Presheaves as Transition Systems

Glynn Winskel Mogens Nielsen
BRICS
Department of Computer Science
University of Aarhus, Denmark

August 23, 1996

1 Introduction

Recall, as background, the content of the handbook chapter [22]. There, a model for process calculi is presented as a class of objects (like transition systems, or Petri nets), equipped with a notion of morphism, so that it forms a category. The morphisms represent a form of simulation between processes, and arise naturally in relating the behaviour of a construction on processes to that of its components. Basic operations of process calculi may now be understood as universal constructions (like product and coproduct) of the category, and so are characterised abstractly, up to isomorphism. Categorical notions also come into play in relating different models, for instance, in relating the model of transition systems to that of Petri nets. Adjunctions, especially coreflections, provide a way to translate between one model and another. The understanding of the operations of process calculi as universal constructions guides definitions away from the *ad hoc*, while the preservation properties of adjoints help relate semantics in one model to a semantics in another.

The richness of the morphisms in the categories of models, a richness which is essential in yielding the universal constructions, means that many objects with strikingly different behaviours are connected by morphisms; in particular, morphisms of transition systems relate transition systems which are far from strongly, or weakly, bisimilar. The categories do not immediately yield useful abstract equivalences between processes. However, in [8] it is shown how a general concept of bisimulation arises from the definition of open map. The definition of open map, applicable to all the categories of models, picks out those morphisms which, roughly speaking, reflect as well as preserve behaviour. It is then sensible to take two processes to be bisimilar, in a generalised sense, if they are connected by open maps.

The definition of open map relies not just on a categorical presentation of a model (for example, as a category rather than just a class of transition systems) but also on an acceptance of a notion of computation path and what it means to extend a computation path by another. For the interleaving model of transition systems a reasonable idea is to take a computation path (or run) as a sequence of consecutive transitions, which we can think of as picked out by a morphism from a string of action labels; here it is hard to escape from the idea that extending a computation path is associated with extending the string of action labels. For an independence model like event structures a reasonable idea is to take a computation path as a configuration, or more generally as a morphism from a pomset to the event structure; this time several ideas suggest themselves as to how we might extend a computation path shaped like a pomset, because, roughly, we can extend a pomset in "width" (adding concurrent events) as well as "height" (adding later, causally dependent events).

In the case of familiar models like transition systems or event structures the general definition of bisimulation specialises to familiar concepts; in particular, on transition systems with strings of actions as paths we obtain Park and Milner's strong bisimulation.

Presheaves offer a method to derive a model directly from a path category whose objects are path shapes and whose morphisms describe the extension of one path by another. Forming the category of presheaves over a path category has the effect of freely closing the path category under small colimits. More intuitively, a presheaf represents the effect of gluing together a set of computation paths to form a nondeterministic computation; the category of presheaves can be thought of as a category of nondeterministic computations. This intuition is backed up by canonical embeddings of traditional models into categories of presheaves over appropriately chosen path categories (*cf.* Theorem 4). Because the original path category embeds via the Yoneda functor into the category of its presheaves, we automatically obtain a notion of open map and bisimulation on presheaves.

A range of models and their notion of bisimulation can be understood in a uniform way via their represention as presheaves. Here we emphasise the view that presheaves can be profitably looked upon as transition systems, in which the labels are morphisms of path extension. This yields transition-system characterisations of open maps and bisimulation on presheaves, and through these to generalisations of Hennessy-Milner logic and games, providing a more operational characterisation. In particular, bisimulation on presheaves coincides with back-and-forth bisimulation between their associated transition systems.

In a way, by regarding presheaves as transition systems we can repay a debt to the foundational influence transitions systems have had in the theory of concurrent computation. Many original motivations and intuitions were formed around the model of transition sytems. Through the medium of presheaves, we are able to cope uniformly with a range of models and their equivalences, from interleaving to independence models, and at the same time, by altering our view a little, see the approach as only a slight adjustment in the perspective that motivated Park and Milner's definition of strong bisimulation.

2 Models, Morphisms and Computation Paths

We quickly describe the models and notions of computation paths we'll use as running examples.

