the behaviour of distributed systems. The development of these methods should be carried out through a formal description of security restrictions for the designed CPS. We assume that we can describe all constraints by a cause-and-effect relationship between events in the system. It is good to take into account the cause-and-effect approach and that it is possible to use only the logical clocks model in distributed systems during the synchronisation of its elements [4]. Therefore, the main problem for us is to maintain the causal relationship between events in the system.

One of the common approaches to this problem solution is to use models based on the theory of categories and coalgebras. This approach is proposed in the works of J. Rutten [5] and B. Jacobs [6]. In the context of implementing this approach to the semantic framework for building behavioural models, it should be noted the work [7, 8], which aims to create a generic tool for developing semantic models for specification language constraints of distributed systems based on the concept of logical clocks.

The main shortcoming of the solutions proposed above [7, 8] is the assumption of complete determinism of the systems under study. Obviously, in practice, such an assumption cannot be fulfilled, or its implementation will be costly. This is due to many factors, the main of which is the distributed nature of the systems, as well as real-time work. Thus, this article is a continuation of the research carried out by G. Zholtkevych and M. Labzhaniia [8] and an extension of the theory to the case when we work with stochastic systems.

This article makes extensive use of category theory concepts and facts. Therefore, we use the nLab [9] web resource for familiarizing readers with these concepts and facts. When citing an nLab article, we provide the corresponding URL.

In the paper, we consider coalgebras over the category of sets and mappings denoted below by **Set**.

This paper is an introduction to the research area and structured as follows

- Section 3 contains a brief survey of general category-theoretic and coalgebraic concepts used in the paper.
- Section 4 devoted to the concept of the finite distribution endofunctor of category **Set**.
- Section 5 contains the definitions and preliminary results related to the coalgebraic approach for studying random systems with outputs.

2. Related Work

The approach whereby we use logical rules to verify the work of CFS is sufficiently widespread. It is also worth noting the scientific community's awareness of the need to deal with the safety of systems by applying formal methods.

2.1. Design Challenges Review

The problems of modern cyber-physical systems design are clearly defined in [10]. It is shown that due to the specific field of application of such systems we are obliged to see increased trespasses to the reliability of its components. The paper concluded that standard models for constructing specifications that cope with creating single-threaded systems are bad at

working with distributed systems. It was also concluded that it was necessary to develop formal approaches to describing the specification of the cyber-physical system.

2.2. Logical rules usage

Due to problems with the complexity of synchronizing the states of the asynchronous system elements, we, as described above, the set of authors use exactly logical dependencies between states. In [11] the authors of such rules use a human-friendly UML diagram format. Later, the authors convert this specification model to a verifiable model in nuXmv format. This approach allows authors to use different models of specification (both UML diagrams and Petri nets), which is an indisputable advantage. However, this article does not raise the problem of working with systems with total or partial uncertainty, that is, randomly.

It is worth noting that the use of logical (cause-effect) rules to describe events in parallel computing systems is the most common approach. Paper [12] points out the disadvantages of using UML associated with the model of time, called Simple Time, which is used in it. Thus, the necessity for a more complex and formal narrative model is evident.

2.3. Coalgebraic approach

As we have discussed above, the priority in the modelling of shared systems is to develop formal models to describe their work with the use of cause-effect relationship between events. One possible solution to this problem is to build coalgebraic models. This is primarily because of the induction principle [5] present in this mathematical model.

In addition to the studies described above [7, 8], the continuation of which is this article attention should also be drawn to [13]. The interest of this article is that the author also works with probabilistic systems. However, the author does not emphasize the application of this theory in distributed systems, which is a disadvantage.

3. Coalgebra Definition and Related Concepts

In this section, we introduce the definition of a coalgebra and related basic concepts following to [9, https://ncatlab.org/nlab/show/coalgebra+for+an+endofunctor]. In addition, some specific concepts are discussed for the case the underlying category is the category **Set**.

Thus, we assume in this section some category $\mathbb{C}$ called underlying and its endofunctor $\boldsymbol{F}$ are given and fixed.

