DOI: 10.2478/ijasitels-2023-0001

A Categorical Metamodel for Reactive Kripke Frames

Daniel C. Crăciunean[1]

[1]*Computer Science and Electrical and Electronics Engineering Department, Faculty of Engineering, "Lucian Blaga" University of Sibiu, Romania*
daniel.craciunean@ulbsibiu.ro

Abstract

In a reactive Kripke model, evaluation of logical operators can cause reconfiguration of the model in which the formula is evaluated. Therefore, in reactive Kripke frames, the evaluation of a logical formula in a world depends both on the world in which the evaluation is made and on the worlds it has passed through previously. The result is an extended semantics, which can specify a class of modal logics more comprehensive than the class specified by ordinary Kripke frames. This paper introduces a metamodel for reactive Kripke frames, based on the concept of categorical sketch. We believe that the categorical sketch is an appropriate metamodel for specifying a Kripke frame model.

Keywords: modal logic, Kripke frames, reactive Kripke frames, categorical sketch

1 Introduction

Modal logic extends classical logic with denoted modal operators, usually with the symbols ☐ and ◊, which express the way in which a logical formula is satisfied. The interpretation of modal operators depends on the concept of truth that we want to specify through logical formulas. Thus, some modalities that have been formalized in modal logic include: alethic or truth modalities, where ☐ has the meaning of "necessary" and ◊ has the meaning of "possible"; deontic modalities where ☐ has the meaning of "mandatory" and ◊ has the meaning of "permitted"; epistemic modality or modes of knowledge where ☐ has the meaning of "it is known that" or modes of belief where ☐ has the meaning of "it is believed that". Real applications define many other interpretations of these symbols. In general, the operators ☐ and ◊ are not independent but are linked by the relation $\Diamond\varphi=\neg\Box\neg\varphi$.

Formulas from modal logic cannot be interpreted on the basis of truth tables. For example, for ☐p, where p is an atomic proposition, we cannot define a truth table because when p is true in one interpretation, it does not follow that p is true in all interpretations. The reference semantic model for modal logic is the "possible worlds" model, introduced by Saul Kripke in 1959. A Kripke frame is a structure consisting of a set S of possible worlds and a binary relation R on the set S, in other words a Kripke frame is a graph whose nodes are possible worlds and whose arcs represent the accessibility of each world to other possible worlds.

In 2004, D. Gabbay introduced a concept of reactivity which assumes that reactive frames are relational structures in which the structure of a frame, at a given moment, is determined not only by the world in which it is located, but also by previous worlds [4]. In other words, each transition, from one world to another, can modify the relational structure of the frame.

The specification of such reactive structures is based on graphs endowed with several types of arrows, single arrows, double arrows and sometimes triple arrows. Double arrows connect nodes with other arrows, or arrows between them [4, 8]. In this approach, the double arrows have the role of alternately changing the state of the target arrows from active to inactive and vice versa during the transitions of the relational structure. This process is cumulative and results in the dynamic transformation of the relational structure of the system.
This idea of dynamic reconfiguration of a relational structure has applications in various modeling fields such as the extension of modal logic to reactive modal logic, the extension of context-independent grammars to reactive context-independent grammars, the definition of reactive automata and other reactive structures [5, 7, 6, 10].
In the relational semantics of modal logic, different worlds are considered in which the values of the propositions change depending on the values of the propositional variables in these worlds. Therefore, the value of a sentence depends exclusively on the accessible worlds. But in a Kripke model the accessible worlds are fixed. In a reactive Kripke model, evaluation of logical operators can cause reconfiguration of the model in which the formula is evaluated. In [4], this reconfiguration is specified by double arrows, which enable or disable other arrows.
The modeling of reactive Kripke frames is based on a type of special graphs that undergo certain transformations when its edges are traversed and which are called reactive graphs. Therefore, in reactive Kripke frames, the evaluation of a logical formula in a world depends both on the world in which the evaluation is made and on the worlds it has passed through previously. The result is an extended semantics, which can specify a class of modal logics more comprehensive than the class specified by ordinary Kripke frames [11].
This paper introduces a metamodel for reactive Kripke frames, based on the concept of categorical sketch. We believe that the categorical sketch is an appropriate metamodel for specifying a metamodel for Kripke frames.
Section 2 contains some general notions and notations used in the rest of the paper. Section 3 presents the static dimension of the Kripke frame models, section 4 presents the reactive dimension of the reactive Kripke frame models, and section 5 concludes the paper with some conclusions.

2 General notions and notations

The basic modal propositional logic language is built on the vocabulary formed by a set P, of atomic propositions, to which are added the usual operators from propositional logic and two unary modal operators, which we denote by □ and ◊. In the modal logic of truth, the operator □ is read "necessary", and ◊ is read "possible". In the BNF notation, well-formed formulas from modal logic are defined as follows:

<Wff>::=<Proposition>|(¬ <Wff>) |
(<Wff > ∧ < Wff >) |(<Wff > ∨ < Wff >) | (<Wff > → < Wff >) |
(<Wff > ↔ < Wff >) |□ (<Wff >) | ◊(<Wff >)

where <Proposition> represents any atomic proposition from the set P and each occurrence of <Wff> represents a well-formed formula. We denote the set of well-formed formulas, with W.

A Kripke model consists of a set S of possible worlds, a binary relation, of accessibility, R on the set S and an application $L:S\rightarrow 2^P$, which associates to each possible world $s\in S$, a set of well-formed formulas, satisfied in the world s. The pair F=(S, R), is a graph called a Kripke frame

We will denote the satisfaction relation of a well-formed formula ϕ in a world $s\in S$ with $s\Vdash\phi$. If we have a Kripke model M=(S, R, L), and a world $s\in S$, then the satisfaction for a well-formed formula ϕ is defined recursively [2] as described below.

