Categorical approach to Petri nets transformations

Milan Urbášek*
urbasek@dcse.fee.vutbr.cz
Milan Češka*
ceska@dcse.fee.vutbr.cz

Abstract: The aim of this paper is to introduce the idea of comprehension of Petri nets and another related state-oriented models using the categorical approach to link the different models together. This idea is based on the description of state-based models as categories with behaviour-preserved morphisms as arrows in the categories. Associating the models allows one to find the similarities of models and use existing analysis methods involving in other theories for investigating properties of Petri nets.

Key Words: Categories, Petri nets transformations, transition systems, functors

1 Introduction

Many applications in the computer science, including software development and simulation, use the approach of concurrent processes in the sense of existence several independent parts in a bigger system, each part doing its own actions and communication with other parts.

This is a natural principle, which has been observed in many living systems, including the human body. The important area of applications is in modelling and simulation.

From this point of view, it looks promising to start developing the models of real problems using this approach, since if this principle is naturally included in the problems of real systems, then using concurrency may yield the more detailed and simpler model, reflecting even the sophisticated features of the system.

Once the model is designed, it is necessary to use it for finding properties and features of the modelled system or use this abstraction for checking the behaviour, safeness or harmless of the system. Developing of the formal analysis methods is then one of the first questions which arise when the new formal model is introduced.

Many models of representation of concurrent or communicating processes were introduced by many authors and investigated more or less detailed. The most interesting ones are Petri nets, transition systems and process algebras. Although using other points of view, based on others foundations and definitions, the spirits of these works can be found to have more in common.

This paper is focused on Petri nets. Several definitions of Petri nets were introduced. The growing importance of Petri nets is related to their usefulness in modelling concurrent systems, distributed processing, communicating systems, all of whose behaviour can be easily expressed as movement of tokens in Petri nets.

There have been developed several main analysis methods. The basic method for investigating of the properties of Petri nets is using the state graph or coverability tree. The methods begin by constructing the state graph, which represents the dynamical behaviour of the Petri net as a transition system, in general infinite.

* Department of Computer Science and Engineering, Božetěchova 2, CZ-612 66 Brno

The powerful and elegant analytical techniques for Petri nets are based on linear algebra. These methods include the investigation of invariants, which reflect particular aspects of the repetitive behaviour of the net.

From this is clear, that a particular Petri net has got features, that some of them can be taken as a behaviour of a particular transition system, and some of them reflect the relationships between members of a vector space in the sense of linear transformation.

The process of investigating properties can be identified as a finding the morphisms which assign to the net the appropriate transition system or vector space, preserving some aspects of state transition behaviour.

In this paper, the notion of morphism was approached from a categorical point of view. In following lines is described an attempt to associate Petri nets with the objects of a category of structured objects, trying to choose the arrows (morphisms) to express the dynamics of the net and link this to transition systems in categorical way.

2 Main Notions of Category Theory

2.1 Definitions

The main idea of the paper is to link another models together. For that reason it is necessary to introduce the notions of categories and functors.

Definition: A *category* **A** consists of a set of *objects* (called *obj*(**A**)) and the set of *morphisms* or *arrows* (called *arr*(**A**)). The objects are usually denoted as

$$A, B, C, ..., X, Y, ...$$

and the morphisms are usually denoted

$$f, g, h, ..., \alpha, \beta, ...$$

Further:

(1) Each morphism has a designated *domain* and *codomain* in *obj*(**A**). When the *domain* of f is A and the *codomain* of f is B we write $f: A \rightarrow B$.

(2) Given morphisms $f: A \rightarrow B$ and $g: B \rightarrow C$, there is a designated *composite* morphism

$$g \bullet f: A \rightarrow C.$$

(3) Given any object A there is a designated identity morphism $1_A: A \rightarrow A$.

(4) The data above are required to satisfy the following:

Identity laws. If $f: A \rightarrow B$ then

$$1_B \bullet f = f \qquad \text{and} \qquad f \bullet 1_A = f.$$

Associative law. If $f: A \rightarrow B$, $g: B \rightarrow C$ and $h: C \rightarrow D$ then

$$h \bullet (g \bullet f) = (h \bullet g) \bullet f.$$

Although the category is described in terms of functions, it is important to realise that the notion of category is axiomatically defined and formal. The objects of a category need not be actual sets and the arrows need not be actual functions. As is usual with axiomatic definitions this allows great flexibility in the construction of categories. This flexibility is crucial in the application of categories to computer science.

