A Categorical Approach to Simulations*

Miguel Palomino[1], José Meseguer[2], and Narciso Martí-Oliet[1]

[1] Departamento de Sistemas Informáticos, Universidad Complutense de Madrid
[2] Computer Science Department, University of Illinois at Urbana-Champaign

Abstract. Simulations are a very natural way of relating concurrent systems, which are mathematically modeled by Kripke structures. The range of available notions of simulations makes it very natural to adopt a categorical viewpoint in which Kripke structures become the objects of several categories while the morphisms are obtained from the corresponding notion of simulation. Here we define in detail several of those categories, collect them together in various institutions, and study their most interesting properties.

1 Introduction

Simulations are a very natural way of relating concurrent systems. They are particularly useful for proving temporal logic properties, because we can use simulations to *shift our ground*; that is, to prove that a system $\mathscr{A}$ satisfies a property φ by considering a perhaps much simpler system (for example a finite-state abstraction) $\mathscr{B}$, proving that $\mathscr{B}$ satisfies φ, and showing that $\mathscr{B}$ *simulates* $\mathscr{A}$. This transfer result holds for the universal fragment ACTL* of CTL*, and in particular for all linear temporal logic formulas. Similarly, we may want to prove that a possibly more complex but more efficient concurrent system $\mathscr{C}$ is a *correct implementation* of another system $\mathscr{A}$; again this amounts to showing that $\mathscr{A}$ simulates $\mathscr{C}$, and will then allow transferring all ACTL* properties already established for $\mathscr{A}$ to its implementation $\mathscr{C}$. Obviously, the more flexibly we can shift our ground by means of suitable simulations, the more easily we can reason about concurrent systems, their abstractions, and their implementations. There are therefore good practical reasons to look for the *most general* notions of simulation possible, as a way to support very general and flexible reasoning methods.

The point of this paper is to systematically exploit a *categorical* point of view in the quest for general notions of simulation. That is, we consider increasingly more general categories whose objects are Kripke structures, and whose morphisms are adequate simulations between them. There are several orthogonal dimensions along which simulations can be generalized as discussed in detail in [13,11]. We can extend them: (1) from functions to relations; (2) from strictly preserving state predicates to only doing so in a looser way; (3) from simulations in which one step is simulated by another to "stuttering" simulations in which several steps in one system can correspond to several steps in the other; and (4) from the case in which all systems we relate share the same

* Research supported by ONR Grant N00014-02-1-0715, NSF Grant CCR-0234524, by DARPA through Air Force Research Laboratory Contract F30602-02-C-0130, and by the Spanish CICYT projects MELODIAS TIC 2002-01167 and MIDAS TIC 2003–0100.

J.L. Fiadeiro et al. (Eds.): CALCO 2005, LNCS 3629, pp. 313–330, 2005.

set AP of atomic predicates to one in which systems with different atomic predicates can be related among each other. All these extensions (1)–(4), and their possible combinations are mathematically characterized in this paper by increasingly more general categories.

A theme running in parallel with such generalizations is characterizing corresponding sets of *temporal logic formulas* that can be "reflected" by (that is, lifted along) increasingly more general simulation maps. This is closely related to another theme also developed in detail, namely the different *temporal logic institutions* involved. Indeed, Kripke structures are the most frequently used models for temporal logic. From an institutional viewpoint we will expect, for a given signature, a corresponding *category* of Kripke structures, which is precisely what we are investigating. The point then is that different choices of increasingly more general categories give rise to a corresponding family of temporal logic institutions, for which we study under what conditions the amalgamation property (semi-exactness) holds.

Another theme also studied in detail is the issue of *categorical constructions*, including limits, colimits, and epi-mono factorizations. As far as we know, most of the constructions we give are new. They shed further light on Kripke structures and the morphisms that we have available for relating them.

An extended version of this paper can be found at `http://maude.sip.ucm.es/~miguelpt/papers/cap.pdf`.

2 Kripke Structures and Simulations

When reasoning about computational systems, it is convenient to abstract from as many details as possible by means of simple mathematical models that can be used to reason about them. For a state-based system we can represent its behavior by means of a *transition system*, which is a pair $\mathscr{A} = (A, \rightarrow_{\mathscr{A}})$ with A a set of states and $\rightarrow_{\mathscr{A}} \subseteq A \times A$ a binary relation called the transition relation. A transition system, however, does not include any information about the relevant properties of the system. In order to reason about such properties it is necessary to add information about the atomic properties that hold in each state. In what follows, we assume a fixed set AP of atomic propositions and define a *Kripke structure* as a triple $\mathscr{A} = (A, \rightarrow_{\mathscr{A}}, L_{\mathscr{A}})$, where $(A, \rightarrow_{\mathscr{A}})$ is a transition system with $\rightarrow_{\mathscr{A}}$ a *total* relation (this is a customary requirement, which simplifies the semantics of the temporal logic), and $L_{\mathscr{A}} : A \rightarrow \mathscr{P}(AP)$ is a labeling function associating to each state the set of atomic propositions that hold in it.

We use the notation $a \rightarrow_{\mathscr{A}} b$ to state that $(a,b) \in \rightarrow_{\mathscr{A}}$. A *path* in $\mathscr{A}$ is a function $\pi : \mathbb{N} \longrightarrow A$ such that, for each $n \in \mathbb{N}$, $\pi(n) \rightarrow_{\mathscr{A}} \pi(n+1)$.

To specify system properties we use the logic $\mathrm{ACTL}^*(AP)$, which is a sublogic of the branching-time temporal logic $\mathrm{CTL}^*(AP)$ (see for example [5, Sect. 3.1]). There are two types of formulas in $\mathrm{CTL}^*(AP)$: state formulas, denoted by $\mathrm{State}(AP)$, and path formulas, denoted by $\mathrm{Path}(AP)$. The semantics of the logic, specifying the satisfaction relations $\mathscr{A}, a \models \varphi$ and $\mathscr{A}, \pi \models \psi$ for a Kripke structure $\mathscr{A}$, an initial state $a \in A$, a state formula φ, a path π, and a path formula ψ, is defined as usual [5]. $\mathrm{ACTL}^*(AP)$ is the restriction of $\mathrm{CTL}^*(AP)$ to those formulas such that their negation-normal forms (with negations pushed to atoms) do not contain any existential path quantifiers. To avoid

introducing existential quantifiers implicitly, it is more convenient to restrict ourselves to the negation-free fragment ACTL$^*\backslash\neg(AP)$ of ACTL$^*(AP)$, defined as follows:[1]

$$\begin{array}{ll} \text{state formulas:} & \varphi = p \in AP \mid \top \mid \bot \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \mathbf{A}\psi \\ \text{path formulas:} & \psi = \varphi \mid \psi \vee \psi \mid \psi \wedge \psi \mid \mathbf{X}\psi \mid \psi\mathbf{U}\psi \mid \psi\mathbf{R}\psi \mid \mathbf{G}\psi \mid \mathbf{F}\psi. \end{array}$$

We write State$\backslash\neg(AP)$ and Path$\backslash\neg(AP)$ for the sets of state and path formulas in ACTL$^*\backslash\neg(AP)$, respectively. When working with stuttering simulations, we also use ACTL$^* \setminus \mathbf{X}$, respectively ACTL$^*\backslash\{\neg, \mathbf{X}\}$, for the fragment of the logic without the operator $\mathbf{X}$, respectively $\mathbf{X}$ and $\neg$.

2.1 Generalized Stuttering Simulations

In general, we are not only interested in the study of isolated systems, but would also like to be able to study their interrelationships. To do that we introduce a very general notion of simulation in increasingly more general steps; in a first step, we slightly extend the simulations in [5] (which essentially correspond to our *strict* simulations). Examples of simulations can be found in [13,12].

Given transition systems $\mathscr{A} = (\mathscr{A}, \rightarrow_{\mathscr{A}})$ and $\mathscr{B} = (\mathscr{B}, \rightarrow_{\mathscr{B}})$, a *simulation of transition systems* $H : \mathscr{A} \longrightarrow \mathscr{B}$ is a binary relation $H \subseteq A \times B$ such that if $a \rightarrow_{\mathscr{A}} a'$ and aHb then there is b' such that $b \rightarrow_{\mathscr{B}} b'$ and $a'Hb'$. A *map of transition systems* H is a simulation such that H is a function. If both H and H^{-1} are simulations, then we call H a *bisimulation*. We can extend a simulation of transition systems H to paths by defining $\pi H \rho$ if $\pi(n) H \rho(n)$ for each $n \in \mathbb{N}$.

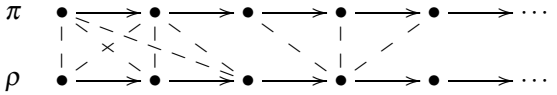
Similarly, for Kripke structures $\mathscr{A} = (A, \rightarrow_{\mathscr{A}}, L_{\mathscr{A}})$ and $\mathscr{B} = (B, \rightarrow_{\mathscr{B}}, L_{\mathscr{B}})$ over the same set AP of atomic propositions, an *AP-simulation* $H : \mathscr{A} \longrightarrow \mathscr{B}$ of $\mathscr{A}$ by $\mathscr{B}$ is given by a simulation $H : (A, \rightarrow_{\mathscr{A}}) \longrightarrow (B, \rightarrow_{\mathscr{B}})$ between the underlying transition systems such that if aHb, then $L_{\mathscr{B}}(b) \subseteq L_{\mathscr{A}}(a)$. We say that H is an *AP-map* if its underlying simulation of transition systems is a map. We call H *strict* if aHb implies $L_{\mathscr{B}}(b) = L_{\mathscr{A}}(a)$. Also, we call H an *AP-bisimulation* if H and H^{-1} are AP-simulations.