For operators from classical logic, the satisfaction condition is that L(s) be a model for the formula φ, that is:

"$s\Vdash p$ iff $p\in L(s)$, where p is an atomic proposition; $s\Vdash\neg\varphi$ iff $s\nVdash\varphi$;

$s\Vdash\varphi \wedge \psi$ iff $s\Vdash\varphi$ and $s\Vdash\psi$; $s\Vdash\varphi \vee \psi$ iff $s\Vdash\varphi$, or $s\Vdash\psi$;

$s\Vdash\varphi \rightarrow \psi$ iff $s\Vdash\varphi$ implies $s\Vdash\psi$; $s\Vdash\varphi \leftrightarrow \psi$ iff ($s\Vdash\varphi$ iff $s\Vdash\psi$)";

and for modal operators, satisfaction is defined as follows:

"$s\Vdash\Box\psi$ iff, $\forall\ s_1\in S$ with $R(s, s_1)$, it results $s_1\Vdash\psi$;

$s\Vdash\Diamond\psi$ iff $\exists s_1 \in S$ with, $R(s, s_1)$ and $s_1\Vdash\psi$."

As we can see, the satisfaction of the formula $\Box\psi$ is conditioned by the satisfaction of ψ in all accessible worlds in s, and the satisfaction of the formula $\Diamond\psi$ is conditioned by the satisfaction of ψ in at least one accessible world in s. This observation tells us that the validity of a modal logic formula, depends to a great extent on the accessibility relation R, i.e., on the graph corresponding to the Kripke model. Therefore, the validity of a modal formula can be realized or cancelled by an appropriate transformation of this graph.

The evaluation of logical formulas is done starting from a set of formulas or schemes of logical formulas that we impose as true, in the context of a model, and which are called axioms. The validity of these formulas, in a Kripke model, can be imposed by conditions on the graph that represents the relation R on the set S of possible worlds.

In this paper, we will enforce the realization of these axioms or axiom schemes, at the metamodel level, using the concept of a categorical sketch. We consider that the categorical sketch is a metamodel capable of specifying all the constraints necessary to specify a Kripke frame, customized and appropriate to the concrete requirements of practical applications.

A categorical sketch is a graph that represents a metamodel together with a series of constraints on the models represented by this metamodel.

In general, this metamodel can be approached in any category, but in this paper, we will use the Graph category, which has graphs as objects and graph homomorphisms as arrows. The graph component of the sketch specifies the structural dimension of the models. The nodes of this graph are typed and most often represent concepts of the model.

The sketch constraints are represented by predicate symbols that together with their signatures form the concept of diagram predicate signature [15, 16]. The predicate symbols together with the logical dependencies between them form a category that we denote by $\boldsymbol{\Pi}$.

If $\boldsymbol{\Pi}$ is a category of predicates and dependencies, then a functor $\boldsymbol{\alpha}:\boldsymbol{\Pi}\rightarrow$Graph is called a graph signature. Each object $\boldsymbol{\alpha}(P)$, $P\in\boldsymbol{\Pi}_0$, is called the shape graph arity of P. Each arrow $\boldsymbol{\alpha}(r)$, $r\in\boldsymbol{\Pi}_1$, is a morphism in the Graph category: $\boldsymbol{\alpha}(r):\boldsymbol{\alpha}(P_1)\rightarrow\boldsymbol{\alpha}(P_2)$, where $P_1,P_2\in\boldsymbol{\Pi}_0$, and is called signature substitution [12].

A diagram is a functor $d:\boldsymbol{\mathcal{P}}\rightarrow \boldsymbol{\mathcal{G}}$, where $\boldsymbol{\mathcal{P}},\boldsymbol{\mathcal{G}}\in Graph_0$. The domain $\boldsymbol{\mathcal{P}}$ of a diagram d is called a shape graph. If $\boldsymbol{\mathcal{G}}\in Graph_0$, then a $\boldsymbol{\Pi}$-formula, over $\boldsymbol{\mathcal{G}}$, is a pair (P,d), where P is a predicate $P \in \boldsymbol{\Pi}_0$, and d is a diagram $d:\boldsymbol{\alpha}P \rightarrow \boldsymbol{\mathcal{G}}$. We will denote a $\boldsymbol{\Pi}$-formula (P,d), more simply, by P(d). We denote the set of $\boldsymbol{\Pi}$-formulas over the graph $\boldsymbol{\mathcal{G}} \in Graph_0$, by $Fm(\boldsymbol{\Pi},\boldsymbol{\mathcal{G}})$. Therefore $Fm(\boldsymbol{\Pi},\boldsymbol{\mathcal{G}})=\{P(d) \mid P \in \boldsymbol{\Pi},\ d \in Graph(\boldsymbol{\alpha}(P), \boldsymbol{\mathcal{G}})\}$. The set of formulas $Fm(\boldsymbol{\Pi},\boldsymbol{\mathcal{G}})$, together with the inference relation form a category [12], If there is no confusion we will call a $\boldsymbol{\Pi}$-formula, simply a formula.