The principal axiom on a category is the associativity of composition. This axiom means we can leave out brackets in a composition of arrows. For example we can write $l \bullet g \bullet h \bullet i \bullet j \bullet l$ instead of $(l \bullet((g \bullet h) \bullet i) \bullet (j \bullet l))$.

In category theory we usually operate with diagrams. Every arrow has its representation in diagram. So, for example arrow f: $S \rightarrow T$ should be drawn like this:

$$S \xrightarrow{\ f\ } T.$$

Having two categories, there is a possibility to define functor from one category to another. A functor from a category $\boldsymbol{A}$ to a category $\boldsymbol{B}$ is a homomorphism which preserves identities and composition. It preserves the structure that a category has. So the functor is the structure-preserving map between categories. We give its formal definition.

Definition: A *functor* F: $\boldsymbol{A} \rightarrow \boldsymbol{B}$ is a pair of functions F_0: $obj(A) \rightarrow obj(B)$ and F_1: $arr(A) \rightarrow arr(B)$ for which

(1) If f: $A \rightarrow B$ in $\boldsymbol{A}$, then $F_1(f)$: $F_0(A) \rightarrow F_0(B)$ is in $\boldsymbol{B}$.

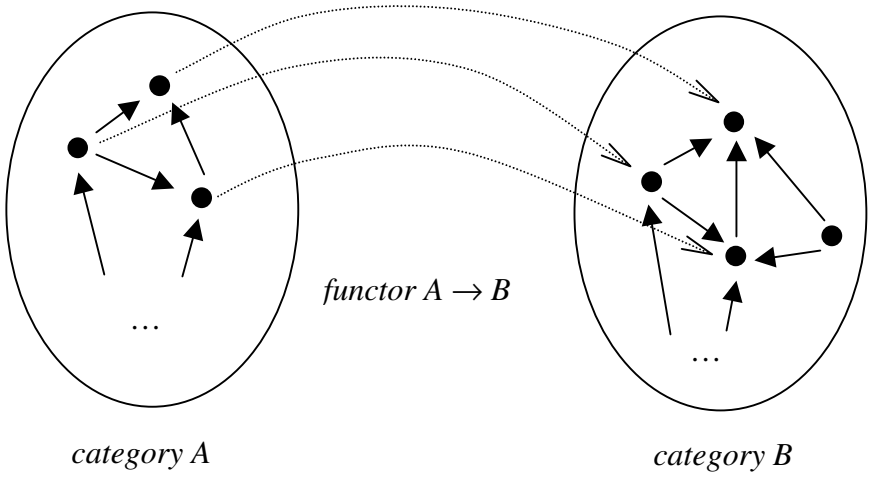
(2) For any object A of $\boldsymbol{A}$, $F_1(1_A) = 1_{F0(A)}$.

(3) If $g \bullet f$ is defined in $\boldsymbol{A}$, then $F_1(g) \bullet F_1(f)$ is defined in $\boldsymbol{B}$ and $F_1(g \bullet f) = F_1(g) \bullet F_1(f)$.

The notation is usually overloaded. We use symbol F for both functions F_0 and F_1. If A is an object, $F(A) = F_0(A)$ is an object, and if f is an arrow, then $F(f) = F_1(f)$ is an arrow.

2.2 Systems as categories: the idea

The idea of the categorical approach is expressed in the following figure.



Consider two categories, say A and B. The objects in categories are shown as dots in the picture. The structure of a category is done by arrows between objects. Assume, that the dots are state systems of a same type (Petri nets, dynamical systems, transition systems, etc.) and that the arrows are behaviour-preserved morphisms (e.g. one system reflects a part of the behaviour of the another, …). We define the functor from A to B to be a morphism from objects of category A to objects of category B preserving the structure of a category. This means, if there is an arrow between two objects in category A, there has to exist the appropriate arrow between images of the objects in category B.

In other words, the functor links together the systems with the similar behaviour, although both systems are expressed in different ways (Petri nets and transition systems, …). The functionality is preserved, the relation to other object too, but the formalism used for describing such kind of dynamical behaviour has changed. This can be seen as a translation between different kinds of dynamical systems. The functor associates two really different description of the same dynamics (in the sense of behaviour preservation morphisms) together. We deal with the more abstract expression of the model.

In the following chapters we describe the typical and important categories for our work, category of transition systems and category of Petri nets. The relationship between both structures is very well know, but we will demonstrate the advance of formal description using categories. For the formal introduction we will need to remark the definitions.

3 Transition systems

3.3 Basic definitions

A transition system is a machine which can react to external or internal events. The action is done by changing of state of the system. The systems changing their state during time are common modelling abstraction for many real systems.