Simulations of transition systems and of Kripke structures *compose* and the identity function $1_{\mathscr{A}} : \mathscr{A} \longrightarrow \mathscr{A}$ is trivially a simulation of transition systems and of Kripke structures. Therefore, transition systems together with their simulations define a category **TSys** with corresponding subcategories for maps and bisimulations. Similarly, Kripke structures together with AP-simulations define a category $\mathbf{KSim}_{AP}$, with two corresponding subcategories $\mathbf{KMap}_{AP}$ and $\mathbf{KBSim}_{AP}$ whose morphisms are, respectively, AP-maps and AP-bisimulations. Of course, there is also a subcategory $\mathbf{KSim}^{\mathrm{str}}_{AP}$ of strict AP-simulations, and corresponding subcategories $\mathbf{KMap}^{\mathrm{str}}_{AP}$ and $\mathbf{KBSim}^{\mathrm{str}}_{AP} = \mathbf{KBSim}_{AP}$. Note that if H is an isomorphism in $\mathbf{KSim}_{AP}$ then it must be a map and a bisimulation. Note, finally, that the mapping $(A, \rightarrow_{\mathscr{A}}, L_{\mathscr{A}}) \mapsto (A, \rightarrow_{\mathscr{A}})$ extends to a forgetful functor $\mathbf{TS} : \mathbf{KSim}_{AP} \longrightarrow \mathbf{TSys}$, with corresponding restrictions to the appropriate subcategories.

The definition of simulation can be extended by allowing the presence of stuttering [3,14,10]. For $\mathscr{A} = (A, \rightarrow_{\mathscr{A}})$ and $\mathscr{B} = (B, \rightarrow_{\mathscr{B}})$ transition systems and $H \subseteq A \times B$

[1] $\mathbf{X}$, $\mathbf{G}$, and $\mathbf{F}$ stand for the classic *next* ($\bigcirc$), *henceforth* ($\Box$), and *eventually* ($\Diamond$) LTL operators.

a relation, we say that a path ρ in $\mathscr{B}$ *H-matches* a path π in $\mathscr{A}$ if there are strictly increasing functions $\alpha, \beta : \mathbb{N} \longrightarrow \mathbb{N}$ with $\alpha(0) = \beta(0) = 0$ such that, for all $i, j, k \in \mathbb{N}$, if $\alpha(i) \leq j < \alpha(i+1)$ and $\beta(i) \leq k < \beta(i+1)$, it holds that $\pi(j) H \rho(k)$. For example, the following diagram shows the beginning of two matching paths, where related elements are joined by dashed lines and $\alpha(0) = \beta(0) = 0$, $\alpha(1) = 2$, $\beta(1) = 3$, $\alpha(2) = 5$.



Then, a *stuttering simulation of transition systems* $H : \mathscr{A} \longrightarrow \mathscr{B}$ is a binary relation $H \subseteq A \times B$ such that if aHb, then for each path π in $\mathscr{A}$ starting at a there is a path ρ in $\mathscr{B}$ starting at b that H-matches π. If H is a function we say that H is a *stuttering map of transition systems*. If both H and H^{-1} are stuttering simulations, then we call H a *stuttering bisimulation*. Stuttering simulations of transition systems compose [10] and together with transition systems define a category that we denote **STSys** and which contains **TSys** as subcategory.

Given Kripke structures $\mathscr{A} = (A, \rightarrow_{\mathscr{A}}, L_{\mathscr{A}})$ and $\mathscr{B} = (B, \rightarrow_{\mathscr{B}}, L_{\mathscr{B}})$ over AP, a *stuttering AP-simulation* $H : \mathscr{A} \longrightarrow \mathscr{B}$ is a stuttering simulation of transition systems $H : (A, \rightarrow_{\mathscr{A}}) \longrightarrow (B, \rightarrow_{\mathscr{B}})$ such that if aHb then $L_{\mathscr{B}}(b) \subseteq L_{\mathscr{A}}(a)$. We call it *strict* if aHb implies $L_{\mathscr{B}}(b) = L_{\mathscr{A}}(a)$. Again, stuttering AP-simulations compose and define a category $\mathbf{KSSim}_{AP}$ with corresponding subcategories of strict and stuttering AP-maps.

We can generalize simulations even further by allowing them to relate Kripke structures over different sets of atomic propositions. This provides a very flexible way of relating Kripke structures and will allow us to gather all the previous categories $\mathbf{KSSim}_{AP}$, for different choices of AP, into a single one.

Given a function $\alpha : AP \longrightarrow \mathrm{State}(AP')$ and a Kripke structure $\mathscr{A} = (A, \rightarrow_{\mathscr{A}}, L_{\mathscr{A}})$ over AP', we define the *reduct Kripke structure* $\mathscr{A}|_\alpha = (A, \rightarrow_{\mathscr{A}}, L_{\mathscr{A}|_\alpha})$ over AP, with labeling function $L_{\mathscr{A}|_\alpha}(a) = \{p \in AP \mid \mathscr{A}, a \models \alpha(p)\}$. α is extended in the expected, homomorphic way to formulas $\varphi \in \mathrm{CTL}^*(AP)$, replacing each atomic proposition p occurring in φ by $\alpha(p)$; we denote this extension by $\overline{\alpha}(\varphi)$.

Proposition 1. *Let $\alpha : AP \to \mathrm{State}(AP')$ be a function and φ be a formula in $\mathrm{CTL}^*(AP)$. Then, for all Kripke structures $\mathscr{A} = (A, \rightarrow_{\mathscr{A}}, L_{\mathscr{A}})$ over AP', states $a \in A$, and paths π:*

- *if φ is a state formula, $\mathscr{A}, a \models \overline{\alpha}(\varphi) \iff \mathscr{A}|_\alpha, a \models \varphi$, and*
- *if φ is a path formula, $\mathscr{A}, \pi \models \overline{\alpha}(\varphi) \iff \mathscr{A}|_\alpha, \pi \models \varphi$.*

The definition of generalized stuttering simulations is now immediate. For Kripke structures $\mathscr{A}$ over a set AP of atomic propositions and $\mathscr{B}$ over a set AP', a *stuttering simulation* (resp. *strict stuttering simulation*) $(\alpha, H) : (AP, \mathscr{A}) \longrightarrow (AP', \mathscr{B})$ consists of a function $\alpha : AP \longrightarrow \mathrm{State}\backslash\{\neg, \mathbf{X}\}(AP')$ (resp. $\alpha : AP \longrightarrow \mathrm{State} \setminus \mathbf{X}(AP')$) and a stuttering AP-simulation (resp. strict stuttering AP-simulation) $H : \mathscr{A} \longrightarrow \mathscr{B}|_\alpha$.

To simplify notation, from now on we will write $(\alpha, H) : \mathscr{A} \longrightarrow \mathscr{B}$ instead of $(\alpha, H) : (AP, \mathscr{A}) \longrightarrow (AP', \mathscr{B})$, except in those cases where it could lead to confusion.

Composition of generalized stuttering simulations can be defined by $(\beta, G) \circ (\alpha, F) = (\overline{\beta} \circ \alpha, G \circ F)$. Using as objects pairs $(AP, \mathscr{M})$ with AP a set of atomic propositions and

$\mathscr{M}$ a Kripke structure over AP, this immediately gives rise to categories **KSSim** and **KSMap** of Kripke structures and stuttering simulations and simulation maps, respectively. However, generalized *strict* simulations between Kripke structures over different sets of atomic propositions *do not* compose, unless we only use functions of the form $\alpha : AP \longrightarrow \mathrm{Bool}(AP')$, where $\mathrm{Bool}(AP')$ is the set of Boolean formulas over AP'. (This situation will recur in Sections 5 and 6.)

The relationships between some of the different categories of Kripke structures introduced can be summarized in the following diagram, where the horizontal arrows are inclusions while the vertical ones are the expected forgetful functors.

$$\begin{array}{ccccccc} \mathbf{KMap}_{AP} & \longrightarrow & \mathbf{KSim}_{AP} & \longrightarrow & \mathbf{KSSim}_{AP} & \longrightarrow & \mathbf{KSSim} \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ \mathbf{TSys} & \longrightarrow & \mathbf{TSys} & \longrightarrow & \mathbf{STSys} & \longrightarrow & \mathbf{STSys} \end{array}$$

The important fact about stuttering simulations is that they reflect satisfaction of appropriate classes of formulas. Given Kripke structures $\mathscr{A}$ over AP and $\mathscr{B}$ over AP', we say that a stuttering simulation $(\alpha, H) : \mathscr{A} \longrightarrow \mathscr{B}$ *reflects* the satisfaction of a formula $\varphi \in \mathrm{CTL}^*(AP)$ if either:

- φ is a state formula, and $\mathscr{B}, b \models \overline{\alpha}(\varphi)$ and aHb imply that $\mathscr{A}, a \models \varphi$; or
- φ is a path formula, and $\mathscr{B}, \rho \models \overline{\alpha}(\varphi)$ and ρ H-matches π imply that $\mathscr{A}, \pi \models \varphi$.

Theorem 1 ([11]). *Stuttering simulations always reflect satisfaction of* $\mathrm{ACTL}^* \setminus \{\neg, \mathbf{X}\}$ *formulas. Strict stuttering simulations also reflect satisfaction of* $\mathrm{ACTL}^* \setminus \mathbf{X}$ *formulas.*

Appendix B contains a summary of the categories presented in this section. The "best" one is **KSSim**, the most general one, in that it provides the greater flexibility for relating arbitrary Kripke structures which otherwise could not be related; on the other hand, as we will see in Section 7, we know less about its categorical properties than for most of the others.

3 Some Categorical Concepts

Almost all the notions from category theory [9,2] that we use are rather basic and we only review those concepts that may not be so familiar. To try to avoid confusions with simulation morphisms, we refer to the morphisms in a category simply as arrows.

Opfibrations. What determines an *opfibration* [8] is the capacity of "lifting" an arrow in a base category to another category in an "initial" (and hence minimal) manner in an appropriate sense.

Let $F : \mathscr{C} \longrightarrow \mathscr{D}$ be a functor. An arrow $f : X \longrightarrow Y$ in $\mathscr{C}$ is *opcartesian* over u if $F(f) = u$ and for every arrow $g : X \longrightarrow Z$ in $\mathscr{C}$ such that $F(g) = v \circ u$ for some $v : F(Y) \longrightarrow F(Z)$ there exists a unique arrow $h : Y \longrightarrow Z$ such that $g = h \circ f$ and $F(h) = v$. The functor F is an *opfibration* if there exists an opcartesian morphism over every arrow $u : F(X) \longrightarrow J$. The dual notions are those of *cartesian morphism* and *fibration*.