A generalized sketch over a signature $\boldsymbol{\Pi}$, is made up of a graph $\boldsymbol{\mathcal{G}} \in Graph_0$, and a subcategory of $\boldsymbol{\Pi}$-formulas $\boldsymbol{\mathcal{T}} \subseteq Fm(\boldsymbol{\Pi},\boldsymbol{\mathcal{G}})$, closed to the inference, i.e., it satisfies the condition: if $P(d) \in \boldsymbol{\mathcal{T}}$ then also $Q(d(\boldsymbol{\alpha}(r))) \in \boldsymbol{\mathcal{T}}$. Therefore, a generalized sketch is a tuple $\boldsymbol{\mathcal{S}}=(\boldsymbol{\mathcal{G}},\boldsymbol{\mathcal{T}})$, where the graph $\boldsymbol{\mathcal{S}}$, specifies the structure of a model, and the objects of the category $\boldsymbol{\mathcal{T}}$ are constraints on the models imposed by formulas.

To define an instance of a generalized sketch we will use the slice category [1]. If $\boldsymbol{\mathcal{G}}$ is an object $\boldsymbol{\mathcal{G}} \in Graph_0$, the slice category, which we denote by $Graph{\downarrow}\boldsymbol{\mathcal{G}}$, has as objects, pairs $(\boldsymbol{\mathcal{H}}, \tau)$, where $\boldsymbol{\mathcal{H}} \in Graph_0$ and $\tau \in Graph_1$ is a morphism $\tau:\boldsymbol{\mathcal{H}} \rightarrow \boldsymbol{\mathcal{G}}$, and as arrows between each two objects $(\boldsymbol{\mathcal{H}}_1, \tau_1)$, $(\boldsymbol{\mathcal{H}}_2, \tau_2)$, a morphism from Graph $\varphi:\boldsymbol{\mathcal{H}}_1 \rightarrow \boldsymbol{\mathcal{H}}_2$.

If $\boldsymbol{\mathcal{S}}=(\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{T}})$, is a sketch, and $\boldsymbol{\mathcal{T}}$ is the empty category, i.e. it does not contain any constraints, then for any object $\tau:\boldsymbol{\mathcal{H}} \rightarrow \boldsymbol{\mathcal{G}}$ from the slice category $Graph{\downarrow}\boldsymbol{\mathcal{G}}$, the domain $\boldsymbol{\mathcal{H}}$ of τ is an instance of the sketch $\boldsymbol{\mathcal{S}}$. We denote by $\boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}})$, the set of all these instances.

To define the instances subject to certain constraints, we will use the concept of pullback. For a cospan(φ,ψ), we denote the corresponding pullback by $PLB(\varphi,\psi)$. The pullback $PLB(\varphi,\psi)$, is a span that we denote by $span(\varphi^*,\psi^*)$, with the property that $\varphi \circ \varphi^* = \psi \circ \psi^*$. We also denote $\varphi^*=PLB_{\psi}(\varphi)$ and $\psi^*=PLB_{\varphi}(\psi)$.

We notice that if we have two instances $\boldsymbol{\mathcal{H}}_1,\boldsymbol{\mathcal{H}}_2 \in \boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}})$, corresponding to the objects $(\boldsymbol{\mathcal{H}}_1,\tau_1),(\boldsymbol{\mathcal{H}}_2,\tau_2) \in ob(Graph{\downarrow}\boldsymbol{\mathcal{G}})$, then $dom(PLB(\tau_1,\tau_2))=dom(\tau_1^*,\ \tau_2^*)$, is also an instance, $dom(PLB(\tau_1,\tau_2)) \in \boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}})$, because there is a morphism $\tau^*= \tau_1 \circ \tau_1^* = \tau_2 \circ \tau_2^*$.

Now we can define an instance $\boldsymbol{\mathcal{H}} \in \boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}})$, which satisfies a formula P(d). If $\boldsymbol{\mathcal{S}}=(\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{T}})$, is a generalized sketch, then an instance $\boldsymbol{\mathcal{H}} \in \boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}})$, corresponding to the object $(\boldsymbol{\mathcal{H}},\tau)$, from the $Graph{\downarrow}\boldsymbol{\mathcal{G}}$ category, satisfies the formula $P(d) \in \boldsymbol{\mathcal{T}}_0$, if there is a another instance $\boldsymbol{\mathcal{H}}_1 \in \boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}})$, corresponding to the object $(\boldsymbol{\mathcal{H}}_1,\tau_1)$, from the $Graph{\downarrow}\boldsymbol{\mathcal{G}}$ category, so that $\boldsymbol{\mathcal{H}}=dom(PLB(\tau_1,d))$. We denote the set of instances that satisfy the formula P(d) by $\boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}},P(d))$.\

We can now define an instance of a sketch. If $\boldsymbol{\mathcal{S}}=(\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{T}})$, is a generalized sketch, then an instance $\boldsymbol{\mathcal{H}} \in \boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}})$, is the instance of the sketch $\boldsymbol{\mathcal{S}}$, if $\boldsymbol{\mathcal{H}}$ satisfies all the formulas $P(d) \in \boldsymbol{\mathcal{T}}_0$. We denote by $\boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{S}})$ the set of all instances of the sketch $\boldsymbol{\mathcal{S}}$.

3 The static dimension of the model

In the modelling of real systems, the set S of possible worlds overlaps, most of the time, with the set of states of the model, characterized by logical formulas [13, 3, 14]. Also, in such models, the transition from one state to another is done by the actions of an agent who acts in order to achieve certain objectives specified by logical formulas that must be satisfied.

In this context, a transition system evolves through the actions of an agent who seeks to achieve some objectives expressed through logical formulas. Usually, the agent acts on the basis of a plan that involves the execution of several successive actions. When there are several plans, which lead to the achievement of the objectives, the agent must choose the optimal plan based on a logic that characterizes it [3, 14].