Transition systems consist of a set of states, with an initial state, together with transitions between states which are labelled to specify the kind of events they represent. Formally, a *transition system* is a structure

$$(S, i, L, \mathit{tran})$$

where

- S is a set of *states* with *initial state* i,
- L is a set of *labels*, and
- $\mathit{tran} \subseteq S \times L \times S$ is the *transition relation*. As usual, we write

  $$s \xrightarrow{a} s'$$

  to indicate that $(s, a, s') \in \mathit{tran}$.

A state s is said to be *reachable* when $i \xrightarrow{a_1} \cdots \xrightarrow{a_n} s$ for some, possibly empty, string $a_1 \cdots a_n$.

As morphisms on transition systems we take functions on states which preserve initial states and transitions. Let

$$T_0 = (S_0, i_0, L_0, \mathit{tran}_0) \text{ and}$$
$$T_1 = (S_1, i_1, L_1, \mathit{tran}_1)$$

be transition systems. A *morphism* $f : T_0 \to T_1$ is a function $\sigma : S_0 \to S_1$ such that $\sigma(i_0) = i_1$ and

$$(s, a, s') \in \mathit{tran}_0 \Rightarrow (\sigma(s), a, \sigma(s')) \in \mathit{tran}_1.$$

Morphisms on transition systems compose as functions. For the concerns of [22], morphisms on transitions systems were more general. They could change labels and even send labels to undefined. This is necessary in relating the behaviour of compound processes to that of their components in languages like Milner's CCS, and in obtaining a repertoire of universal constructions, rich enough to yield a general process language. Here we concentrate on bisimulation for which we can take the simpler label-preserving morphisms as our starting point—such label preserving morphisms play an important role in the categorical account of [22], for example, in understanding restriction and relabelling operations of CCS-like languages as universal constructions.

We'll call transition systems which look like trees *synchronisation trees*. More precisely, synchronisation trees are those transition systems with no loops, no distinct transitions *to* the same state, in which all states are reachable. Synchronisation trees inherit morphisms from transition systems, and themselves form a category. The inclusion of synchronisation trees in transition systems is a left adjoint to the functor unfolding a transition systems to a synchronisation tree.

Special synchronisation trees will play a role in our treatment of bisimulation. Consider a (finite) computation (or run) in a transition system T. It is a sequence of transitions

$$i = s_0 \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n$$

—the sequence might possibly be empty. Let us identify strings like $s = a_1 a_2 \cdots a_n$ in L^* with "path shapes", rather special synchronisation trees consisting of a single branch of transitions

$$\bullet \xrightarrow{a_1} \bullet \xrightarrow{a_2} \cdots \xrightarrow{a_n} \bullet \ .$$

Then the computation path in T is identified with the morphism

$$p : s \to T$$

picking out the chain of transitions in T. Morphisms between such path shapes, consisting of a single-branch synchronisation trees, inherited from transition systems correspond to extensions of the associated strings. So we can identify the category of such path shapes with the (partial-order) category of strings L^*; a morphism from string s to string t corresponds to s being an initial prefix of t.

We focus event structures as our primary example of an independence model—other independence models like Petri nets and Mazurkiewicz trace languages are related to event structures via adjunctions in [22] in such a way that they inherit a common notion of bisimulation (see [8, 15]).

Define a *(labelled) event structure* to be a structure $(E, \leq, Con, l)$ consisting of a set E, of *events* which are partially ordered by $\leq$, the *causal dependency relation*, a *consistency relation* Con consisting of finite subsets of events, and a *labelling function* $l : E \to L$, which satisfy

$$\begin{array}{l}\{e' \mid e' \leq e\} \text{ is finite,} \\ \{e\} \in Con, \\ Y \subseteq X \in Con \Rightarrow Y \in Con, \\ X \in Con \ \& \ e \leq e' \in X \Rightarrow X \cup \{e\} \in Con,\end{array}$$

for all events e, e' and their subsets X, Y.

Two events $e, e' \in E$ are said to be *concurrent* (causally independent) iff

$$(e \not\leq e' \ \& \ e' \not\leq e \ \& \ \{e, e'\} \in Con).$$

Define a *configuration* (or state) of E to be a subset $x \subseteq E$ which is

downwards-closed: $\forall e, e'.\ e' \leq e \in x \Rightarrow e' \in x$, and

consistent: $\forall X.\ X$ finite $\&\ X \subseteq x \Rightarrow X \in Con$.