Institutions. The notion of institution is due to Goguen and Burstall's seminal work [6]; their goal was to capture the notion of model in a formalism independent way. An *institution* is a 4-tuple $\mathscr{I} = (\mathbf{Sign}, sen, \mathbf{Mod}, \models)$ such that:

- **Sign** is a category whose objects are called *signatures*,
- $sen : \mathbf{Sign} \longrightarrow \mathbf{Set}$ is a functor that associates to each signature Σ a set of Σ-*sentences*,
- $\mathbf{Mod} : \mathbf{Sign}^{\mathrm{op}} \longrightarrow \mathbf{Cat}$ is a functor that associates to each signature Σ a category whose objects are called Σ-*models*, and
- $\models$ is a function that associates to each $\Sigma \in |\mathbf{Sign}|$ a binary relation $\models_\Sigma \subseteq |\mathbf{Mod}(\Sigma)| \times sen(\Sigma)$ called Σ-*satisfaction*, in such a way that the following property holds for every $H : \Sigma \longrightarrow \Sigma', M' \in |\mathbf{Mod}(\Sigma')|$, and every $\varphi \in sen(\Sigma)$: $M' \models_{\Sigma'} sen(H)(\varphi) \iff \mathbf{Mod}(H)(M') \models_\Sigma \varphi$.

A *theory morphism* $H : (\Sigma, \Gamma) \longrightarrow (\Sigma', \Gamma')$ is a signature morphism $H : \Sigma \longrightarrow \Sigma'$ such that every model in $\mathbf{Mod}(\Sigma')$ that satisfies Γ' also satisfies $sen(H)(\varphi)$, for all $\varphi \in \Gamma$. By defining $\mathbf{Mod}(\Sigma, \Gamma)$ as the full subcategory of $\mathbf{Mod}(\Sigma)$ determined by those models that satisfy Γ, we can extend **Mod** to a functor $\mathbf{Mod} : \mathbf{Th}^{\mathrm{op}} \longrightarrow \mathbf{Cat}$, where **Th** is the category of theories and theory morphisms.

A property expressing the possibility of "putting theories together" by colimits is the exactness of an institution. An institution is *exact* if its category of signatures is cocomplete and the model functor **Mod** preserves limits, and is *semiexact* if **Sign** has pushouts and **Mod** sends pushouts in **Sign** to pullbacks in **Cat**.

Monads and Kleisli categories. A *monad* (called a *triple* in [2]) is a tuple (T, η, μ), where $T : \mathscr{C} \longrightarrow \mathscr{C}$ is a functor, and $\eta : 1_{\mathscr{C}} \longrightarrow T$ and $\mu : T \circ T \longrightarrow T$ are natural transformations satisfying $\mu \circ \eta T = \mu \circ T\eta = 1_T$ and $\mu \circ \mu T = \mu \circ T\mu$.

All monads can be obtained from adjunctions. One possible construction makes use of the so-called *Kleisli category*. The Kleisli category $\mathscr{C}_T$ of a monad (T, η, μ) has as objects those of $\mathscr{C}$. If X and Y are objects of $\mathscr{C}$, an arrow $X \longrightarrow Y$ in the Kleisli category is an arrow $X \longrightarrow T(Y)$ in $\mathscr{C}$. Composition of two arrows $f : X \longrightarrow T(Y)$ and $g : Y \longrightarrow T(Z)$ is defined as $\mu_Z \circ Tg \circ f$.

Grothendieck construction. Often we are interested in considering all the components of an *indexed category* together in a "flat" category obtained by taking the disjoint union of the components and adding some new arrows. This is called, for example in [15], the *Grothendieck construction*. Given a functor $\mathbf{C} : \mathscr{I}^{\mathrm{op}} \longrightarrow \mathbf{Cat}$, the associated *Grothendieck construction* is defined by:

- its objects are pairs (I, X), where I is an object of $\mathscr{I}$ and X is an object of $\mathbf{C}(I)$;
- an arrow $(I, X) \longrightarrow (J, Y)$ is a pair (u, f) with $u : I \longrightarrow J$ in $\mathscr{I}$ and $f : X \longrightarrow \mathbf{C}(u)(Y)$ in $\mathbf{C}(I)$;
- the composition of arrows $(u, f) : (I, X) \longrightarrow (J, Y)$ and $(v, g) : (J, Y) \longrightarrow (K, Z)$ is given by $(v, g) \circ (u, f) = (v \circ u, \mathbf{C}(u)(g) \circ f)$.

Regular epis and monos. As defined in [7], an arrow $m : X \longrightarrow Y$ is a *regular monomorphism* if there exist arrows f and g such that m is the equalizer of f and g. Dually, $e : X \longrightarrow Y$ is a *regular epimorphism* if it is the coequalizer of two arrows.

Given two classes $\mathscr{E}$ and $\mathscr{M}$ of epimorphisms and monomorphisms respectively, closed under composition with isomorphisms, a $(\mathscr{E}, \mathscr{M})$-*factorization* of an arrow f is a factorization $f = m \circ e$ with e in $\mathscr{E}$ and m in $\mathscr{M}$. A category is (univocally) $(\mathscr{E}, \mathscr{M})$-*factorizable* if every arrow has a (unique up to isomorphism) $(\mathscr{E}, \mathscr{M})$-factorization.

A category is a $(\mathscr{E},\mathscr{M})$-*category* if it is univocally factorizable and both $\mathscr{E}$ and $\mathscr{M}$ are closed under composition.

4 Minimal Kripke Structures

Theorem 1 is the basis of the method of model checking by abstraction: given an infinite (or too large) system $\mathscr{M}$, one tries to find a system $\mathscr{A}$ with a finite set of reachable states that simulates it and uses a model checker to prove properties of $\mathscr{M}$ by means of $\mathscr{A}$. But usually, one only has the concrete system $\mathscr{M}$ and a *surjective* function $h : M \longrightarrow A$ mapping concrete states to a simplified abstract domain A. In this situation, we are interested in using h to find a Kripke structure that best simulates $\mathscr{M}$ under certain conditions. In [4] the *minimal transition system* associated to a transition system $\mathscr{M}$ and a surjective function $h : M \to A$ was defined; using our notion of simulation this can be extended to the level of Kripke structures.

Definition 1. *The* minimal *Kripke structure* $\mathscr{M}^h_{\min}$ *corresponding to a Kripke structure* $\mathscr{M}$ *and the surjective function* $h : M \longrightarrow A$ *is given by the triple* $(A,(h\times h)(\to_{\mathscr{M}}), L_{\mathscr{M}^h_{\min}})$, *where* $L_{\mathscr{M}^h_{\min}}(a) = \bigcap_{x\in h^{-1}(a)} L_{\mathscr{M}}(x)$.

The following proposition is an immediate consequence of the definitions.

Proposition 2 ([13]). *For any Kripke structure* $\mathscr{M}$ *and any surjective function h,* $h : \mathscr{M} \longrightarrow \mathscr{M}^h_{\min}$ *is an AP-map.*

The use of the adjective "minimal" is appropriate since, as pointed out in [4], $\mathscr{M}^h_{\min}$ is the most accurate approximation to $\mathscr{M}$ that is consistent with h. Within our framework, the notion of minimality can be expressed in a precise categorical sense by means of an opcartesian morphism.

Proposition 3. *For a Kripke structure* $\mathscr{M}$ *and surjective function* $h : M \longrightarrow A$, *the AP-map* $h : \mathscr{M} \longrightarrow \mathscr{M}^h_{\min}$ *is an opcartesian morphism in the context of the forgetful functor* $U : \mathbf{KMap}_{AP} \longrightarrow \mathbf{Set}$ *mapping a Kripke structure* $\mathscr{M} = (M, \to_{\mathscr{M}}, L_{\mathscr{M}})$ *to its underlying set M and an AP-map to itself.*

Proof. Given $f : \mathscr{M} \longrightarrow \mathscr{N}$ in $\mathbf{KMap}_{AP}$ such that it can be factorized in **Set** as $f = g \circ h$ for some function $g : A \longrightarrow N$, we have to find a unique g' in $\mathbf{KMap}_{AP}$ such that $g' : \mathscr{M}^h_{\min} \longrightarrow \mathscr{N}$, $f = g' \circ h$, and $U(g') = g$. By definition of U, it must be $g' = g$; we have to check that g is actually an AP-map.

By definition of $\mathscr{M}^h_{\min}$, if $a \to_{\mathscr{M}^h_{\min}} b$ there exist x and y in M such that $h(x) = a$, $h(y) = b$, and $x \to_{\mathscr{M}} y$. Hence, since f is an AP-map, $g(a) = g(h(x)) = f(x) \to_{\mathscr{N}} f(y) = g(h(y)) = g(b)$. In addition, using again the fact that f is an AP-map, if $p \in L_{\mathscr{N}}(s)$ then $p \in L_{\mathscr{M}}(x)$ for all x in M such that $f(x) = s$. Let then $a \in A$ such that $g(a) = s$: for all y in M such that $h(y) = a$, since $f(y) = g(h(y)) = s$, it is the case that $p \in L_{\mathscr{M}}(y)$. Therefore, $p \in L_{\mathscr{M}^h_{\min}}(a)$, and for all a with $g(a) = s$ we have $L_{\mathscr{N}}(s) \subseteq L_{\mathscr{M}^h_{\min}}(a)$. □

Note that this result can be extended to the category $\mathbf{KSMap}_{AP}$: then, whenever f is a stuttering AP-map, so will be g. The result also holds for generalized simulations in which the set of atomic propositions may vary.

Proposition 4. *The simulation* $(\eta_{AP}, h) : \mathscr{M} \longrightarrow \mathscr{M}^h_{\min}$*, where* $h : M \longrightarrow A$ *is a surjective function and* η_{AP} *is the inclusion* $AP \hookrightarrow \text{State}\backslash\neg(AP)$*, is an opcartesian morphism for the forgetful functor* $U :$ **KMap** $\longrightarrow$ **Set** *mapping a pair* $(AP, \mathscr{M})$ *to the underlying set* M *and a simulation map* (α, h) *to the corresponding function* h*.*

Proof. The proof follows the same steps as the one for Proposition 3, despite the fact that the set of atomic propositions now may vary from one Kripke structure to another. □

5 Borrowing

Simulations, in all their different variants, require suitable preservation of transitions and of atomic propositions. Sometimes, however, it is more natural and easier to think just in terms of the underlying transition systems; in those cases we can still recover a full-fledged simulation by *borrowing* the Kripke structure of the domain using the labeling function of the codomain.