Of course, transition systems can be made into a category.

Definition: A *transition system* is a quadruple $A = \langle S, T, \alpha, \beta \rangle$, where

(1) S .. is a finite or infinite set of states;

(2) T .. is a finite or infinite set of transitions;

(3) α .. is mapping α: $T \to S$, which take each transition t in T to the source state $\alpha(t)$ in S;

(4) β .. is mapping β: $T \to S$, which take each transition t in T to the target state $\beta(t)$ in S;

A transition t with source s and target s' is written t: $s \to s'$. Several transitions can have the same source and target, i.e. the product mapping $\langle \alpha, \beta \rangle$: $T \to S \times S$ is not necessarily injective. In general the transition system is infinite.

The description of the similarity in the behaviour of the transition system is expressed as transition systems homomorphism.

Definition: Let $A_1 = \langle S_1, T_1, \alpha_1, \beta_1 \rangle$ and $A_2 = \langle S_2, T_2, \alpha_2, \beta_2 \rangle$ be two transition systems. A *homomorphism* $h: A_1 \to A_2$ is a pair $\langle \sigma, \tau \rangle$ of mappings:

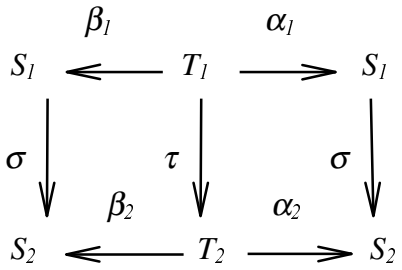
$$\sigma: S_1 \to S_2,$$
$$\tau: T_1 \to T_2.$$

which satisfies the compatibility conditions:

$$\forall t \in T_1.$$
$$\alpha_2(\tau(t)) = \sigma(\alpha_1(t)),$$
$$\beta_2(\tau(t)) = \sigma(\beta_1(t)).$$

Preparing the definition for a categorical approach the compatibility conditions may be stated in the form that the following diagram commutes:

Structure preservation law



3.4 Category of transition systems

The category of transition systems has the following objects and arrows.

Objects are transition systems.

Arrows are all homomorphisms between transition systems.

The composition of arrows should be defined as element by element composition of functions, i.e. for every composable arrows $f = \langle \sigma_f, \tau_f \rangle : A \to B$ and $g = \langle \sigma_g, \tau_g \rangle : B \to C$ is the composition arrow $g \bullet f$ defined $g \bullet f = \langle \sigma_g \bullet \sigma_f, \tau_g \bullet \tau_f \rangle$. The associative law holds for functions, so the laws comes true in this case either. The existence of identity arrows is obvious.

Very easily can the definition be extended to the labelled or initialised transition systems.

4 Petri nets

4.5 Basic notions

Although the definitions of Petri nets were introduced many times in the literature, we recall the main point and introduce the used symbols.

Definitions: 4-tuple $N = \langle P, T, F, W \rangle$ is called *a net* (in some books *Petri net*), where

(1) P .. is a finite set of *places*;

(2) T .. is a finite set of *transitions*, such that $P \cap T = \emptyset$;

(3) F .. is *a flow relation*: $F \subseteq (P \times T) \cup (T \times P)$, such that

$$\forall t \in T.\ \exists p_1, p_2 \in P.\ (p_1, t) \in F \text{ and } (t, p_2) \in F;$$

This later guarantees the existence of two important functions:

(3a) $Pre: T \to \wp(P)$ such that: $\forall t \in T.\ p \in Pre(t) \Leftrightarrow (p, t) \in F$ and

(3b) $Post: T \to \wp(P)$ such that: $\forall t \in T.\ p \in Post(t) \Leftrightarrow (t, p) \in F$.

(4) W ..is *a weight function for arcs*, $W: F \to \mathbf{N}^+$.

A function $M: P \to \mathbf{N}$ is called *marking* of Petri net N, $\mathbf{N}$ is a set of natural numbers, $\wp(P)$ is a power set of the set P. The dynamical behaviour of a Petri net is introduced as usual (enabled transitions, firing of a transition).

Firing of transition t from marking M to marking M' is usually written as:

$$M\ [t > M'.$$

Now, we extend the definition to marked Petri nets.

Definition: *Marked net* is 5-tuple $N = \langle P, T, F, W, M \rangle$, where

(1) $\langle P, T, F, W \rangle$ is a net;

(2) M .. is nonempty *set of markings,* $M = \{M_1, M_2, ..., M_n\}$, satisfying

if $M_1 \in M$, such that $M_1 [t > M_2$ for some $t \in T$ then $M_2 \in M$ (M is said to be *T-closed*).