Modal logics are characterized by schemes of axioms which are schemes of logical formulas, assumed to be true without demonstration, and which are the basis of reasoning. Thus an important scheme of formulas, assumed to be true in almost all modal logics, is "$\Box(\varphi\rightarrow\psi) \wedge \Box\varphi\rightarrow\Box\psi$", called axiom K, after Saul Kripke, who introduced the Kripke models [2, 9].

A series of other axioms, which were imposed in modal logic, received names such as: axiom T "$\Box\varphi\rightarrow\varphi$"; axiom B "$\varphi\rightarrow\Box\Diamond\varphi$"; axiom D "$\Box\varphi\rightarrow\Diamond\varphi$"; axiom 4 "$\Box\varphi\rightarrow\Box\Box\varphi$" and axiom 5 "$\Diamond\varphi\rightarrow\Box\Diamond\varphi$". Based on these names of the axioms, modal logics were also named over time. Similarly, many modal logics have the prefix of the name K, which expresses the satisfaction of the axiom K, followed by the names of the other axioms. For example, KT45 logic, characterized by axioms T, 4 and 5, which is also sometimes called S5 modal logic and is used to reason about knowledge.

As we saw in section 2, the evaluation of a formula, in modal logic, largely depends on the graph structure of the Kripke model, i.e. the Kripke frame. It is demonstrated that in a Kripke model M=(S,R,L), there is an implicit correspondence between the relation R and the satisfaction of the modal logic formulas [2, 14]. Thus, axiom K is satisfied in any Kripke frame, axiom T is satisfied if and only if R is reflexive, axiom B is satisfied if and only if R is symmetric, axiom D is satisfied if and only if R is serial, axiom 4 is satisfied if and only if R is transitive, axiom 5 is satisfied if and only if R is Euclidean, and so on. Therefore, if we want, for example, an agent to rationalize in KT45 logic, we will have to set the condition that the Kripke model according to which it evaluates the logical formulas is equipped with a relation R, reflexive, transitive and Euclidean.

All these restrictions on the relation R can be expressed by logical predicates. Therefore, the axioms of modal logic can be expressed by logical predicates. Since a Kripke frame involves the concepts of world and relationship, the predicates will have parameters of these types, that is, the world type, which we denote by w, and the relation type, which we denote by ρ. Also, if $u,v\in w$, and $u\rho v$, we will denote this fact with $(u,v)\in\rho$.

With these notations, the axiom T can be expressed by the predicate: $PT(u,v)=(u\in w)\rightarrow(u,u)\in\rho$.

Axiom B is satisfied if and only if the predicate: $PB(u,v)=(u\in w)\wedge(v\in w)\wedge(u,v)\in\rho\rightarrow (v,u)\in\rho$, is satisfied.

Axiom D, is satisfied if and only if the predicate: $PD(u,v)=(u\in w)\rightarrow(\exists v\in w)\wedge(u,v)\in\rho$, is satisfied.

Axiom 4 is satisfied if and only if the predicate: $P4(u,v)=(u\in w)\wedge(v\in w)\wedge(t\in w)\wedge ((u,v)\in\rho)\wedge((v, t)\in\rho) \rightarrow (u,t)\in\rho$, is satisfied.

Axiom 5 is satisfied if and only if the predicate: $P5(u,v)=(u\in w)\wedge(v\in w)\wedge(t\in w)\wedge ((u,v)\in\rho)\wedge((u, t)\in\rho) \rightarrow (v,t)\in\rho$, is satisfied.

In this way, we can impose other logical formulas on the Kripke frame, specific to the logic we want to impose on an agent. For example, the formula "$\Box\varphi \leftrightarrow \Diamond\varphi$" is satisfied if and only if the Kripke frame is functional, i.e. if and only if it satisfies the predicate $PF(u,v)=(u\in w\rightarrow(\exists!v\in w)\wedge(u,v)\in\rho$, and the formula "$\Box(\varphi\wedge\Box\varphi\rightarrow\psi)\vee\Box(\psi\wedge\Box\psi\rightarrow\varphi)$" is satisfied if and only if the Kripke frame is linear, i.e. if and only if it satisfies the predicate :

$PL(u,v)=(u\in w)\wedge(v\in w)\wedge(t\in w)\wedge((u,v)\in\rho)\wedge((u,t)\in\rho)\rightarrow((v,t)\in\rho)\vee(v=t)\vee((t,v)\in\rho)$.

When we want to specify a Kripke model, we will include in the set of objects of the $\boldsymbol{\Pi}$ category, all the predicates that represent formulas that we want to impose on the model. For example, if we want an agent to reason in a KTB4 logic, we will include in the set of objects of the $\boldsymbol{\Pi}$ category, the predicates PT, PB and P4. We will denote this set of objects by Π_0^i, and we will call them atomic formulas.

We then define the category $\boldsymbol{\Pi}$, inductive, as follows:

i) If $P\in\Pi_0^i$, then $P\in\boldsymbol{\Pi}_0$.

ii) If $P_1,P_2\in\boldsymbol{\Pi}_0$ then $P=\neg(P_1)$, $P=(P_1)\wedge(P_2)$, $P=(P_1)\vee(P_2)$, $P=(P_1)\rightarrow(P_2)$, $P=(P_1)\leftrightarrow(P_2)\in\boldsymbol{\Pi}_0$, and the dependencies $P_1\dashv P$, $P_2\dashv P \in \boldsymbol{\Pi}_1$.

iii) If $P_1\in\boldsymbol{\Pi}_0$, then $P= (\forall xP_1)$, $P=(\exists xP_1)\in\boldsymbol{\Pi}_0$, and the dependencies $P_1\dashv P\in\boldsymbol{\Pi}_1$.

iv) Any object in $\boldsymbol{\Pi}_0$, and any arrow in $\boldsymbol{\Pi}_1$, are obtained by successively applying, a finite number of times, rules i), ii) and iii).