As before, we restrict attention to label-preserving morphisms on event structures over a common labelling set L. Let $E = (E, \leq, Con, l), E' = (E', \leq', Con', l')$ be event structures over L. A *morphism* from E to E' consists of a function $\eta : E \to E'$ on events which preserves labels (*i.e.* $l = l' \circ \eta$) such that

> if x is a configuration of E, then ηx is a configuration of E' and if for $e_1, e_2 \in x$ their images are equal, *i.e.* $\eta(e_1) = \eta(e_2)$, then $e_1 = e_2$.

In the category of event structures, morphisms are composed componentwise. The definition of morphism on event structures is rather abrupt—see [22] for motivation.

In the case of an independence model like event structures a computation path carries more structure than simply a string of actions. This time we take path shapes to be finite pomsets. Pomsets are special event structures where all finite subsets of events are consistent. They are essentially labelled partial orders, and morphisms between them, got by restricting those of event structures, are injective functions which send downwards-closed sets to downwards-closed sets. Thus a morphism from pomset P to pomset Q may not just extend P by extra events but also relax the causal dependency relation; two events causally related in P may have images no longer causally related in Q. We separate the forms of morphismcoresponding to the different ways one pomset can extend another.

Definition: Let L be a labelling set. Define $\mathbf{Pom}_L$ tbe the full subcategory of event structures with objects finite pomsets with labels in L.

Say a morphism $m : P \to Q$ in $\mathbf{Pom}_L$ is a *prefix* morphism iff m preserves and reflects the causally dependency order. Define $\mathbf{Pom}_L^p$ to be the subcategory of $\mathbf{Pom}_L$ where all morphisms are prefix morphisms.

Say a morphism $m : P \to Q$ in $\mathbf{Pom}_L$ is an *augmentation* morphism iff m is epimorphic. Define $\mathbf{Pom}_L^a$ to be the subcategory of $\mathbf{Pom}_L$ where all morphisms are augmentation morphisms.

Proposition 1 *Any morphism $m : P \to Q$ factors uniquely to within isomorphism as a composition $m = P \xrightarrow{a} Q_0 \xrightarrow{j} Q$ where a is an augmentation and j is a prefix morphism.*

3 Open-Maps and Bisimulation

Assume a category of models $\mathbf{M}$–this could be any one of the categories of models with label preserving morphisms of the previous section. Assume also a choice of path category, a subcategory $\mathbf{P} \hookrightarrow \mathbf{M}$ consisting of path objects (these could be branches, or pomsets) together with morphisms expressing how they can be extended.

Define a *path* in an object X of $\mathbf{M}$ to be a morphism

$$p : P \to X,$$

in $\mathbf{M}$, where P is an object in $\mathbf{P}$. A morphism $f : X \to Y$ in $\mathbf{M}$ takes such a path p in X to the path $f \circ p : P \to Y$ in Y. The morphism f expresses the sense in which Y simulates X; any computation path in X is matched by the computation path $f \circ p$ in Y.

We might demand a stronger condition of a morphism $f : X \to Y$ expressed succinctly in the following *path-lifting condition* which when satisfied picks out the *open* morphisms. For our purposes later, it's convenient to define open morphisms with respect to a subclass of morphisms $\mathbf{P}_0$ of $\mathbf{P}$—of course $\mathbf{P}_0$ could consist of all the morphisms of the whole category $\mathbf{P}$, when we'll identify the class of morphisms with $\mathbf{P}$ itself.

Whenever, for $m : P \to Q$ a morphism in $\mathbf{P}_0$, a "square"

$$\begin{tikzcd} P \arrow[r, "p"] \arrow[d, "m"'] & X \arrow[d, "f"] \\ Q \arrow[r, "q"] & Y \end{tikzcd}$$

in $\mathbf{M}$ commutes, *i.e.* $q \circ m = f \circ p$, meaning the path $f \circ p$ in Y can be extended via m to a path q in Y, then there is a morphism p' such that in the diagram

$$\begin{tikzcd} P \arrow[r, "p"] \arrow[d, "m"'] & X \arrow[d, "f"] \\ Q \arrow[r, "q"'] \arrow[ur, "p'"] & Y \end{tikzcd}$$

the two "triangles" commute, *i.e.* $p' \circ m = p$ and $f \circ p' = q$, meaning the path p can be extended via m to a path p' in X which matches q. When the morphism f satisfies this condition we shall say it is $\mathbf{P}_0$-*open*.