The dynamics is expressed by next-state function as follows:

Definition: The *next-state function* is function δ:

$$\delta: M_S \times T \to M_S, \text{ such that}$$

$$\forall t \in T.\ \forall M, M' \in M_S.\ \delta(M, t) = M' \ \Leftrightarrow\ M [t > M'.$$

4.6 Homomorphisms of Petri nets

Petri nets were introduced as powerful models of real systems, with their syntactic structure built from primitives: place, transition and arc, and with dynamical behaviour based on tokens and markings providing an operational semantics. The structure may be made into a category when some types of morphisms between Petri nets are introduced.

Definition: Let $N_1 = \langle P_1, T_1, F_1, W_1, M_1 \rangle$ and $N_2 = \langle P_2, T_2, F_2, W_2, M_2 \rangle$ be two marked nets. The *marked Petri net homomorphism* f: $N_1 \to N_2$, is the triple $f = \langle \pi, \tau, \mu \rangle$, where

$$\pi: P_1 \to P_2, \qquad \tau: T_1 \to T_2, \qquad \mu: M_1 \to M_2,$$

the π, τ satisfy the structure preserving law as follows:

$$\forall t \in T_1: \pi^*(Pre_1(t)) = Pre_2(\tau(t)),$$

$$\pi^*(Post_1(t)) = Post_2(\tau(t)).$$

The dynamical behaviour of the net is preserved in the sense:

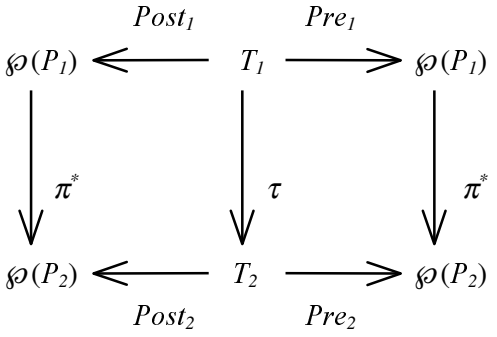
$$\forall \langle m, t \rangle.\ m \in M_1 \text{ and } t \in T_1$$

$$\mu(\delta_1(m, t)) = \delta_2(\mu(m), \tau(t)),$$

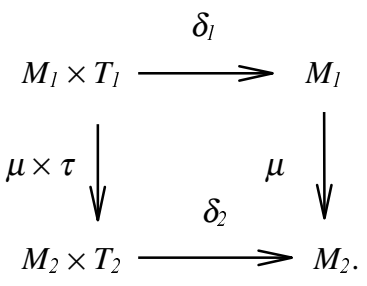
δ_1, δ_2 being next-state functions for marked nets N_1 and N_2.

Categorically, the definitions are equivalent to saying that the following diagrams commutes:

Preserving of structure



Preserving of dynamics



4.7 Categories of Petri nets

Objects are marked Petri nets.

Arrows are marked Petri nets homomorphisms.

Composition of arrows is defined as componentwise composition of function tuples:

$$\text{if } f = \langle \pi_f, \tau_f, \mu_f \rangle : A \to B,\ g = \langle \pi_g, \tau_g, \mu_g \rangle : B \to C, \text{ then}$$

$$g \bullet f : A \to C,\ g \bullet f = \langle \pi_g \bullet \pi_f, \tau_g \bullet \tau_f, \mu_g \bullet \mu_f \rangle.$$

It is easy to prove that there exists an identity arrow, which has as all terms of the tuple identity functions, and that admissible arrows are composable, which means composition of two morphisms in every category of Petri nets produces a morphism.

5 Petri nets and transition systems

We have introduced transition systems and Petri nets and their homomorphisms. Both concepts can be expressed easily in a categorical way, obtained categories of transition systems and categories of Petri nets. It is interesting that both categories have many points of resemblance, arising from the fact that Petri nets behave dynamically as transition systems. Every Petri net is connected with one transition system. This system is its state space together with its next-state function.

To find corresponding (underlying) transition system, reflecting the behaviour of the net is easy. Let $PN = \langle P, T, F, W, M \rangle$ be marked Petri net with the next-state function δ_{PN}. The underlying transition system is then $TS = \langle M, T_T, \alpha, \beta \rangle$, where M becomes the set of states of transition system *TS*.

Set of transitions of *TS* is determined as the biggest subset T_T of the cartesian product $M \times T$, such that for every element in T_T the next-state function δ_{PN} is defined, this means $T_T = \{\langle m, t \rangle \mid t \text{ is } m\text{-enabled}\}$.