For example, in the case of KTB4 logic, the predicates: $PTB_1(u,v)=PT(u,v)\wedge PB(u,v)$, $PTB_2(u,v)=PT(u,v)\vee PB(u,v)$, $PTB4_1(u,v)=PT(u,v)\wedge PB(u,v)\wedge P4(u,v)$, will be contained in the set of objects $\boldsymbol{\Pi}_0$, and the dependencies $PT(u,v)\dashv PTB_1(u,v)$, $PB(u,v)\dashv PTB_1(u,v)$, $PT(u,v)\dashv PTB_2(u,v)$, $PB(u,v)\dashv PTB_2(u,v)$, $PT(u,v)\dashv PTB4_1(u,v)$, $PB(u,v)\dashv PTB4_1(u,v)$, $P4(u,v)\dashv PTB4_1(u,v)\in\boldsymbol{\Pi}_1$.

Since a Kripke frame has only one type of node, namely type w, and only one type of arrow, namely ρ, all the predicates defined above have as shape graph, a graph $\boldsymbol{\mathcal{P}}$, with a single node and a single arrow on which we denote, as in the case of parameters, with w and ρ respectively. Therefore, the shape graph arity application $\boldsymbol{\alpha}:\boldsymbol{\Pi}\rightarrow$Graph, is defined as $\boldsymbol{\alpha}(w)=w$ and $\boldsymbol{\alpha}(\rho)=\rho$, for all predicates in $\boldsymbol{\Pi}_0$.

Let's now build the categorical sketch corresponding to a Kripke frame. As we defined it, in section 2, a generalized sketch is a tuple $\boldsymbol{\mathcal{S}}=(\boldsymbol{\mathcal{G}}, \boldsymbol{\mathcal{T}})$, where the graph $\boldsymbol{\mathcal{S}}$, specifies the structure of a model and the category $\boldsymbol{\mathcal{T}}$, specifies the constraints imposed on the models by formulas.

In a categorical sketch, the graph $\boldsymbol{\mathcal{G}}$ has the role of classifying the concepts in a model, by specifying the types of concepts and the relationships between them. In the case of the Kripke frame model, we have only two concepts, namely; the world concept, which we denote with $\boldsymbol{w}$, and the relation concept, which we denote with $\boldsymbol{r}$. We also have a single diagram $d:\boldsymbol{\mathcal{P}}\rightarrow\boldsymbol{\mathcal{G}}$, defined as follows: $d(w)=\boldsymbol{w}$ and $d(\rho)=\boldsymbol{r}$.

In this context, the objects of category $\boldsymbol{\mathcal{T}}$ are generated by subsets of the set: {PT(d), PB(d), PD(d), P4(d), P5(d), PF(d), PL(d)}, in accordance with the definition of category $\boldsymbol{\Pi}$, from above. Of course, this set can be expanded with other formulas specific to the concrete model we want to build. Notice that the set of objects $\boldsymbol{\mathcal{T}}_0$ contains all well-formed formulas based on the constraints imposed on the model by the set $\boldsymbol{\Pi}_0$ of predicates. Therefore, there is a formula $P(d)\in\boldsymbol{\mathcal{T}}_0$, so that the constraints of the specific Kripke frame model can only be imposed by satisfying the formula P(d).

For example, if we want the relation R, of the model to be an equivalence relation, it is enough to impose the satisfaction of the formula $P(d)=PT(d)\wedge PB(d)\wedge P4(d)$, i.e. to be reflexive, symmetric and transitive.

But an instance $\boldsymbol{\mathcal{H}}\in\boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}})$, corresponding to the object $(\boldsymbol{\mathcal{H}},\tau)$, from the category Graph$\downarrow\boldsymbol{\mathcal{G}}$, satisfies the formula $P(d)\in\boldsymbol{\mathcal{T}}_0$, if there is another instance $\boldsymbol{\mathcal{H}}_1\in\boldsymbol{\mathcal{I}}(\boldsymbol{\mathcal{G}})$, corresponding to the object $(\boldsymbol{\mathcal{H}}_1, \tau_1)$, from the Graph$\downarrow\boldsymbol{\mathcal{G}}$ category, so that $\boldsymbol{\mathcal{H}}=dom(PLB(\tau_1,d))$.