Say two objects X_1, X_2 of $\mathbf{M}$ are $\mathbf{P}_0$-*bisimilar* iff there is a *span* of $\mathbf{P}_0$-open morphisms f_1, f_2:

$$\begin{tikzcd} & X \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\ X_1 & & X_2 \end{tikzcd}$$

For the well-known model of transition systems open morphisms and the bisimulation induced by them are already familiar:

Proposition 2 *With respect to a labelling set L, the L^*-open morphisms of the category of transition systems with labelling set L are the "zig-zag morphisms" of [20] (the "p-morphism" of [18], the "abstraction homomorphisms" of [4], and the "pure morphisms" of [3]) i.e. those label-preserving morphisms $(\sigma, 1_L) : T \to T'$ on transition systems over labelling set L with the property that for all reachable states s of T*

if $\sigma(s) \xrightarrow{a} s'$ in T' then $s \xrightarrow{a} u$ in T and $\sigma(u) = s'$,
for some state u of T.

Two transition systems (and so synchronisation trees), over the same labelling set L, are L^*-bisimilar iff they are strongly bisimilar in the sense of [12].

In the case of event structures with $\mathbf{Pom}_L$ as the path category we obtain the equivalence of strong history preserving bisimulation on event structures (see [8] or [15]).

In checking whether a morphism is $\mathbf{P}$-open or for $\mathbf{P}$-bisimulation, for a path category $\mathbf{P}$, it suffices to consider a restricted class of morphisms, sufficient to generate the catgegory $\mathbf{P}$.

Definition: Let $\mathbf{P}$ be a category. Let $\mathbf{P}_0$ consist of a subclass of morphisms of $\mathbf{P}$. Say $\mathbf{P}_0$ *generates* $\mathbf{P}$ iff the only subcategory of $\mathbf{P}$ which includes $\mathbf{P}_0$ and all isomorphisms of $\mathbf{P}$ is $\mathbf{P}$ itself.

In particular, if $\mathbf{P}_0$ is a skeletal subcategory of $\mathbf{P}$, then $\mathbf{P}_0$ generates $\mathbf{P}$.

Example: The category L^* is generated by the set of morphisms representing the extension of a string by a single label.

The category $\mathbf{Pom}_L$ is generated by the class of "atomic" morphisms of two kinds:

> *prefix:* morphisms $m : P \to Q$ in $\mathbf{Pom}_L$ expressing that pomset P is a prefix of pomset Q where Q contains one more event than P; so m expresses that pomset Q consists of a copy of P with one additional event adjoined on top;
>
> *augmentation:* morphisms $m : P \to Q$ in $\mathbf{Pom}_L$ expressing that pomset P is an augmentation of pomset Q but where the graph of the causal dependency relation in P contains one more pair than that of Q; so the pomset P consists of a copy of Q with one extra link of causal dependency between previously concurrent events.

To see that this class of morphisms generates $\mathbf{Pom}_L$, note that any morphism $m : P \to Q$ in $\mathbf{Pom}_L$ factors uniquely (to within isomophism) as a composition $m = j \circ a$ where

$$a : P \to Q_0$$

expresses that P is an augmentation of Q_0 and

$$j : Q_0 \to Q$$

expresses that Q_0 is a prefix of Q. Then, clearly any augmentation, or prefix, breaks down into a composition of basic augmentations, or prefixes, respectively, as above.

Clearly $\mathbf{Pom}_L^p$ is generated by the atomic prefix morphisms while $\mathbf{Pom}_L^a$ is generated by the atomic augmentation morphisms described above.

Proposition 3 *Suppose* $\mathbf{P}$ *is generated by a subclass of morphisms* $\mathbf{P}_0$*.*

1. *Letting* f *be a morphism of* $\mathbf{M}$*,* f *is* $\mathbf{P}$*-open iff* f *is* $\mathbf{P}_0$*-open.*
2. *Let* X_1, X_2 *be objects of* $\mathbf{M}$*. Then,* X_1, X_2 *are* $\mathbf{P}$*-bisimilar iff* X_1, X_2 *are* $\mathbf{P}_0$*-bisimilar.*