Definitions of α, β are also based on next-state function δ_{PN} of Petri net *PN*. For every pair $\langle m, t \rangle$, $m \in M$ and $t \in T_T$, $\alpha(t) = m$ and $\beta(t) = \delta_{PN}(m, t)$. Values $\alpha(t)$ and $\beta(t)$ are unique for particular t.

An assignment of Petri net to its behaviour expressed by transition system is possible to express as equivalence of next-state functions in the sense that the next-state functions are equivalent. For both, Petri nets and transition systems the next-state function may be found.

We introduce the next-state functions for transition systems. This function is defined along the definition of functions α, β such way:

$$\delta_{TS}(\alpha(t), t) = \beta(t).$$

This means, this function is defined only as partial function on the set $S \times T$, where S is set of states and T is set of transitions. When we define the subset $H \subseteq S \times T$, such that $\langle s, t \rangle \in H$ if and only if δ_{TS} is defined, then δ_{TS} is totaly defined on H. H is then a set $H = \{\langle \alpha(t), t \rangle \mid \text{for every } t\}$ See now $|H| = |T|$, since for every transition t there is exactly one pair $\langle s, t \rangle$ when δ_{TS} defined.

The next-state function for Petri nets is a bit different. $\delta_{PN}(m, t)$, $m \in M$, $t \in T$ is defined if and only if t is m-enabled. δ_{PN} is partially defined on $M \times T$. Call E such subset of $M \times T$, which contains all pairs $\langle m, t \rangle$, $m \in M$, $t \in T$, such that $\delta_{PN}(m, t)$ is defined. As said before $E = \{\langle m, t \rangle \mid t \text{ is } m\text{-enabled}\}$. It is *not true* that $|E| = |T|$. One particular transition can be enabled for several markings, so there in general exist several pairs $\langle m_i, t \rangle$ for particular t, such that δ_{PN} is defined.

It is not possible to define the state space of Petri nets as a transition system only just that next-state functions are equivalent. The next-state function for transition system has an unique name for every transition. To construct from δ_{PN} the next-state function of underlying transition system, named δ_{TS}, is necessary to find an unique representation for every transition in Petri net state space.

This condition is satisfied, when the set of transition of δ_{TS} is taken a set D, which contains all pairs $<m, t>$, $m \in M$, $t \in T$, such that t is m-enabled, i.e. $D = \{<m, t> \mid t \text{ is } m\text{-enabled}\}$. δ_{TS} is then partially defined on $M \times D$ this way:

$$\delta_{TS}: M \times D \to M,\ \delta_{TS}(m, <m, t>) = \delta_{PN}(m, t).$$

Of course this equation is valid if and only if δ_{PN} is defined. Otherwise δ_{TS} is undefined too. δ_{TS} then determines α and β such way:

$$\alpha(<m, t>) = m,\ \beta(<m, t>) = \delta_{TS}(m, <m, t>) = \delta_{PN}(m, t).$$

6 Conclusion

This paper shows the possibility of an categorical description of Petri nets and other state-based models. This may be useful for finding stronger connections between different models in order to enable using common formal analysis methods for several models. This paper describes the similarity and connection between Petri nets and transition systems, introducing the formal definition of homomorphism. Although the relationship between Petri nets and transition system is well known, it is a first step for finding other connections to other models, which have not been revealed yet. It sounds promising the connection of Petri nets and linear dynamical systems, that might allow to use more methods from linear algebra in the world of Petri nets, not only methods based on invariants. This may reveal the new field of formal analysis methods for models using Petri nets and can help in investigation of more interesting but more sophisticated features of models. This technique is important for modelling and simulation too. The understanding of similarities and differences between several models used in modelling and simulation should make clearer the application field, where an particular type of models get the biggest benefit or reveal the non suitable use of some models in appropriate problems of real word.

This work has been supported by the Grant Agency of Czech Republic grants No.102/98/0552 "Research and Applications of Heterogenous Models".

References

1. Arbib M. A., Manes E. G.: *Arrows, Structures and Functors: The Categorical Imperative*, Academic Press, Inc. (London) Ltd., 1975.
2. Arnold A.: *Finite Transition Systems*, Prentice Hall (UK), Masson (F), 1992.
3. David R., Alla H.: *Petri Nets and Grafcet*, Prentice Hall (UK), 1992.
4. Lawvere F. W., Schanuel S. H.: *Conceptual Mathematics: A first introduction to Categories*, Cambridge University Press, 1997 (translated from Italian edition 1991).
5. Walters R.F.C.: *Categories and Computer Science*, Cambridge University Press, 1991.