Definition 1. *A morphism $\boldsymbol{a}$ of $\mathbb{C}$ is called an $\boldsymbol{F}$-coalgebra if the equality $\mathrm{cod}\,\boldsymbol{a} = \boldsymbol{F}(\mathrm{dom}\,\boldsymbol{a})$ is fulfilled. In the case, $\mathrm{dom}\,\boldsymbol{a}$ is called the carrier of $\boldsymbol{a}$ and denoted below by $\underline{\boldsymbol{a}}$.*

More detail see [9, https://ncatlab.org/nlab/show/coalgebra+for+an+endofunctor].

Definition 2. *Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be $\boldsymbol{F}$-coalgebras then a morphism $f : \underline{\boldsymbol{a}} \to \underline{\boldsymbol{b}}$ is called an $\boldsymbol{F}$-*

morphism from $\boldsymbol{a}$ into $\boldsymbol{b}$ (symbolically, $f : \boldsymbol{a} \to \boldsymbol{b}$) if the diagram

$$
\begin{array}{ccc}
\underline{\boldsymbol{a}} & \xrightarrow{\ f\ } & \underline{\boldsymbol{b}} \\
{\scriptstyle a}\Big\downarrow & & \Big\downarrow{\scriptstyle b} \\
\boldsymbol{F}\,\underline{\boldsymbol{a}} & \xrightarrow{\ \boldsymbol{F}f\ } & \boldsymbol{F}\,\underline{\boldsymbol{b}}
\end{array}
$$

commutes or, equivalently, the equation $(\boldsymbol{F}f)a = bf$ holds.

More detail see [9, https://ncatlab.org/nlab/show/coalgebra+for+an+endofunctor].

Proposition 1. *The class of $\boldsymbol{F}$-coalgebras equipped with $\boldsymbol{F}$-morphisms is a category, which is usually denoted by $\mathbf{Coalg}_{\boldsymbol{F}}(\mathbb{C})$ or by $\mathbf{Coalg}_{\boldsymbol{F}}$ if the underlying category $\mathbb{C}$ is clear from the context.*

Another important concept for a coalgebraic theory is the notion of bisimulation. For coalgebras of the category $\mathbf{Set}$, this notion is introduced by the next definition [5]

Definition 3. *A bisimulation of $\boldsymbol{F}$-coalgebras $\boldsymbol{a}$ and $\boldsymbol{b}$ is a span $\boldsymbol{a} \xleftarrow{\mathrm{pr}_a} \boldsymbol{r} \xrightarrow{\mathrm{pr}_b} \boldsymbol{b}$ in the category of $\boldsymbol{F}$-coalgebras where $\underline{\boldsymbol{r}} \subset \underline{\boldsymbol{a}} \times \underline{\boldsymbol{b}}$, and pr_a and pr_b are restrictions on $\underline{\boldsymbol{r}}$ of the corresponding projections i.e. the diagram*

$$
\begin{array}{ccccc}
\underline{\boldsymbol{a}} & \xleftarrow{\ \mathrm{pr}_a\ } & \underline{\boldsymbol{r}} & \xrightarrow{\ \mathrm{pr}_b\ } & \underline{\boldsymbol{b}} \\
{\scriptstyle a}\Big\downarrow & & {\scriptstyle r}\Big\downarrow & & \Big\downarrow{\scriptstyle b} \\
\boldsymbol{F}\,\underline{\boldsymbol{a}} & \xleftarrow{\ \boldsymbol{F}\,\mathrm{pr}_a\ } & \boldsymbol{F}\,\underline{\boldsymbol{r}} & \xrightarrow{\ \boldsymbol{F}\,\mathrm{pr}_b\ } & \boldsymbol{F}\,\underline{\boldsymbol{b}}
\end{array}
$$

commutes.

Final objects of category of coalgebras are very important for constructing semantic models of programming and specification languages.

Definition 4. *A terminal object of $\mathbf{Coalg}_{\boldsymbol{F}}$ if it exists is called a final $\boldsymbol{F}$-coalgebra, which is denoted by $\nu\boldsymbol{F}$.*

Definition 5. *For any $\boldsymbol{F}$-coalgebra $\boldsymbol{a}$, the unique $\boldsymbol{F}$-morphism from $\boldsymbol{a}$ into $\nu\boldsymbol{F}$ is called an anamorphism and denoted by $[\![a]\!]$.*

The concepts of a final coalgebra and an anamorphism one can found in [9] with hyperlinks http://ncatlab.org/nlab/show/terminal+coalgebra+for+an+endofunctor and http://ncatlab.org/nlab/show/cocycle respectively.