Definition 2. *Let* $\mathscr{A} = (A, \rightarrow_{\mathscr{A}})$ *be a transition system and let* $\mathscr{B} = (B, \rightarrow_{\mathscr{B}}, L_{\mathscr{B}})$ *be a Kripke structure over a set AP of atomic propositions. If* $h : A \longrightarrow B$ *is a stuttering map between the underlying transition systems, then* $\mathscr{A}$ *can be extended to a Kripke structure over AP with* $L_{\mathscr{A}} = L_{\mathscr{B}} \circ h$*. We say that* $\mathscr{A}$ borrows *its properties from* $\mathscr{B}$*.*

Proposition 5. *If* $\mathscr{A} = (A, \rightarrow_{\mathscr{A}})$ *borrows its properties from a Kripke structure* $\mathscr{B} = (B, \rightarrow_{\mathscr{B}}, L_{\mathscr{B}})$ *over a set AP of atomic propositions through a stuttering map of transition systems* $h : (A, \rightarrow_{\mathscr{A}}) \longrightarrow (B, \rightarrow_{\mathscr{B}})$*, then* h *becomes a strict stuttering AP-map. Furthermore,* h *is a cartesian morphism for the forgetful functor* $U :$ **KSMap**$_{AP} \longrightarrow$ **STSys** *mapping a Kripke structure to its underlying transition system.*

Proof. h is clearly a strict stuttering AP-map because, by definition of $L_{\mathscr{A}}$, atomic propositions are preserved. To show that it is a cartesian morphism, assume a stuttering AP-map $f : \mathscr{C} \longrightarrow \mathscr{B}$ and a stuttering map of transition systems $g : U(\mathscr{C}) \longrightarrow (A, \rightarrow_{\mathscr{A}})$ such that $f = h \circ g$: we have to show that there is a unique stuttering AP-map g' such that $h \circ g' = f$ and $U(g') = g$. The only possible candidate is g, and we have to check that $g : \mathscr{C} \longrightarrow \mathscr{A}$ is indeed a stuttering AP-map. By hypothesis, g is a map of transition systems. Now, assume that $g(c) = a$ and $p \in L_{\mathscr{A}}(a)$. It follows that $p \in L_{\mathscr{B}}(h(a))$, and since $f(c) = (h \circ g)(c) = h(a)$ and f is a stuttering AP-map, $p \in L_{\mathscr{C}}(c)$ as required. □

It is interesting to note that this proposition also holds even if h is not a function (but the resulting AP-simulation may not be strict).

One could ask whether this result can be extended to the Grothendieck category **KSMap** so that (η_{AP}, h) becomes a cartesian morphism for the forgetful functor $U :$ **KSMap** $\longrightarrow$ **STSys**. The answer is *no* and the reason lies in the generality of the functions $\alpha : AP \longrightarrow \text{State}(AP')$ used to relate Kripke structures over different sets of atomic propositions. However, the result can be recovered by working in the subcategory **KSMap**^{bool} of **KSMap** in which the codomain of the functions α is restricted to $\text{Bool}(AP')$. That is the content of the following proposition.

Proposition 6. *If $\mathscr{A} = (A, \rightarrow_{\mathscr{A}})$ borrows its properties from $\mathscr{B} = (B, \rightarrow_{\mathscr{B}}, L_{\mathscr{B}})$ through a stuttering map of transition systems $h : (A, \rightarrow_{\mathscr{A}}) \longrightarrow (B, \rightarrow_{\mathscr{B}})$, then (η_{AP}, h) becomes a strict stuttering map. Furthermore, (η_{AP}, h) is a cartesian morphism for the forgetful functor U : **KSMap**$^{bool} \longrightarrow$ **STSys** mapping a Kripke structure to its underlying transition system.*

Proof. (η_{AP}, h) is clearly a strict stuttering map because, by definition of $L_{\mathscr{A}}$, atomic propositions are preserved. To show that it is a cartesian morphism, assume a stuttering map $(\alpha, f) : \mathscr{C} \longrightarrow \mathscr{B}$ and a stuttering map of transition systems $g : U(\mathscr{C}) \longrightarrow (A, \rightarrow_{\mathscr{A}})$ such that $f = h \circ g$. We have to show that there is a unique stuttering map (α', g') such that $(\eta_{AP}, h) \circ (\alpha', g') = (\alpha, f)$ and $U(\alpha', g') = g$. The only possible candidate is (α, g), and therefore we have to check that $g : \mathscr{C} \longrightarrow \mathscr{A}|_{\alpha}$ is indeed a stuttering AP'-map, where AP' is the set of atomic propositions of $\mathscr{C}$. By hypothesis, g is a map of transition systems. Now, assume that $g(c) = a$ and $p \in L_{\mathscr{A}|_{\alpha}}(a)$; it follows that $\mathscr{A}, a \models \alpha(p)$. Since $L_{\mathscr{A}}(a) = L_{\mathscr{B}}(h(a))$, it is immediate to show that for all $\varphi \in \mathrm{Bool}(AP)$, $\mathscr{A}, a \models \varphi$ iff $\mathscr{B}, h(a) \models \varphi$. Therefore, $\mathscr{B}, h(a) \models \alpha(p)$, and since $f(c) = (h \circ g)(c) = h(a)$ and (α, f) is a stuttering map, $\mathscr{C}, c \models p$ by Theorem 1, that is, $p \in L_{\mathscr{C}}(c)$ as required. □

6 Temporal Logic Institutions

It is not hard to notice that the result in Proposition 1 has a distinct institutional flavor. Indeed, Kripke structures can be organized as the models of a temporal logic institution [6] in which Proposition 1 corresponds to the property required of the satisfaction relation. Other institutions for temporal logics are discussed in [1], but their notions of signature morphism and of simulation are more restricted. Some of the ideas in this section were presented in [11]: we also include them here for the sake of completeness.

Let us first define the category of signatures. For that, let $\mathrm{State}\backslash\neg$: **Set** $\longrightarrow$ **Set** be the functor mapping a set AP to the set of state formulas $\mathrm{State}\backslash\neg(AP)$, and a function $\alpha : AP \longrightarrow AP'$ to its homomorphic extension $\overline{\alpha} : \mathrm{State}\backslash\neg(AP) \longrightarrow \mathrm{State}\backslash\neg(AP')$. Then, the triple $\langle \mathrm{State}\backslash\neg, \eta, \mu \rangle$ is a monad (Section 3), where $\eta : Id_{\mathbf{Set}} \Rightarrow \mathrm{State}\backslash\neg$ and $\mu : \mathrm{State}\backslash\neg \circ \mathrm{State}\backslash\neg \Rightarrow \mathrm{State}\backslash\neg$ are natural transformations such that $\eta_{AP}(p) = p$ and μ "unnests" a formula into its basic atomic propositions. Our category of signatures will be $\mathbf{Set}_{\mathrm{State}\backslash\neg}$, the Kleisli category of the monad; its objects are just sets, and the morphisms $AP \longrightarrow AP'$ are functions $\alpha : AP \longrightarrow \mathrm{State}\backslash\neg(AP')$.

Definition 3. *The institution of Kripke structures is given by:*

- $\mathbf{Sign}_{\mathbf{K}} = \mathbf{Set}_{\mathrm{State}\backslash\neg}$.
- $sen_{\mathbf{K}}$ *is the functor mapping a set AP to* $\mathrm{State}\backslash\neg(AP)$*, and a function $\alpha : AP \longrightarrow \mathrm{State}\backslash\neg(AP')$ to its homomorphic extension $\overline{\alpha} : \mathrm{State}\backslash\neg(AP) \longrightarrow \mathrm{State}\backslash\neg(AP')$.*
- $\mathbf{Mod}_{\mathbf{K}} : \mathbf{Set}_{\mathrm{State}\backslash\neg} \longrightarrow \mathbf{Cat}^{\mathrm{op}}$ *is given by* $\mathbf{Mod}_{\mathbf{K}}(AP) = \mathbf{KSim}_{AP}$ *and, for $\alpha : AP \longrightarrow AP'$ in* $\mathbf{Set}_{\mathrm{State}\backslash\neg}$, $\mathbf{Mod}_{\mathbf{K}}(\alpha)(\mathscr{A}) = \mathscr{A}|_{\alpha}$ *and* $\mathbf{Mod}_{\mathbf{K}}(\alpha)(H) = H$.
- *The satisfaction relation is defined as $\mathscr{A} \models \varphi$ iff $\mathscr{A}, a \models \varphi$ for all $a \in A$.*

Proposition 7. $\mathscr{I}_{\mathbf{K}} = (\mathbf{Sign}_{\mathbf{K}}, sen_{\mathbf{K}}, \mathbf{Mod}_{\mathbf{K}}, \models)$ *is an institution.*

Analogously, we could think of defining an institution for Kripke structures and strict morphisms. However, the fact that α can map an atomic proposition to an arbitrary formula makes it impossible. The problem is that the putative model functor is not such: the reduct of a strict simulation may not itself be strict. As happened in Sections 2.1 and 5, to solve this problem and get an institution for strict simulations it is enough to restrict the signature morphisms to be functions of the form $\alpha : AP \longrightarrow \mathrm{Bool}(AP)$.

Notice also that the category **KSim** can be obtained by means of the Grothendieck construction [15]. Indeed, **KSim** is just the Grothendieck category corresponding to the indexed category $\mathbf{Mod_K}$. (The same would happen for strict simulations if we were to work with the restricted α functions.) Similarly, **KMap** and **KBSim** can be obtained by modifying $\mathbf{Mod_K}$ so that AP is mapped to $\mathbf{KMap}_{AP}$ and $\mathbf{KBSim}_{AP}$, respectively.

Of course, analogous results exist for the general case of stuttering simulations. Now the functor used to define the Kleisli category of signatures is $\mathrm{State}\backslash\{\neg,\mathbf{X}\}$, mapping AP to the set of state formulas $\mathrm{State}\backslash\{\neg,\mathbf{X}\}(AP)$ (and to $\mathrm{Bool}(AP)$ for the strict case). Similarly, the model functor maps the set of atomic propositions AP to the corresponding category of sttutering AP-simulations, $\mathbf{KSSim}_{AP}$. Actually, as the proof reveals, the construction also applies to any temporal logic whose formulae are reflected by simulations; in particular, we could restrict the institutions to the LTL sublogic of ACTL*.