Next, we will show that any instance of the sketch is represented by a fixed point of an endofunctor of the Graph$\downarrow\mathcal{G}$ category. For this we define an endofunctor Φ:Graph$\downarrow\mathcal{G}\rightarrow$Graph$\downarrow\mathcal{G}$, like this: for each object $(\mathcal{H},\tau)\in$Graph$\downarrow\mathcal{G}$, $\Phi((\mathcal{H},\tau))=$ (dom(PLB(τ,d), $\tau\circ\tau^*$), and the arrows between two objects $\Phi((\mathcal{H}_1, \tau_1)$ and $\Phi(\mathcal{H}_2, \tau_2)$ are the arrows from the Graph$\downarrow\mathcal{G}$ category, between the objects dom(PLB(τ_1,d)), dom(PLB(τ_2,d)) $\in$ob(Graph$\downarrow\mathcal{G}$).
Since the pullback never adds new components to a graph, but selects all the components that respect the constraints imposed on the model, it follows that if $\mathcal{H}$ is a model of the sketch, represented by the object $(\mathcal{H},\tau)\in$Graph$\downarrow\mathcal{G}$, then $\Phi((\mathcal{H},\tau))=(\mathcal{H},\tau)$ that is, $\mathcal{H}$ is represented by a fixed point of the endofunctor Φ. Obviously, if the graph $\mathcal{H}$, has components that do not satisfy the constraints P(d), these components will be eliminated and therefore $(\mathcal{H}, \tau)$ is not a fixed point for the endofunctor Φ.
In the case of systems modelling, when the set of possible worlds overlaps with the set of states, the set of states is given by the possible evolutions of the system and therefore the set of possible worlds is fixed. Also, the set of transitions forms a minimal relation on the set of possible worlds that must be included in the relation of any instance of the Kripke frame of the model. From here it follows that any instance of the Kripke frame model does nothing but extend the relation R to a minimal relation that respects the constraints. We also note that the graph of the sketch $\mathcal{G}$ is a terminal object in the Graph category, from which it follows that each instance, of the sketch, is represented by a single object $(\mathcal{H},\tau)$, from the Graph$\downarrow\mathcal{G}$ category. These observations lead us to the conclusion that, in the case of systems modelling, we can determine a minimal relationship on the set of system states that satisfies the constraints imposed on the model.

4 The reactive dimension of the model

When there are several plans, which lead to the achievement of the objectives, the agent must choose the optimal plan based on some values [3]. Therefore, the agent must be able to carry out reasoning in various logics, to fulfil the objectives, depending on the values it is pursuing.
The process of tracking the achievement of some objectives, through different types of reasoning, involves the dynamic adaptation of the Kripke frame in which the logical formula is evaluated. A Kripke model is reactive if the model instance changes as it is traversed to evaluate a formula. This means that the constraints applied to an instance change during the evaluation of a logical formula.
We consider a transition system that evolves through the actions of an agent. The actions of the agent are decided according to the realization of some logical formulas. Thus, if we have a finite set of propositional variables P={p,q,...}, a transition system is a construct of the form TS=(S,A,T,L), where S is the set of states, A is the set of actions that the agent can perform, T is a set of transitions, $t\in S\times A\times S$, between states by executing some actions, and L is an application $L:S\rightarrow 2^P$, which selects the propositions in P, which are satisfied in each state $s\in S$.
In this transition system, in each state, the choice of an action by the agent is conditioned by the realization of a formula from the modal logic. The evaluation of these formulas is done in a Kripke frame in which the set of possible worlds overlaps with the set S of states, and the accessibility relation is given by the transition relation.

In a classic Kripke model, the accessibility relationship does not change in the evaluation process, but in the reactive model, this relationship adapts dynamically, depending on the mode of reasoning that the agent will adopt in the respective context.

It is obvious that the set of possible worlds will overlap with the set of possible states of the transition system also in the reactive case and only the accessibility relation will be adapted. Also, the accessibility relation defined by the transitions of the system will be included in all instances of the reactive Kripke model, because the imposition of additional constraints involved the expansion of this relation. We will denote by $F_0=(\mathcal{W},\mathcal{R}_0)$, the Kripke frame defined by a transition system as follows: $\mathcal{W}$=S and for each pair $u,v\in\mathcal{W}$, we make $\mathcal{R}_0(u,v)$=true if there is a transition $(u,a,v)\in T$, $a\in A$ and $\mathcal{R}_0(u,v)$=false, otherwise. Obviously, $F_0\in\in\mathcal{J}(\mathcal{G})$, where $\mathcal{G}$, is the graph of the sketch $\mathcal{S}$. We will call the relation $\mathcal{R}_0$, the initial relation of the Kripke frame instance.

Example 4.1. If we consider the transition system from Fig. 1, where P={p,q}, then the initial instance F_0, is defined as follows (Fig. 2): $\mathcal{W}=\{S_1,S_2,S_3,S_4\}$,

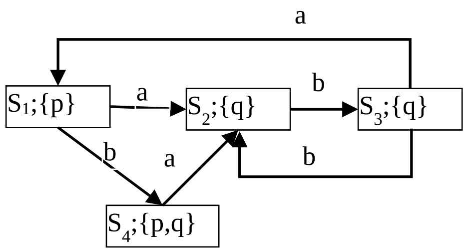

Fig. 1. Example, transition system

$\mathcal{R}_0(S_1,S_2)$=true;

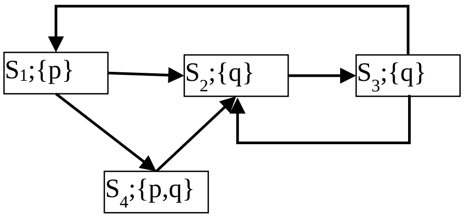

Fig. 2 Example, the initial instance

$\mathcal{R}_0(S_1,S_4)$=true; $\mathcal{R}_0(S_2,S_3)$=true; $\mathcal{R}_0(S_3,S_1)$=true; $\mathcal{R}_0(S_3,S_2)$=true; $\mathcal{R}_0(S_4,S_2)$=true and otherwise $\mathcal{R}_0(S_i,S_j)$=false.

Let's evaluate the formula $S_1|=\Box^2q$, in $F_0=(\mathcal{W},\mathcal{R}_0)$. In order for this formula to be satisfied in S_1, the subformula $\Box q$ will have to be satisfied in all accessible states from S_1, that is, the formulas: $S_2|=\Box q$, $S_4|=\Box q$ must be satisfied. In order for $S_2|=\Box q$ to be satisfied, q must be satisfied in state S_3, which is true, and for $S_4|=\Box q$ to be satisfied, q must be satisfied in S_2, which is true. Therefore, F_0 satisfies the formula $S_1|=\Box^2q$.