Further, we follow the approach proposed by J. Rutten [5] namely to consider an $\boldsymbol{F}$-coalgebra for endofunctor $\boldsymbol{F}$ of the category $\mathbf{Set}$ as a discrete system called $\boldsymbol{F}$-system.

Remark 1. Coalgebra is a powerful tool for modelling dynamic systems by definition. Inclusively it illustrates the change of some mathematical entity under the influence of some transformation of it. We will always think of a dynamic system as a coalgebra, morphism as a set of rules to change its states. The F-functor, in its turn, we interpret as a tool to preserve our hypotheses when switching to another system, thus creating an upper-level abstraction of the operation of an entire class of dynamic systems.

4. Distribution Endofunctor

The concept of an endofunctor of finitary distributions is here introduced similarly as in [9, http://ncatlab.org/nlab/show/distribution+monad] and [14].

To expand the System with Output in the case of random selection elements we need to set a probability distribution rule. Since we use a coalgebra approximation as the basis of the model, we need to provide an appropriate design. That's why we need Distribution Endofunctor.

For any X : **Set**, let us define

$$\mathcal{D}X = \left\{ p : X \to [0,1] \;\middle|\; \begin{array}{l} p(x) \neq 0 \text{ for finite number } x \in X \text{ only} \\ \text{and } \sum_{x \in X} p(x) = 1 \end{array} \right\} \qquad (\mathcal{D}\text{-obj})$$

Now for any mappings $f : X \to Y$, let us define $\mathcal{D}f : \mathcal{D}X \to \mathcal{D}Y$ as follows

$$\mathcal{D}f = \boldsymbol{\lambda}\, p \,.\, \boldsymbol{\lambda}\, y \in Y \,.\, \sum_{x \in X | fx=y} p(x). \qquad (\mathcal{D}\text{-mor})$$

Of course, we need to check ($\mathcal{D}$-mor) specifies an element of $\mathcal{D}Y$. Indeed,

$$0 \leq \big((\mathcal{D}f)p\big)(y) = \sum_{x \in X | fx=y} p(x) \leq \sum_{x \in X} p(x) = 1$$

and

$$\sum_{y \in Y} \big((\mathcal{D}f)p\big)(y) = \sum_{y \in Y} \sum_{x \in X | fx=y} p(x) = \sum_{x \in X} p(x) = 1.$$

Remark 2. Informally within our model ($\mathcal{D}$-obj) will mean the following: the function P gives the probability of this transition for each variant of the following state. Then to the end of the section, we simply prove the correctness of the work of this function and that it is a function of the probability distribution.

Proposition 2. *Formulae* (1) *determine an endofunctor of the category* **Set**.

Proof. For any set X, we have evidently $(\mathcal{D}\,\mathrm{id}_X)p = \boldsymbol{\lambda}\, x \in X \,.\, p(x) = p$ i.e. $\mathcal{D}\,\mathrm{id}_X = \mathrm{id}_{\mathcal{D}X}$. Further, for $X \xrightarrow{f} Y \xrightarrow{g} Z$,

$$\big((\mathcal{D}g)(\mathcal{D}f)\big)p = (\mathcal{D}g)\big((\mathcal{D}f)p\big) = (\mathcal{D}g)\left(\boldsymbol{\lambda}\, y \in Y \,.\, \sum_{x \in X | fx=y} p(x) \right)$$

$$= \boldsymbol{\lambda}\, z \in Z \,.\, \sum_{y \in Y | g(y)=z} \left(\boldsymbol{\lambda}\, y \in Y \,.\, \sum_{x \in X | fx=y} p(x) \right)(y)$$

$$= \boldsymbol{\lambda}\, z \in Z \,.\, \sum_{y \in Y | g(y)=z} \left(\sum_{x \in X | fx=y} p(x) \right) = \boldsymbol{\lambda}\, z \in Z \,.\, \sum_{x \in X | g(f(x))=z} p(x)$$

$$= \big(\mathcal{D}(gf)\big)p.$$

Thus, $\mathcal{D}$ is an endofunctor of **Set**. $\qquad\square$

Let us take two sets X and Y then we have the universal arrow $X \xleftarrow{\mathrm{pr}_X} X \times Y \xrightarrow{\mathrm{pr}_Y} Y$ representing the product in the category **Set**. Applying the endofunctor $\mathcal{D}$ gives