The institutions just introduced use the most general notion of signature morphism compatible with the reflection of suitable temporal formulas. But precisely because of this generality, they do not have the *exactness* property. To see this, it is enough to consider the set of atomic propositions $AP = \{p,q\}$ and signature morphisms $\alpha_1, \alpha_2 : AP \longrightarrow \mathrm{State}\backslash\neg(AP)$ such that $\alpha_1(p) = p \wedge q$ and $\alpha_2(p) = p \vee q$. Then, for any signature morphisms $\beta_1, \beta_2 : AP \longrightarrow \mathrm{State}\backslash\neg(AP')$, $(\beta_1 \circ \alpha_1)(p) = \beta_1(p) \wedge \beta_1(q)$, which is different from $(\beta_2 \circ \alpha_2)(p) = \beta_2(p) \vee \beta_2(q)$. This shows that $\mathbf{Sign_K}$ does not have pushouts. The situation, however, changes when the signature morphisms are restricted to mapping atomic propositions to atomic propositions. Note that the counterexample shows also that this time it is not enough to consider $\mathrm{Bool}(AP)$: we have to map *atomic* propositions to *atomic* propositions.

Proposition 8. *Let $\mathscr{I}'_{\mathbf{K}}$ be obtained from the institution $\mathscr{I}_{\mathbf{K}}$ by replacing $\mathbf{Set}_{\mathrm{State}\backslash\neg}$ by* **Set** *as the category of signatures. Then $\mathscr{I}'_{\mathbf{K}}$ is a semiexact institution.*

The same result also applies to the institutions of strict and stuttering simulations described above.

7 Limits and Colimits in Categories of Simulations

We collect in this section categorical properties about existence of limits and colimits in some of the categories of Kripke structures that have been presented in Section 2. We focus on the categories over a fixed set of atomic propositions. For the Grothendieck categories, colimits can be obtained by mimicking the constructions presented below; however, we conjecture that in such Grothendieck categories limits do not exist in general.

7.1 Products and Coproducts

Proposition 9. *For all sets of atomic propositions AP, the category $\mathbf{KMap}_{AP}$ has finite products.*

Proof. Given Kripke structures $\mathscr{A}$ and $\mathscr{B}$, define $\mathscr{A} \times \mathscr{B} = (A \times B, \rightarrow_{\mathscr{A}\times\mathscr{B}}, L_{\mathscr{A}\times\mathscr{B}})$, where $(a,b) \rightarrow_{\mathscr{A}\times\mathscr{B}} (a',b')$ iff $a \rightarrow_{\mathscr{A}} a'$ and $b \rightarrow_{\mathscr{B}} b'$, and $L_{\mathscr{A}\times\mathscr{B}}(a,b) = L_{\mathscr{A}}(a) \cup L_{\mathscr{B}}(b)$, with the usual projections $\pi_{\mathscr{A}} : \mathscr{A} \times \mathscr{B} \longrightarrow \mathscr{A}$ and $\pi_{\mathscr{B}} : \mathscr{A} \times \mathscr{B} \longrightarrow \mathscr{B}$. The relation $\rightarrow_{\mathscr{A}\times\mathscr{B}}$ is total and thus $\mathscr{A} \times \mathscr{B}$ is well-defined, and it is immediate to check that $\pi_{\mathscr{A}}$ and $\pi_{\mathscr{B}}$ are *AP*-maps.

Now, if $f : \mathscr{C} \longrightarrow \mathscr{A}$ and $g : \mathscr{C} \longrightarrow \mathscr{B}$ are *AP*-maps, the unique arrow $\langle f,g\rangle : \mathscr{C} \longrightarrow \mathscr{A} \times \mathscr{B}$ such that $\pi_{\mathscr{A}} \circ \langle f,g\rangle = f$ and $\pi_{\mathscr{B}} \circ \langle f,g\rangle = g$ is given by $\langle f,g\rangle(c) = (f(c),g(c))$. Uniqueness is clear: we have to check that $\langle f,g\rangle$ is indeed an *AP*-map. If $c \rightarrow_{\mathscr{C}} c'$ then $f(c) \rightarrow_{\mathscr{A}} f(c')$ and $g(c) \rightarrow_{\mathscr{B}} g(c')$, and therefore $\langle f,g\rangle(c) \rightarrow_{\mathscr{A}\times\mathscr{B}} \langle f,g\rangle(c')$. And if $p \in L_{\mathscr{A}\times\mathscr{B}}(\langle f,g\rangle(c))$ then $p \in L_{\mathscr{A}}(f(c))$ or $p \in L_{\mathscr{B}}(g(c))$: either way, $p \in L_{\mathscr{C}}(c)$. □

Note that this construction can be extended to infinite products in the expected way and, since the Kripke structure with a single state *, single transition $* \rightarrow *$, and $L(*) = \emptyset$ is a final object, $\mathbf{KMap}_{AP}$ has arbitrary products.

This result is also true for the category of strict *AP*-maps, but the constructions are slightly more involved. The final object in $\mathbf{KMap}^{\mathrm{str}}_{AP}$ is $(\mathscr{P}(AP), \mathscr{P}(AP) \times \mathscr{P}(AP), id_{\mathscr{P}(AP)})$. The construction of finite products is shown in the proof of the next result; this is sometimes called the *synchronous product* of Kripke structures in the literature.

Proposition 10. *For all sets of atomic propositions AP, the category* $\mathbf{KMap}^{\mathrm{str}}_{AP}$ *has finite products.*

Proof. Given Kripke structures $\mathscr{A}$ and $\mathscr{B}$, let $\mathscr{A} \times \mathscr{B}$ be the Kripke structure defined in the proof of Proposition 9. Let us define

$$\mathrm{Path}(\mathscr{A} \times \mathscr{B})^{=} = \{\pi \in \mathrm{Path}(\mathscr{A} \times \mathscr{B}) \mid \text{for all } i,\ L_{\mathscr{A}}(\pi_{\mathscr{A}}(\pi(i))) = L_{\mathscr{B}}(\pi_{\mathscr{B}}(\pi(i)))\},$$
$$D = \{(a,b) \mid \text{there exists } \pi \in \mathrm{Path}(\mathscr{A} \times \mathscr{B})^{=} \text{ and } i \in \mathbb{N} \text{ such that } (a,b) = \pi(i)\}.$$

The product of $\mathscr{A}$ and $\mathscr{B}$ in $\mathbf{KMap}^{\mathrm{str}}_{AP}$ is given by $\mathscr{A} \times^{\mathrm{st}} \mathscr{B} = (D, \rightarrow_{\mathscr{A}\times\mathscr{B}}|_{D^2}, L_{\mathscr{A}\times\mathscr{B}}|_D)$ with the expected projections. By construction, $\rightarrow_{\mathscr{A}\times\mathscr{B}}|_{D^2}$ is total. Note that for arbitrary strict *AP*-maps $f : \mathscr{C} \longrightarrow \mathscr{A}$ and $g : \mathscr{C} \longrightarrow \mathscr{B}$, the function $\langle f,g\rangle : \mathscr{C} \longrightarrow \mathscr{A} \times^{\mathrm{st}} \mathscr{B}$ is well-defined. For each $c \in C$, let π be a path with $\pi(0) = c$ (it must exist because $\rightarrow_{\mathscr{C}}$ is total). Since both f and g are strict, $L_{\mathscr{A}}(f(\pi(i))) = L_{\mathscr{B}}(g(\pi(i)))$ and the path

$$(f(\pi(0)),g(\pi(0))) \rightarrow_{\mathscr{A}\times\mathscr{B}} (f(\pi(1)),g(\pi(1))) \rightarrow_{\mathscr{A}\times\mathscr{B}} \cdots$$

belongs to $\mathrm{Path}(\mathscr{A} \times \mathscr{B})^{=}$; thus, $(f(c),g(c)) \in D$. And $\langle f,g\rangle$ is clearly strict. □

Note that in some cases we may have $\mathscr{A} \times^{\mathrm{str}} \mathscr{B} = \emptyset$ even though neither $\mathscr{A}$ nor $\mathscr{B}$ are empty. This simply means that the *only* Kripke structure $\mathscr{C}$ from which there exist strict *AP*-simulations to both $\mathscr{A}$ and $\mathscr{B}$ is the empty one. Note also that the construction can be extended in the expected way to infinite products.

If we take a look at what happens when considering *AP*-simulations instead of just maps, it turns out that finite producs also exist in $\mathbf{KSim}_{AP}$ although its definition is quite different from the previous ones.

Proposition 11. *For all sets of atomic propositions AP, the category* $\mathbf{KSim}_{AP}$ *has finite products.*

Proof. Define the product of $\mathscr{A}$ and $\mathscr{B}$ to be $\mathscr{A} \times \mathscr{B} = (A \uplus B, \rightarrow_{\mathscr{A}} \uplus \rightarrow_{\mathscr{B}}, L_{\mathscr{A}\times\mathscr{B}})$, where $L_{\mathscr{A}\times\mathscr{B}}(x)$ is $L_{\mathscr{A}}(x)$ if $x \in A$ or $L_{\mathscr{B}}(x)$ if $x \in B$, with projections $\Pi_{\mathscr{A}}$ and $\Pi_{\mathscr{B}}$ defined by $a\Pi_{\mathscr{A}}a$ for all $a \in A$ and $b\Pi_{\mathscr{B}}b$ for all $b \in B$. Then, for AP-simulations $F : \mathscr{C} \longrightarrow \mathscr{A}$ and $G : \mathscr{C} \longrightarrow \mathscr{B}$, the unique $\langle F, G\rangle$ is defined by $c\langle F, G\rangle a$ iff cFa, and $c\langle F, G\rangle b$ iff cGb. □

Again, the above construction can be extended to arbitrary families $\{\mathscr{A}_i\}_{i\in I}$ of Kripke structures, and since the empty Kripke structure is trivially a final object, the category $\mathbf{KSim}_{AP}$ has arbitrary products.