Let us now consider a set of predicates $P_1,P_2,\ldots P_n\in\Pi_0$, which specify constraints on a Kripke frame $F_0=(\mathcal{W},\mathcal{R}_0)$. Then for each predicate P_i, we have an instance $F_i=(\mathcal{W},\mathcal{R}_i)$, which satisfies the constraint imposed by this predicate. As we saw, in section 3, each instance $\mathcal{R}_i$ is represented by a fixed point of the functor $\Phi:\text{Graph}\downarrow\mathcal{G}\rightarrow\text{Graph}\downarrow\mathcal{G}$.

In the Reactive Kripke frame model, we need to impose restrictions specified by a predicate P_i, i=1,n, or by a conjunction of such predicates, when we want to impose several such constraints simultaneously.

It is easy to prove that if the relation $\mathcal{R}_i$, imposes the constraint P_i, and the relation $\mathcal{R}_j$, imposes the constraint P_j, then the relation $\mathcal{R}_{i,j}= \mathcal{R}_i \vee \mathcal{R}_j$, imposes the condition $P_i \wedge P_j$. Therefore, if we know the relations $\mathcal{R}_i$, i=1,n, we can easily build any relation that is a disjunction of known relations. We will denote by $\mathcal{R}$, the set of all possible relations in a Kripke reagent.
Therefore, the set $\mathcal{R}$, of all possible relations in a reactive Kripke frame, is defined recursively as follows:
i) $\mathcal{R}_0 \in \mathcal{R}$ and if $P_i \in \Pi_0$, then $\mathcal{R}_i \in \mathcal{R}$;
ii) If $\mathcal{R}_i$, $\mathcal{R}_j \in \mathcal{R}$ then $\mathcal{R}_i \vee \mathcal{R}_j \in \mathcal{R}$.

At each step of evaluating a modal formula, one of the relations from the set of relations $\mathcal{R}$ will be selected, depending on the path travelled to the current world. The set of possible paths is a language on the set of possible worlds $\mathcal{L} \subseteq \mathcal{W}^*$.
Therefore, a Kripke frame instance is a construct of the form $F_R = (\mathcal{W}, \mathcal{R}, \mathcal{L}, \Psi)$, where $\mathcal{W}$ is the set of possible worlds, $\mathcal{R}$ is the set of possible relations, $\mathcal{L}$ is a language on the set $\mathcal{W}$, and $\Psi: \mathcal{L} \rightarrow \mathcal{R}$ is an application that selects the appropriate relationship for each possible path. In this context, if we denote by δ the path through which the world s was reached, the evaluation of the modal operators will be done as follows:
„$s \Vdash \Box_\delta \psi$ iff, $\forall$ $s_1 \in \mathcal{W}$ with $\Psi(\delta)(s, s_1)$, it results $s_1 \Vdash \psi$;
$s \Vdash \Diamond_\delta \psi$ iff $\exists$ $s_1 \in \mathcal{W}$ with $\Psi(\delta)(s, s_1)$ and $s_1 \Vdash \psi$".

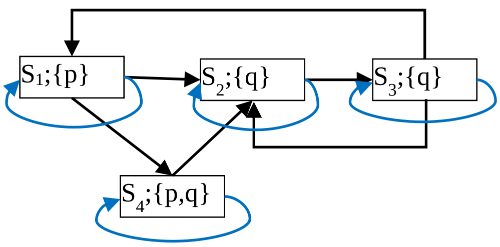

Fig. 3. Example, reflexive instance

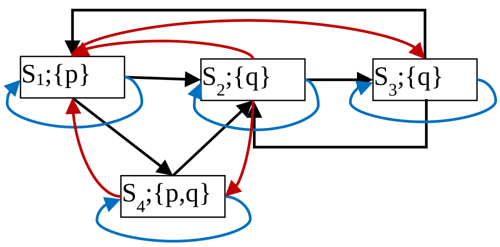

Fig. 4. Example, symmetrical instance

Example 4.2. If in example 4.1, we impose the restriction specified by the PT(u,v) predicate, then the resulting relation, which we denote by $\mathcal{R}_1$, becomes a reflexive relation (Fig.3): $\mathcal{R}_1(S_1,S_2)$=true; $\mathcal{R}_1(S_1,S_4)$=true; $\mathcal{R}_1(S_2,S_3)$=true; $\mathcal{R}_1(S_3,S_1)$=true; $\mathcal{R}_1(S_3,S_2)$=true; $\mathcal{R}_1(S_4,S_2)$=true; $\mathcal{R}_1(S_1,S_1)$=true; $\mathcal{R}_1(S_2,S_2)$=true; $\mathcal{R}_1(S_3,S_3)$=true; $\mathcal{R}_1(S_4,S_4)$=true; and otherwise $\mathcal{R}_1(S_i,S_j)$=false.
The relation corresponding to the predicate PB(u,v), which we denote by $\mathcal{R}_2$, becomes a symmetrical relation (Fig.4): $\mathcal{R}_2(S_1,S_2)$=true; $\mathcal{R}_2(S_1,S_4)$=true; $\mathcal{R}_2(S_2,S_3)$=true; $\mathcal{R}_2(S_3,S_1)$=true; $\mathcal{R}_2(S_3,S_2)$=true; $\mathcal{R}_2(S_4,S_2)$=true; $\mathcal{R}_2(S_1,S_1)$=true; $\mathcal{R}_2(S_2,S_2)$=true; $\mathcal{R}_2(S_3,S_3)$=true; $\mathcal{R}_2(S_4,S_4)$=true; $\mathcal{R}_2(S_2,S_1)$=true; $\mathcal{R}_2(S_1,S_3)$=true; $\mathcal{R}_2(S_4,S_1)$=true; $\mathcal{R}_2(S_2,S_4)$=true; and otherwise $\mathcal{R}_2(S_i,S_j)$=false.
By imposing the constraints specified by the predicates PT(u,v) and PB(u,v), together we obtain a symmetric and transitive relation $\mathcal{R}_{1,2} = \mathcal{R}_1 \vee \mathcal{R}_2$, and therefore $\mathcal{R}_= = \{\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_{1,2}\}$. Let's define the application Ψ, as follows: $\Psi(S_1)= \mathcal{R}_0$; $\Psi(S_1S_2)= \Psi(S_1S_1)=\mathcal{R}_1$; $\Psi(S_1S_4)= \mathcal{R}_2$; $\Psi(\delta)= \mathcal{R}_{1,2}$ otherwise.
We notice that if we evaluate the formula $S_1 |= \Box^2 q$, using the new Kripke frame $F_R=(\mathcal{W}, \mathcal{R}, \mathcal{L}, \Psi)$, it is no longer satisfied, because reflexivity requires that the sentence q should also be satisfied in the state S_1.