$$\mathcal{D}X \xleftarrow{\mathcal{D}\,\mathrm{pr}_X} \mathcal{D}(X \times Y) \xrightarrow{\mathcal{D}\,\mathrm{pr}_Y} \mathcal{D}Y$$

where $(\mathcal{D}\,\mathrm{pr}_X)p = \boldsymbol{\lambda}\, x \in X \,.\, \sum\limits_{y \in Y} p(x,y)$ and $(\mathcal{D}\,\mathrm{pr}_Y)p = \boldsymbol{\lambda}\, y \in Y \,.\, \sum\limits_{x \in X} p(x,y)$.

In other words, $(\mathcal{D}\,\mathrm{pr}_X)p$ and $(\mathcal{D}\,\mathrm{pr}_Y)p$ are marginal distributions of the joint distribution p.

5. Concept of a Random System with Output and Related Concepts

Remind that a system with output [7] is a coalgebra of type determines by the endofunctor $\mathcal{S}_N$ of the category **Set** where

$$\mathcal{S}_N X = N \times X \qquad \text{for any set } X \qquad\qquad (\mathcal{S}_N\text{-obj})$$
$$\mathcal{S}_N f = \mathrm{id}_N \times f \qquad \text{for any sets } X, Y \text{ and mapping } f : X \to Y \qquad (\mathcal{S}_N\text{-mor})$$

where N is some finite set whose elements are interpreted as system notifications.

We use the idea of randomization for obtaining random systems with output.

5.1. Random Systems with Outputs

More precisely, let us consider the endofunctor $\mathcal{R}_N$ of **Set** defined as follows

$$\mathcal{R}_N X = (\mathcal{D}\mathcal{S}_N)X = \mathcal{D}(N \times X) \quad \text{for any set } X \qquad\qquad (\mathcal{R}_N\text{-obj})$$
$$\mathcal{R}_N f = (\mathcal{D}\mathcal{S}_N)f = \mathcal{D}(\mathrm{id}_N \times f) \quad \text{for any sets } X, Y \text{ and mapping } f : X \to Y. \quad (\mathcal{R}_N\text{-mor})$$

Definition 6. *Let X be a set then a mapping $\boldsymbol{\sigma} : X \to \mathcal{R}_N X$ is below called a random system with output. In this case, the set X is called the carrier of $\boldsymbol{\sigma}$ and denoted by $\underline{\boldsymbol{\sigma}}$.*

Remark 3. It is obvious that the endofunctor ($\mathcal{R}_N$-obj) manages the modeling problem of Random Systems with Outputs. Meaningfully, this means that we define a set of following possible states with probability distributions on them for each state of the system.

Example 1. *Let us denote by $\mathcal{T}_N$ the set of infinite trees such that (see Fig. 1)*

1. *each internal node has at most finite children;*
2. *each internal node excluding the root node is labelled by an element of N;*
3. *each edge is labelled by a number from $[0,1]$ such that the sum of labels for edges outgoing from the same node is 1.*

Note that each subtree of a tree of the specified kind is a tree of this kind if to erase the label of its root. This note leads us to the definition of a system $\boldsymbol{\mu} : \mathcal{T}_N \to \mathcal{R}_N \mathcal{T}_N$.