By contrast, there are no products in the category $\mathbf{KSMap}_{AP}$ of stuttering AP-simulations. To see this, consider $\mathscr{A}$ given by $a_1 \rightarrow_{\mathscr{A}} a_2 \rightarrow_{\mathscr{A}} \ldots$ and $\mathscr{B}$ by $b_1 \rightarrow_{\mathscr{B}} b_2 \rightarrow_{\mathscr{B}} \ldots$, where both labeling functions are empty. Now, assume $\mathscr{C}$ is given by $c_1 \rightarrow_{\mathscr{C}} c_2 \rightarrow_{\mathscr{C}} \ldots$. Consider now stuttering AP-simulations $f : \mathscr{C} \longrightarrow \mathscr{A}$ with $f(c_1) = a_1$, $f(c_{2*i}) = a_{i+1}$, and $f(c_{2*i+1}) = a_{i+1}$ for $i \geq 1$, and $g : \mathscr{C} \longrightarrow \mathscr{B}$ with $g(c_{2*i+1}) = a_{i+1}$ and $g(c_{2*i+2}) = a_{i+1}$ for $i \geq 0$. Assume that $\mathscr{D}$ is the product of $\mathscr{A}$ and $\mathscr{B}$ with projections $\pi_{\mathscr{A}}$ and $\pi_{\mathscr{B}}$, and let $d_1 \rightarrow_{\mathscr{D}} d_2 \rightarrow_{\mathscr{D}} \ldots$ be the path in $\mathscr{D}$ $\langle f, g\rangle$-matching the path in $\mathscr{C}$ that starts at c_1. We have $\langle f, g\rangle(c_1) = d_1$, $\pi_{\mathscr{A}}(d_1) = a_1$, and $\pi_{\mathscr{B}}(d_1) = b_1$. Now, $\langle f, g\rangle(c_2)$ must be equal to d_1 or to d_2. But the first alternative cannot hold because then $\pi_{\mathscr{A}}(\langle f, g\rangle(c_2)) \neq f(c_2)$; therefore $\langle f, g\rangle(c_2) = d_2$, and $\pi_{\mathscr{A}}(d_2)$ and $\pi_{\mathscr{B}}(d_2)$ have to be a_2 and b_1, respectively. And we are done, because if we swap the definitions of f and g the same argument leads to $\pi_{\mathscr{A}}(d_1) = \pi_{\mathscr{B}}(d_2) = a_1$: a contradiction.

Coproducts exist in all the categories mentioned in the previous section and their definition is the same in all cases. Here we present the details for $\mathbf{KSim}_{AP}$.

Proposition 12. *For all sets of atomic propositions AP, the category* $\mathbf{KSim}_{AP}$ *has finite coproducts.*

Proof. Given Kripke structures $\mathscr{A}$ and $\mathscr{B}$, define $\mathscr{A} + \mathscr{B}$ as $(A \uplus B, \rightarrow_{\mathscr{A}} \uplus \rightarrow_{\mathscr{B}}, L_{\mathscr{A}+\mathscr{B}})$, where $L_{\mathscr{A}+\mathscr{B}}(x)$ is $L_{\mathscr{A}}(x)$ if $x \in A$ or $L_{\mathscr{B}}(x)$ if $x \in B$, and with inclusions $I_{\mathscr{A}}$ and $I_{\mathscr{B}}$ defined by $aI_{\mathscr{A}}a$ for all $a \in A$ and $bI_{\mathscr{B}}b$ for all $b \in B$. $\mathscr{A} + \mathscr{B}$ is clearly well-defined and it is trivial to check that $I_{\mathscr{A}}$ and $I_{\mathscr{B}}$ are AP-simulations. Now, for $F : \mathscr{A} \longrightarrow \mathscr{C}$ and $G : \mathscr{B} \longrightarrow \mathscr{C}$ arbitrary AP-simulations, define $[F, G] : \mathscr{A} + \mathscr{B}$ by $a[F, G]c$ iff aFc, and $b[F, G]c$ iff bGc. It is easy to check that $[F, G]$ so defined is the only AP-simulation that satisfies $I_{\mathscr{A}} \circ [F, G]$ and $I_{\mathscr{B}} \circ [F, G]$. □

Note that the Kripke structure $\mathscr{A} + \mathscr{B}$ is the same as the Kripke structure $\mathscr{A} \times \mathscr{B}$ of Proposition 11, and that the construction also applies to infinite families. The initial object corresponds to the empty Kripke structure.

7.2 Equalizers and Coequalizers

Proposition 13. *For all sets of atomic propositions AP, the category* $\mathbf{KMap}_{AP}$ *has equalizers.*

Proof. Let $f, g : \mathscr{A} \longrightarrow \mathscr{B}$ be AP-maps. Let us define

$$\mathrm{Path}(\mathscr{A})_{f,g} = \{\pi \in \mathrm{Path}(\mathscr{A}) \mid f \circ \pi = g \circ \pi\},$$
$$E = \{a \in A \mid \text{there exists } \pi \in \mathrm{Path}(\mathscr{A})_{f,g} \text{ and } i \in \mathbb{N} \text{ such that } a = \pi(i)\}.$$

The equalizer of f and g is given by the Kripke structure $\mathscr{E} = (E, \rightarrow_{\mathscr{A}}|_{E^2}, L_{\mathscr{A}}|_E)$ and the inclusion $e : \mathscr{E} \longrightarrow \mathscr{A}$. By definition, $\rightarrow_{\mathscr{E}}$ is total and thus $\mathscr{E}$ is a well-defined Kripke structure; e is trivially a (strict) AP-map. Now, suppose that $h : \mathscr{D} \longrightarrow \mathscr{A}$ is an AP-map such that $f \circ h = g \circ h$. Define $m : \mathscr{D} \longrightarrow \mathscr{E}$ by $m(d) = h(d)$. Obviously $f(h(d)) = g(h(d))$ and, since $\rightarrow_{\mathscr{D}}$ is total, there is a path π in $\mathscr{D}$ such that $\pi(0) = d$: its image by h belongs to $\mathrm{Path}(\mathscr{A})_{f,g}$ and therefore $h(d) \in E$ and m is well-defined. It is clear that m is unique and that $h = e \circ m$. Finally, m is an AP-map: if $d \rightarrow_{\mathscr{D}} d'$ then $h(d) \rightarrow_{\mathscr{A}} h(d')$ and by definition of m and $\rightarrow_{\mathscr{E}}$ it is $m(d) \rightarrow_{\mathscr{E}} m(d')$; and if $p \in L_{\mathscr{E}}(m(d))$ then $p \in L_{\mathscr{A}}(h(d))$ and hence $p \in L_{\mathscr{D}}(d)$. □

It is easy to check that the same construction gives equalizers in the categories $\mathbf{KMap}^{\mathrm{str}}_{AP}$ and $\mathbf{KSMap}_{AP}$. As for $\mathbf{KSim}_{AP}$, we have not been able to prove or disprove the existence of equalizers.

Proposition 14. *For all sets of atomic propositions AP, the category* $\mathbf{KMap}_{AP}$ *has coequalizers.*

Proof. Assume that $f, g : \mathscr{A} \longrightarrow \mathscr{B}$ are AP-simulations, and define $\equiv$ to be the least equivalence relation over B containing $\{(f(a), g(a)) \mid a \in A\}$. Then the coequalizer of f and g is given by the quotient Kripke structure $\mathscr{B}/\!\equiv$ and the projection $c : \mathscr{B} \longrightarrow \mathscr{B}/\!\equiv$. For assume that $h : \mathscr{B} \longrightarrow \mathscr{D}$ is an AP-map such that $h \circ f = h \circ g$; we can define $m : \mathscr{B}/\!\equiv \longrightarrow \mathscr{D}$ by $m([b]) = h(b)$ with $h = m \circ c$. We have to check that m is well-defined and that it is an AP-map. The first part is proved by showing that if $b_1 \equiv b_2$ then $h(b_1) = h(b_2)$, by induction on the definition of $\equiv$. The base case corresponds to $f(a) \equiv g(a)$, and by hypothesis it is $h(f(a)) = h(g(a))$. And it is immediate that the result also holds for $b \equiv b$, for $b_2 \equiv b_1$ if it holds for $b_1 \equiv b_2$, and for $b_1 \equiv b_3$ if it holds for $b_1 \equiv b_2$ and for $b_2 \equiv b_3$. For the second part, if $[b_1] \rightarrow_{\mathscr{B}/\equiv} [b_2]$ it must be $b'_1 \rightarrow_{\mathscr{B}} b'_2$ for some $b'_1 \equiv b_1$ and $b'_2 \equiv b_2$ and hence $h(b'_1) \rightarrow_{\mathscr{D}} h(b'_2)$. And if $p \in L_{\mathscr{D}}(m([b]))$ then $p \in L_{\mathscr{B}}(b')$ for all $b' \in [b]$ and therefore $p \in L_{\mathscr{B}/\equiv}([b])$. □

Again, this construction also applies to the category $\mathbf{KMap}^{\mathrm{str}}_{AP}$ but we do not know what happens in $\mathbf{KSim}_{AP}$ or $\mathbf{KSMap}_{AP}$.

7.3 Satisfaction for Products and Coproducts

At this point, it is interesting to ask ourselves whether there is any relation between the formulas satisfied by two Kripke structures $\mathscr{A}$ and $\mathscr{B}$ and those satisfied by their product $\mathscr{A} \times \mathscr{B}$, or more generally, whether there is any relation between the properties satisfied by a family of Kripke structures and those of their corresponding limits and colimits. Unfortunately, there is no general pattern.

Let us consider products. In one direction, the relation is immediate: there exist simulations from $\mathscr{A} \times \mathscr{B}$ to both $\mathscr{A}$ and $\mathscr{B}$ (the projections) and therefore any property that holds in any of the latter will also be true of $\mathscr{A} \times \mathscr{B}$. In the other direction, since products and coproducts coincide in $\mathbf{KSim}_{AP}$ there are also simulations from $\mathscr{A}$ and $\mathscr{B}$ to $\mathscr{A} \times \mathscr{B}$ (the inclusions) and thus properties of $\mathscr{A} \times \mathscr{B}$ can be transferred to both $\mathscr{A}$ and $\mathscr{B}$. This relation however does not hold when simulations are restricted to be maps. For example, in the category $\mathbf{KMap}_{AP}$ for $AP = \{p, q\}$, if $\mathscr{A} = (\{a\}, a \rightarrow_{\mathscr{A}}$

$a, L_{\mathscr{A}})$ with $L_{\mathscr{A}}(a) = \{p\}$, and $\mathscr{B} = (\{b\}, b \rightarrow_{\mathscr{B}} b, L_{\mathscr{B}})$ with $L_{\mathscr{B}}(b) = \{q\}$, we have $\mathscr{A} \times \mathscr{B}, (a,b) \models \mathbf{G}(p \wedge q)$ but $\mathscr{A}, a \not\models \mathbf{G}(p \wedge q)$ and $\mathscr{B}, b \not\models \mathbf{G}(p \wedge q)$.