5 Conclusions

The most important conclusion is that category theory offers all the necessary mechanisms for the specification and analysis of diagrammatic models, including classical or reactive Kripke models. We can find, from section 3, that based on some atomic predicates that impose constraints on Kripke structures, we can then easily construct any constraint, formulated in terms of these atomic constraints.

In the case of systems modeling, the set of possible worlds overlaps with the set of possible states, and therefore the set of possible worlds is also determined and the set of transitions forms a minimal relation on the set of possible worlds that must be included in the relation of any instance of the Kripke frame of the model.

Any instance of the Kripke frame model, extends the relationship given by the evolution of the system, to a minimal relationship that respects the constraints. These observations lead us to the conclusion that, in the case of systems modeling, we can determine a minimal relationship on the set of system states that satisfies the constraints imposed on the model. This subject will be treated in a future work where we intend to provide algorithms for the effective calculation of reactive Kripke frames for semantic models expressed in terms of transition systems.

References

[1] Michael Barr, Charles Wells, 2012. Category Theory For Computing Science, Reprints in Theory and Applications of Categories, No. 22.

[2] M. Huth, M. Ryan, 2004, Logic in Computer Science, Modelling and Reasoning about Systems, Published in the United States of America by Cambridge University Press, New York.

[3] Luo, Jieting; Liao, Beishui; Gabbay, Dov. 2022, Value-based Practical Reasoning: Modal Logic + Argumentation (In Press). In: COMMA - Computational Models of Argument. Conference Proceedings.

[4] D. Gabbay. 2004, Reactive Kripke semantics. In Proceedings of CompLog 2004, W. Carnielli, ed., pp. 7–20. Centre of Logic and Computation, University of Lisbon.

[5] H. Barringer and D. M. Gabbay. 2010, Modal and temporal argumentation networks. In Time for Verification. Essays in Memory of Amir Pnueli, D. Peled and Z. Manna, eds., pages 1–25. LNCS 6200, Springer, Berlin.

[6] M. Crochemore and D. M. Gabbay. 2011, Reactive Automata. Information and Computation, 209(4), 692–704. Published online: DOI: 10.1016/j.ic.2011.01.002.

[7] S. Modgil. 2009, Reasoning about preferences in argumentation frameworks. Artif. Intell., 173:901–934.

[8] Gabbay, D.M. 2013, Reactivity and Grammars: An Exploration. In: Reactive Kripke Semantics. Cognitive Technologies. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41389-6_9.

[9] Olivier Gasquet , Andreas Herzig , Bilal Said ,François Schwarzentruber, 2014, Kripke's Worlds, An Introduction to Modal Logics via Tableaux, Springer Basel AG.

[10] Barringer, H., Rydeheard, D., Gabbay, D. 2014. Reactivity and Grammars: An Exploration. In: Dershowitz, N., Nissan, E. (eds) Language, Culture, Computation. Computing - Theory and Technology. Lecture Notes in Computer Science, vol 8001. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-45321-2_6

[11] Gabbay, D.M. 2008. Introducing Reactive Kripke Semantics and Arc Accessibility. In: Avron, A., Dershowitz, N., Rabinovich, A. (eds) Pillars of Computer Science. Lecture Notes in Computer Science, vol 4800. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-78127-1_17

[12] Zinovy Diskin, Uwe Wolter, 2008, A Diagrammatic Logic for Object-Oriented Visual Modeling, Electronic Notes in Theoretical Computer Science, Volume 203, Issue 6, 21 November 2008.

[13] Virginia Dignum, 2009, A Logic for Agent Organizations, Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models, Information Science Reference, by IGI Global.

[14] Y. Shoham, K. Leyton-Brown, 2009, Multiagent systems_ algorithmic, game-theoretic, and logical foundations -Cambridge University Press.

[15] Uwe Wolter, Zinovy Diskin, 2015, The Next Hundred Diagrammatic Specification Techniques, A Gentle Introduction to Generalized Sketches, 02 September 2015 : https://www.researchgate.net/publication /253963677.

[16] D.C. Crăciunean, D. Karagiannis, 2019, A categorical model of process cosimulation, Journal of Advanced Computer Science and Applications(IJACSA), 10(2).