If $t \in \mathcal{T}_N$ and $t_1, \ldots, t_{k_1}$ are the subtrees corresponding children of the root of t then denote by $t'_1, \ldots, t'_{k_1}$ the trees obtained by erasing labels in their roots and define the distribution
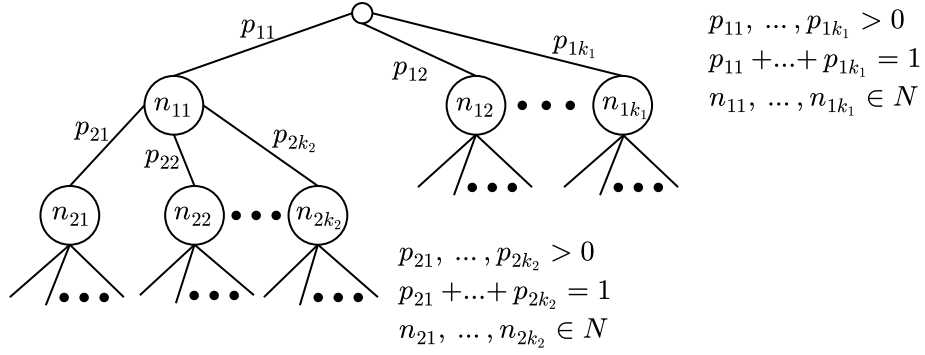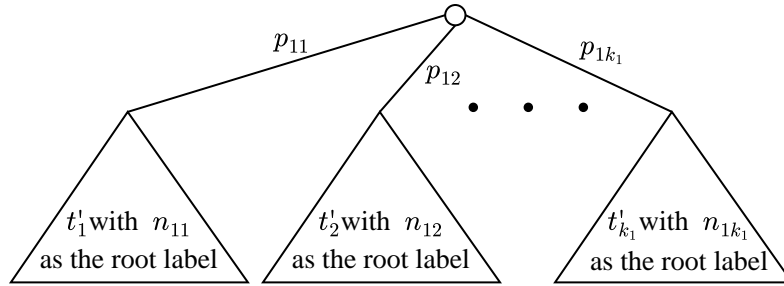
Figure 1: An element of $\mathcal{T}_N$

$$\mu t = \sum_{i=1}^{k_1} p_{1i} \cdot \delta_{(n_{1i}, t_i')} \qquad (2)$$

Easy to understand that μt ensures restoring uniquely of t by follows (see Fig. 2):

1. restore tree t_i for $i = 1, \ldots, k_1$ by labelling its root with n_{1i};
2. create a new root;
3. create an edge from the new root to the root of t_i and label it with p_{1i}.

Figure 2: Restoring t using μt

5.2. Morphisms of Random Systems with Outputs

Morphism is very important when working with coalgebra structures. Substantively, within the framework of our model, morphism allows us to move from one particular Random Systems with Outputs to another. Thus, our reasoning will be true not only for a particular task but for a whole class of tasks.

The next proposition gives a refinement of the general definition of a coalgebraic morphism (see [9, https://ncatlab.org/nlab/show/coalgebra+for+an+endofunctor]) for the case of $\mathcal{R}_N$-coalgebras.

Proposition 3. *For systems with output $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$, a mapping $f : \underline{\boldsymbol{\sigma}} \to \underline{\boldsymbol{\tau}}$ is an $\mathcal{R}_N$-morphism from $\boldsymbol{\sigma}$ into $\boldsymbol{\tau}$ if*

$$\big(\boldsymbol{\tau}(fx)\big)(n,y) = \sum_{x' \in \underline{\boldsymbol{\sigma}} \,|\, fx'=y} (\boldsymbol{\sigma}x)(n,x') \tag{3}$$

whenever $n \in N$ and $y \in \underline{\boldsymbol{\tau}}$.

Proof. The mapping f is an $\mathcal{R}_N$-morphism iff the next diagram commutes

or, in other words, the next equation holds.

$$(\mathcal{R}_N f)\boldsymbol{\sigma} = \boldsymbol{\tau} f \tag{4a}$$

Therefore, let us compute the left and right sides of (4a).
For the left side, we have

$$\Big(\big((\mathcal{R}_N f)\boldsymbol{\sigma}\big)x\Big)(n,y) = \big((\mathcal{R}_N f)(\boldsymbol{\sigma}x)\big)(n,y)$$

$$= \big(\mathcal{D}(\mathrm{id}_N \times f)(\boldsymbol{\sigma}x)\big)(n,y) = \sum_{x' \in \underline{\boldsymbol{\sigma}} \,|\, fx'=y} (\boldsymbol{\sigma}x)(n,x'). \tag{4b}$$

And for the right side, we have

$$\Big((\boldsymbol{\tau} f)x\Big)(n,y) = \big(\boldsymbol{\tau}(fx)\big)(n,y). \tag{4c}$$

Thus, equation (3) ensures the validity of equation (4a) due to (4b) and (4c).
Conversely, equation (4a) ensures evidently the validity of equation (3). $\qquad\square$

Remark 4. Let us denote by $x \xrightarrow[\boldsymbol{\sigma}]{n} x'$ the event "system $\boldsymbol{\sigma}$ has passed from state x into state x' having sent notification n" then $(\boldsymbol{\sigma}x)(n,x')$ is interpreted as the probability of this event i.e.