The same reasoning applies in general and hence it follows that limits inherit the properties of their objects while these satisfy those of their colimits, but the converse implications do not always hold.

8 Factorizations in Categories of Simulations

First we characterize the classes of AP-simulations that correspond to the (regular) epimorphisms and monomorphisms.

Proposition 15. *A morphism in* $\mathbf{KMap}_{AP}$, $\mathbf{KMap}^{\mathrm{str}}_{AP}$, *or* $\mathbf{KSMap}_{AP}$ *is an epimorphism if and only if it is a surjective function.*

Proof. Assume that $f : \mathscr{A} \longrightarrow \mathscr{B}$ is surjective. Then, if $g \circ f = h \circ f$ it must be the case that $g = h$, and hence f is epi, because the range of f is B.

Conversely, assume now that f is an epimorphism. If f were not surjective there would be an element $b \in B$ not in the image of f. Define a Kripke structure $\mathscr{B}'$ which is like $\mathscr{B}$ but with b replaced by b_1 and b_2 with the same labeling as b and such that $b_i \rightarrow_{\mathscr{B}'} b'$ iff $b \rightarrow_{\mathscr{B}} b'$ and $b' \rightarrow_{\mathscr{B}'} b_i$ iff $b' \rightarrow_{\mathscr{B}} b$. Now, if $g : \mathscr{B} \longrightarrow \mathscr{B}'$ maps b to b_1 and the other elements to themselves, and $h : \mathscr{B} \longrightarrow \mathscr{B}'$ maps b_2 to b and is the identity elsewhere, we have that g and h are well-defined (strict/stuttering) AP-maps, $g \circ f = h \circ f$, but $g \neq h$: a contradiction with the assumption that f was an epimorphism. □

Proposition 16. *A morphism in* $\mathbf{KMap}_{AP}$, $\mathbf{KMap}^{\mathrm{str}}_{AP}$, *or* $\mathbf{KSMap}_{AP}$ *is a monomorphism if and only if it is injective over paths (which is weaker than just injectivity).*

Proof. Assume that $f : \mathscr{A} \longrightarrow \mathscr{B}$ is injective over paths, that is, that the function f from $\mathrm{Path}(\mathscr{A})$ to $\mathrm{Path}(\mathscr{B})$ defined by $f(\pi) = f \circ \pi$ is injective. Let $g, h : \mathscr{C} \longrightarrow \mathscr{A}$ be morphisms such that $f \circ g = f \circ h$. Let $c \in C$ and π be a path starting at c: then it is $f(g(\pi)) = f(h(\pi))$ from where it follows $g(\pi) = h(\pi)$ and therefore $g(c) = h(c)$.

Conversely, assume that f is mono but there are paths π and π' in $\mathscr{A}$ such that $f(\pi) = f(\pi')$ and $\pi \neq \pi'$. Let us define a Kripke structure $\mathscr{C}$ with a single path $c_1 \rightarrow_{\mathscr{C}} c_2 \rightarrow_{\mathscr{C}} \ldots$ and with $L_{\mathscr{C}}(c_i) = L_{\mathscr{A}}(\pi(i)) \cup L_{\mathscr{A}}(\pi'(i))$. Then, if $g, h : \mathscr{C} \longrightarrow \mathscr{A}$ are defined by $f(c_i) = \pi(i)$ and $g(c_i) = \pi'(i)$, g and h are AP-maps by construction (and strict, if f is so), and it is $f \circ g = f \circ h$ and $g \neq h$: a contradiction. □

The characterization of regular monomorphisms is now immediate.

Proposition 17. *A morphism* $f : \mathscr{A} \longrightarrow \mathscr{B}$ *is a regular mono in* $\mathbf{KMap}_{AP}$, $\mathbf{KMap}^{\mathrm{str}}_{AP}$, *or* $\mathbf{KSMap}_{AP}$ *if and only if* f *is injective,* $L_{\mathscr{A}}(a) = L_{\mathscr{B}}(f(a))$ *for all* $a \in A$, *and* $a \rightarrow_{\mathscr{A}} a'$ *iff* $f(a) \rightarrow_{\mathscr{B}} f(a')$.

Proof. The implication from left to right follows from the construction in the proof of Proposition 13. In the other direction, let $\mathscr{C}$ be the Kripke structure obtained from $\mathscr{B}$ by splitting each state b into b_1 and b_2, with $L_{\mathscr{C}}(b_1) = L_{\mathscr{C}}(b_2) = L_{\mathscr{B}}(b)$, and with

$b_i \to_{\mathscr{C}} b'_j$ iff $b \to_{\mathscr{B}} b'$. Now, define $f, g : \mathscr{B} \longrightarrow \mathscr{B}'$ by $g(b) = b_1$ and $h(b) = b_1$ if $b \in f(A)$ and $h(b) = b_2$ otherwise: f and g so defined are (strict) AP-maps and, since $f(\mathscr{A})$ is a Kripke substructure of $\mathscr{B}$ and it is isomorphic to $\mathscr{A}$ due to the assumptions, it is easy to check that the result of the construction of the equalizer in Proposition 13 is (isomorphic to) f. □

There is also a rather more involved characterization of regular epis which is described in the next proposition.

Proposition 18. *A morphism $f : \mathscr{A} \longrightarrow \mathscr{B}$ is a regular epi in* $\mathbf{KMap}_{AP}$ *or* $\mathbf{KMap}^{\mathrm{str}}_{AP}$ *if and only if: (1) $f(a) = f(a')$ implies that there are paths π and π' such that $\pi(0) = a$, $\pi'(0) = a'$, and $f(\pi) = f(\pi')$; (2) if $b \to_{\mathscr{B}} b'$ there exist $a, a' \in A$ with $f(a) = b$, $f(a') = b'$, and $a \to_{\mathscr{A}} a'$; (3) for all b, $L_{\mathscr{B}}(b) = \bigcap_{f(a)=b} L_{\mathscr{A}}(a)$.*

Proof. The implication from left to right follows from Proposition 14; item (2) is proved by induction over $\equiv$. In the other direction, we define a Kripke structure $\mathscr{C}$ and two AP-maps $g, h : \mathscr{C} \longrightarrow \mathscr{A}$ as follows. For each pair of states a, a' such that $f(a) = f(a')$ let π and π' be paths as in (1). Now, add to $\mathscr{C}$ a fresh path ρ in such a way that $g(\rho(i)) = \pi(i)$ and $h(\rho(i)) = \pi'(i)$. Then, if we apply the construction in Proposition 14 that returns the coequalizer of g and h through a quotient Kripke structure $\mathscr{A}/\equiv$, the result is, by items (2) and (3), isomorphic to $f : \mathscr{A} \longrightarrow \mathscr{B}$. □

Proposition 19. $\mathbf{KMap}_{AP}$, $\mathbf{KMap}^{\mathrm{str}}_{AP}$, *and* $\mathbf{KSMap}_{AP}$ *are (epi, regular mono)-categories.*

Proof. Let $f : \mathscr{A} \longrightarrow \mathscr{B}$ be a morphism in any of these categories, and let us write $f(\mathscr{A})$ for $(f(A), \to_{\mathscr{B}}|_{f(A)}, L_{\mathscr{B}}|_{f(A)})$. Then, define $e : \mathscr{A} \longrightarrow f(\mathscr{A})$ to be like f and $m : f(\mathscr{A}) \longrightarrow \mathscr{B}$ to be the obvious inclusion: (e, m) is the unique (epi, regular mono)-factorization of f (up to isomorphism).

By Propositions 15 and 17, e and m are indeed epi and regular mono respectively. Now, assume that $e' : \mathscr{A} \longrightarrow \mathscr{C}, m' : \mathscr{C} \longrightarrow \mathscr{B}$ is another (epi, regular mono)-factorization of f. Define $g : f(\mathscr{A}) \longrightarrow \mathscr{C}$ by $g(b) = e'(a)$ where $e(a) = b$ (recall that e is surjective), and $h : \mathscr{C} \longrightarrow f(\mathscr{A})$ by $h(c) = e(a)$ where $e'(a) = c$. Let us check that they are well-defined. If $e(a) = e(a')$ then $m(e(a)) = m(e(a'))$ and therefore $m'(e'(a)) = m'(e'(a'))$; now, since m' is regular mono, $e'(a) = e'(a')$ and g is well-defined, and analogously for h. It is also clear that they are inverses of each other and that $g \circ e = e'$ and $m' \circ g$, so we are only left with checking that they are simulations; we present the arguments for g: those for h are symmetric. If $b \to_{f(\mathscr{A})} b'$, where $e(a) = b$ and $e(a') = b'$, then $m(e(a)) \to_{\mathscr{B}} m(e(a'))$ or, equivalently, $m'(e'(a)) \to_{\mathscr{B}} m'(e'(a'))$ and thus, since m' is a regular mono, $e'(a) \to_{\mathscr{B}} e'(a')$ and hence $g(a) \longrightarrow_{f(\mathscr{A})} g(a')$. In the case of stuttering simulations, a path π in $f(\mathscr{A})$ translates to a path $m(\pi)$ in $\mathscr{B}$ which is m'-matched by ρ in $\mathscr{C}$; this same ρ also h-matches π. Finally, $L_{f(\mathscr{A})}(b) = L_{\mathscr{B}}(e(a)) = L_{\mathscr{B}}(m(e(a))) = L_{\mathscr{B}}(m'(e'(a))) = L_{\mathscr{C}}(e'(a)) = L_{\mathscr{C}}(g(b))$. The first equality assumes that $b = e(a)$, and the second and the fourth one hold because m and m' are regular monos. □

Although we do not have a counterexample we believe that (regular epi, mono) factorizations do not exist in general. When they do, however, they are unique: the argument is similar to that for (epi, regular mono)-factorizations.

9 Conclusions

In previous papers [13,11] we have studied the suitability of different kinds of simulations between transition systems and Kripke structures for the study of the relationships between formal models of concurrent systems. The range of available notions of simulations makes it very natural to adopt a categorical viewpoint in which Kripke structures become the objects of several categories while the morphisms are obtained from the corresponding notion of simulation. In this paper we have defined in detail several of those categories and studied their most interesting properties: minimal Kripke structures as opcartesian morphisms, borrowing of properties as cartesian morphisms, temporal logic institutions, constructions of limits and colimits, and factorizations.

There are two main directions left open for future work. On the one hand, we would like to finally prove or disprove the existence of limits in the Grothendieck categories. On the other hand, as briefly discussed in [11], rewriting logic theories representing Kripke structures can also be organized in categories: we plan to organize them in an institution and to study its relationship with $\mathscr{I}_{\mathbf{K}}$.

Acknowledgments. We would like to thank the anonymous referees for very interesting and useful comments.

References

1. M. Arrais and J. L. Fiadeiro. Unifying theories in different institutions. In M. Haveraaen, O. Owe, and O.-J. Dahl, editors, *Recent Trends in Data Type Specification. 11th Workshop on Specification of Abstract Data Types*, volume 1130 of *LNCS*, pages 81–101. Springer, 1996.
2. M. Barr and C. Wells. *Category Theory for Computing Science. Third Edition*. Centre de Recherches Mathématiques, 1999.
3. M. C. Browne, E. M. Clarke, and O. Grümberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59:115–131, 1988.
4. E. M. Clarke, O. Grumberg, and D. E. Long. Model checking and abstraction. *ACM Transactions on Programming Languages and Systems*, 16(5):1512–1542, Sept. 1994.
5. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 1999.
6. J. Goguen and R. Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.
7. H. Herrlich and G. E. Strecker. *Category Theory: An Introduction*. Advanced Mathematics. Allyn and Bacon, Boston, 1973.
8. B. Jacobs. *Categorical Logic and Type Theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1999.
9. S. Mac Lane. *Categories for the Working Mathematician. Second Edition*. Springer, 1998.
10. P. Manolios. *Mechanical Verification of Reactive Systems*. PhD thesis, University of Texas at Austin, Aug. 2001.
11. N. Martí-Oliet, J. Meseguer, and M. Palomino. Theoroidal maps as algebraic simulations. In J. L. Fiadeiro and P. Mosses and F. Orejas, editors, *Recent Trends in Algebraic Development Techniques, WADT 2004*, volume 3423 of *LNCS*, pages 126–143. Springer, 2005.
12. N. Martí-Oliet, J. Meseguer, and M. Palomino. Algebraic simulations. Submitted. `http://maude.sip.ucm.es/~miguelpt/bibliography`, 2005.
13. J. Meseguer, N. Martí-Oliet, and M. Palomino. Equational abstractions. In F. Baader, editors, *Automated Deduction - CADE-19*, volume 2741 of *LNCS*, pages 2–16. Springer, 2003.

14. K. S. Namjoshi. A simple characterization of stuttering bisimulation. In S. Ramesh and G. Sivakumar, editors, *Foundations of Software Technology and Theoretical Computer Science, 17th Conference*, volume 1346 of *LNCS*, pages 284–296. Springer, 1997.
15. A. Tarlecki, R. M. Burstall, and J. A. Goguen. Some fundamental algebraic tools for the semantics of computation. Part 3: Indexed categories. *Theoretical Computer Science*, 91(2):239–264, 1991.

A Proofs of Some of the Results

Proposition 7. $\mathscr{I}_{\mathbf{K}} = (\mathbf{Sign}_{\mathbf{K}}, sen_{\mathbf{K}}, \mathbf{Mod}_{\mathbf{K}}, \models)$ *is an institution.*

Proof. It is a routine exercise to check that the purported functors are actually so. For example, let us check that $\mathbf{Mod}_{\mathbf{K}}$ is well-defined. Given $\alpha : AP \longrightarrow AP'$, $\mathbf{Mod}_{\mathbf{K}}(\alpha)$ is a functor. It is well-defined over objects and preserves identities and composition: we only need to check that $\mathbf{Mod}_{\mathbf{K}}(\alpha)(H) : \mathscr{A}|_\alpha \longrightarrow \mathscr{B}|_\alpha$ is an AP-simulation whenever $H : \mathscr{A} \longrightarrow \mathscr{B}$ is an AP'-simulation. Since the transition systems do not change, $\mathbf{Mod}_{\mathbf{K}}(\alpha)(H)$ preserves the transition relation. Now, if aHb and $p \in L_{\mathscr{B}|_\alpha}(b)$, then by definition we have $\mathscr{B}, b \models \alpha(p)$ and, by Theorem 1, this yields $\mathscr{A}, a \models \alpha(p)$, which again by definition implies that $p \in L_{\mathscr{A}|_\alpha}(a)$, as required. Thus, $\mathbf{Mod}_{\mathbf{K}}$ is well-defined over both objects and morphisms. It clearly preserves identities, so we are only left with showing that it preserves composition, for which it is enough to show that, given arrows $\alpha : AP \longrightarrow AP'$ and $\beta : AP' \longrightarrow AP''$, and a Kripke structure $\mathscr{A}$ over AP'', $\mathscr{A}_{\beta\circ\alpha} = (\mathscr{A}|_\beta)|_\alpha$. The equality at the level of transition systems is immediate. For the labeling function, $p \in L_{\mathscr{A}|_{\beta\circ\alpha}}(a)$ iff $\mathscr{A}, a \models \overline{\beta}(\alpha(p))$ (by definition) iff $\mathscr{A}|_\beta \models \alpha(p)$ (by Proposition 1) iff $p \in L_{(\mathscr{A}|_\beta)|_\alpha}(a)$ (by definition). Finally the property required of the satisfaction relation follows from Proposition 1. □

Proposition 8. *Let $\mathscr{I}'_{\mathbf{K}}$ be obtained from the institution $\mathscr{I}_{\mathbf{K}}$ by replacing $\mathbf{Set}_{\mathrm{State}\setminus\neg}$ by* **Set** *as the category of signatures. Then $\mathscr{I}'_{\mathbf{K}}$ is a semiexact institution.*

Proof. That $\mathscr{I}'_{\mathbf{K}}$ is an institution is immediate, and since the category of signatures is **Set** we know that it has pushouts. Therefore, we are left with checking that pushouts are transformed into pullbacks by the model functor. Consider then a pushout

$$\begin{array}{ccc} AP_0 & \xrightarrow{\alpha_2} & AP_2 \\ {\scriptstyle\alpha_1}\downarrow & & \downarrow{\scriptstyle\beta_2} \\ AP_1 & \xrightarrow{\beta_1} & AP_3 = (AP_1 \uplus AP_2)/\equiv \end{array}$$

where $\equiv$ is the least equivalence relation on $AP_1 \uplus AP_2$ verifying $\alpha_1(p) \equiv \alpha_2(p)$, and β_1 and β_2 take each element to its quotient class. To see that it is mapped to a pullback, let $F_1 : C \longrightarrow \mathbf{KSim}_{AP_1}$ and $F_2 : C \longrightarrow \mathbf{KSim}_{AP_2}$ be functors such that $\_|_{\alpha_1} \circ F_1 = \_|_{\alpha_2} \circ F_2$; we have to find a unique functor $F : C \longrightarrow \mathbf{KSim}_{AP_3}$ such that $\_|_{\beta_1} \circ F = F_1$ and $\_|_{\beta_2} \circ F = F_2$.

Let c be an object in C and $f : c \longrightarrow c'$ an arrow, with $F_1(c) = \mathscr{A}$ and $F_2(c) = \mathscr{B}$. It follows from the hypothesis that A is equal to B, $\rightarrow_{\mathscr{A}}$ equal to $\rightarrow_{\mathscr{B}}$, and $F_1(f)$ equal

to $F_2(f)$. This leads us to define $F(c) = (A, \rightarrow_{\mathscr{A}}, L_{F(c)})$ and $F(f) = F_1(f)$, where we choose to define the labeling function as $L_{F(c)} = \beta_1(L_{\mathscr{A}}) \cup \beta_2(L_{\mathscr{B}})$. Since it is straightforward to check that $F(f)$ is an AP_3-simulation, F is well-defined, and it is a functor because F_1 (or F_2) is so.

We are left with checking that F satisfies the commutativity condition and proving that it is the only one that does it. For the first part, note that by the definition of the pushout it is not possible for any two p and p' in AP_1 to be such that $\beta_1(p) = \beta_1(p')$ and $p \in L_{\mathscr{A}}(a)$ but $p' \notin L_{\mathscr{A}}(a)$ (a detailed proof proceeds by induction on the definition of $\equiv$). We need to use this property to show that $F(c)|_{\beta_1} = F_1(c)$. We already know that their objects and transition relations are the same; as for the atomic predicates:

$$p \in L_{F(c)|_{\beta_1}}(a) \iff F(c), a \models \beta_1(p) \iff \beta_1(p) \in L_{F(c)}(a) \iff p \in L_{F_1(c)}(a),$$

where the property is required for the last implication to the right. The result for F_2 is symmetric. Uniqueness follows from the definitions of the functors and the previous equivalences. □

B Summary of Categories

The following table summarizes most of the categories introduced in this paper; for each of them, the third column contains the (co)limits for which explicit constructions have been given.

Categories	Objects	Arrows	(Co)limits
TSys	transition systems	simulations of transition systems	(Co)products
$\mathbf{KSim}_{AP}$	Kripke str. over AP	AP-simulations	(Co)products
$\mathbf{KMap}_{AP}$	Kripke str. over AP	AP-simulation maps	(Co)products, (co)equalizers
$\mathbf{KMap}_{AP}^{\mathrm{str}}$	Kripke str. over AP	strict AP-simulations	(Co)products, (co)equalizers
$\mathbf{KSSim}_{AP}$	Kripke str. over AP	stuttering AP-simulations	?
$\mathbf{KSMap}_{AP}$	Kripke str. over AP	stuttering AP-simulation maps	Equalizers
KSim	arbitrary Kripke str.	simulations	?/colimits as above
KSSim	arbitrary Kripke str.	stuttering simulations	?/colimits as above