$$(\boldsymbol{\sigma}x)(n,x') = \Pr\left(x \xrightarrow[\boldsymbol{\sigma}]{n} x'\right).$$

Using this denotation, one can rewrite formula (3) as follows

$$\Pr\left(fx \xrightarrow[\boldsymbol{\tau}]{n} y\right) = \sum_{x' \in \underline{\boldsymbol{\sigma}} \,|\, fx'=y} \Pr\left(x \xrightarrow[\boldsymbol{\sigma}]{n} x'\right). \tag{3*}$$

Due to general theory, the class of random systems with output equipped with $\mathcal{R}_N$-morphism is a category, which is denoted by $\mathbf{Coalg}_{\mathcal{R}_N}$.

Example 2. *Let $\boldsymbol{\sigma}$ be a random system with output then for any $x \in \underline{\boldsymbol{\sigma}}$, one can build a tree $\kappa_{\boldsymbol{\sigma}} x$ from $\mathcal{T}_N$ as follows*

1. *set of children for the root is formed by pairs of $(n, y) \in N \times \underline{\boldsymbol{\sigma}}$ such that $(\boldsymbol{\sigma} x)(n, y) > 0$;*
2. *node (n, y) is labelled with n;*
3. *the edge connected the root and node (n, y) is labelled with $(\boldsymbol{\sigma} x)(n, y)$;*
4. *then this process is repeated for each node (n, y) using y instead of x.*

It is evident the tree $\kappa_{\boldsymbol{\sigma}} x$ meets all constraints characterising trees from $\mathcal{T}_N$.
Moreover, $\boldsymbol{\mu}(\kappa_{\boldsymbol{\sigma}} x) = (\mathcal{R}_N \kappa_{\boldsymbol{\sigma}})(\boldsymbol{\sigma} x)$ by construction of $\kappa_{\boldsymbol{\sigma}}$.

Conjecture. *The constructed above random system with output $\boldsymbol{\mu}$ is a final random system with output and the constructed family of $\mathcal{R}_N$-morphisms $\{\kappa_{\boldsymbol{\sigma}} : \boldsymbol{\sigma} \to \boldsymbol{\mu} \mid \boldsymbol{\sigma} : \mathbf{Coalg}_{\mathcal{R}_N}\}$ is the family of anamorphisms.*

5.3. Bisimulation of Systems with Output

As it was said above, yet another important concept used for studying coalgebras is the concept of bisimulation. In general point of view, the concept was discussed in [15].

Proposition 4. *A system with output $\boldsymbol{\rho}$ is a bisimulation of systems with output $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ if the next is true*

$$\underline{\boldsymbol{\rho}} \subset \underline{\boldsymbol{\sigma}} \times \underline{\boldsymbol{\tau}} \tag{5a}$$

$$\Pr\left(x \xrightarrow[\boldsymbol{\sigma}]{n} x'\right) = \sum_{y' \in \underline{\boldsymbol{\sigma}} \mid (x', y') \in \underline{\boldsymbol{\rho}}} \Pr\left((x, y) \xrightarrow[\boldsymbol{\rho}]{n} (x', y')\right) \tag{5b}$$

$$\Pr\left(y \xrightarrow[\boldsymbol{\tau}]{n} y'\right) = \sum_{x' \in \underline{\boldsymbol{\tau}} \mid (x', y') \in \underline{\boldsymbol{\rho}}} \Pr\left((x, y) \xrightarrow[\boldsymbol{\rho}]{n} (x', y')\right). \tag{5c}$$

Proof. $\boldsymbol{\rho}$ is a bissimulation of $\boldsymbol{\sigma}$ and $\boldsymbol{\tau}$ iff the next diagramm commutes

$$
\begin{array}{ccccc}
\underline{\boldsymbol{\sigma}} & \xleftarrow{\mathrm{pr}_\sigma} & \underline{\boldsymbol{\rho}} & \xrightarrow{\mathrm{pr}_\tau} & \underline{\boldsymbol{\tau}} \\
\sigma \downarrow & & \rho \downarrow & & \tau \downarrow \\
\mathcal{R}_N \underline{\boldsymbol{\sigma}} & \xleftarrow{\mathcal{R}_N \mathrm{pr}_\sigma} & \mathcal{R}_N \underline{\boldsymbol{\rho}} & \xrightarrow{\mathcal{R}_N \mathrm{pr}_\tau} & \mathcal{R}_N \underline{\boldsymbol{\tau}}
\end{array}
$$

or in other worlds, (5a) is evident and pr_σ and pr_τ are both $\mathcal{R}_N$-morphisms. Thus, we can apply Prop. 3 for these mappings. It and Remark 4 give (5b) and (5c). $\qquad \square$

6. Conclusions

Summarizing the above discussion, it can be argued that the study of the category of $\mathcal{R}_N$-colgebras is interesting in the context of constructing semantic models for random discrete systems with output. Thanks to the methods described in this article we can solve the problem of the impossibility of full determinacy (deterministic) of a system that occurs in CPS. Based on this discussion, we can outline the following areas of research:

Based on this discussion, we can outline the following areas of research

- To develop the formal technique for specifying and analysing the final random system with output. This model should be less geometrical and more algebraical than the model given in Subsec. 4.1 and 4.2.
- To prove the Conjecture of Subsec. 4.2 with basing this formal technique.
- To construct an analogue of bifunctor $\mathcal{J}$ introduced in [7, 8] as a tool for studying safety constraints and causality constraints by joining a system with output and a detector.
- To construct a random analogue of a system with termination for defining indicators of system failures.
- To give a probability-theoretic interpretation of system random properties recognised by the construction with use the analogue of bifunctor $\mathcal{J}$.
- To extend the technique being developed to discrete distributed systems for the analysis of causality relations based on the logical clock model.

References

[1] G. Moore, No exponential is forever: but "forever" can be delayed! [semiconductor industry], in: 2003 IEEE International Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC., 2003, pp. 20–23 vol.1. doi:10.1109/ISSCC.2003.1234194.

[2] A. D. Kshemkalyani, M. Singhal, Distributed Computing: Principles, Algorithms, and Systems, Cambridge University Press, 2008.

[3] S. Tyszberowicz, D. Faitelson, Emergence in cyber-physical systems: potential and risk, Frontiers of Information Technology & Electronic Engineering 21 (2020) 1554–1566.

[4] L. Lamport, Time, clocks, and the ordering of events in a distributed system, CACM 21 (1978) 558–565.

[5] J. Rutten, Universal coalgebra: a theory of systems, Theoretical Computer Science 249 (2000) 3–80.

[6] B. Jacobs, Introduction to Coalgebra: Towards Mathematics of States and Observation, Cambridge Tracts in Theoretical Computer Science, 2016.

[7] G. Zholtkevych, M. Labzhaniia, Understanding Safety Constraints Coalgebraically, in: V. Lytvyn, et al. (Eds.), Computational Linguistics and Intelligent Systems, volume 2604 of *CEUR Workshop Proceedings*, 2020, pp. 1–19.

[8] G. Zholtkevych, M. Labzhaniia, Coalgebraic Approach to Studying Discrete Systems with Output, in: A. Bollin, et al. (Eds.), Information and Communication Technologies in Education, Research, and Industrial Applications, volume 1308 of *CCIS*, Springer, Cham, 2021, pp. 141–165.

[9] nLab authors, nLab, http://ncatlab.org/nlab/, 2021. Revision 286.

[10] E. A. Lee, Cyber physical systems: Design challenges, in: 2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC), 2008, pp. 363–369. doi:10.1109/ISORC.2008.25.

[11] I. Grobelna, Formal verification of control modules in cyber-physical systems, Sensors (Basel, Switzerland) 20 (2020) 5154. doi:10.3390/s20185154.

[12] C. André, Syntax and semantics of the clock constraint specification language (ccsl), [Research Report] (2009).

[13] T. Gu, F. Zanasi, Coalgebraic semantics for probabilistic logic programming, Logical Methods in Computer Science 17 (2021). URL: https://arxiv.org/pdf/2012.03916.pdf.

[14] B. Jacobs, From Probability Monads to Commutative Effectuses, Journal of Logical and Algebraic Methods in Programming 94 (2018) 200–237.

[15] B. Jacobs, J. Rutten, A tutorial on (co)algebras and (co)induction, EATCS Bulletin 62 (1997) 62–222.