4 Presheaf Models

Given a path category $\mathbf{P}$ we can build the category $\widehat{\mathbf{P}}$ of presheaves over $\mathbf{P}$.[1] The objects of $\widehat{\mathbf{P}}$ consist of functors $\mathbf{P}^{op} \to \mathbf{Set}$, to the category of sets. The

[1]Proofs for presheaf models can be found in [8]. A good introduction to presheaves can be found in Chapter 1 of [10].

morphisms of $\widehat{\mathbf{P}}$ are natural transformations between functors. Intuitively a presheaf $F : \mathbf{P}^{op} \to \mathbf{Set}$ can be thought of as specifying for a typical path object P the set $F(P)$ of paths from P. It acts on a morphism $m : P \to Q$ in $\mathbf{P}$ to give a function $F(m) : F(Q) \to F(P)$ saying how Q-paths restrict to P-paths.

Let us see how a model, like a transition system or a labelled event structure, gives rise to a presheaf. Consider a category of models $\mathbf{M}$ and a choice of path category forming a subcategory $\mathbf{P} \hookrightarrow \mathbf{M}$. There is a *canonical functor* from the category of models $\mathbf{M}$ to the category of presheaves $\widehat{\mathbf{P}}$. It takes an object X of $\mathbf{M}$ to the presheaf $\mathbf{M}(-, X)$—more intuitively, it takes the model X to the to the presheaf which for each path object P yields the set of paths $\mathbf{M}(P, X)$ from P into X. The canonical functor takes a morphism $f : X \to Y$ in $\mathbf{M}$ to the natural transformation

$$\mathbf{M}(-, f) : \mathbf{M}(-, X) \to \mathbf{M}(-, Y)$$

whose component at an object P of $\mathbf{P}$ is the function $\mathbf{M}(P, X) \to \mathbf{M}(P, Y)$ taking p to $f \circ p$—intuitively, a path $p : P \to X$ in X is taken to a path $f \circ p : P \to Y$ in Y.

Theorem 4

(i) The canonical functor synchronisation trees, all with labelling set L, to $\widehat{L^*}$ is full, faithful and dense.

(ii) The canonical functor from event structures,all with labelling set L, to $\widehat{\mathbf{Pom}_L}$ is full faithful and dense.

The embeddings of theorem 4 extend the Yoneda embedding of $\mathbf{P} \to \widehat{\mathbf{P}}$, regarding a path object P as the presheaf $\mathbf{P}(-, P) = \mathbf{M}(-, P)$ because, in these cases, the subcategory $\mathbf{P} \hookrightarrow \mathbf{M}$ is full. Now, if we regard presheaves as the model $\mathbf{M}'$ and the image of $\mathbf{P}$ under the Yoneda embedding as its path category $\mathbf{P}'$, we can apply the general definition of section 3, to obtain the class of $\mathbf{P}'$-open morphisms of the presheaf category. They form a category of *open maps* of the topos $\widehat{\mathbf{P}}$, in the sense of Joyal and Moerdijk.[2] The two notions of $\mathbf{P}$-open and open map agree for the models of synchronisation trees and event structures, because generally:

Proposition 5 *Let $\mathbf{P}$ be a dense, full subcategory of $\mathbf{M}$. A morphism $f : X \to Y$ of $\mathbf{M}$ is $\mathbf{P}$-open iff the morphism $\mathbf{M}(-, f) : \mathbf{M}(-, X) \to \mathbf{M}(-, Y)$ is an open map of presheaves.*

[2] See [7] P.3, Example 1.1, though there the definition is expressed in terms of the existence of certain quasi-pullbacks; its equivalence with $\mathbf{P}'$-openness, expressed as a path-lifting property, follows by the Yoneda Lemma.

When it comes to relating notions of bisimilarity, we must be a little careful. It is not the case that two synchronisation trees are L^*-bisimilar iff their associated presheaves are related by a span of open maps in $\widehat{L^*}$. But this is only because there are presheaves which correspond to processes without an initial state; in particular, there is always a span of open maps between any two presheaves subtended from the initial (always empty) presheaf.

A way to get a correspondence is to restrict the objects in the presheaf category.

Definition: In the situation where the path category $\mathbf{P}$ of a model $\mathbf{M}$ has an initial object I, a *rooted presheaf* is a presheaf F in which $F(I)$ is a singleton.

Remark: Another way to get a correspondence is to define bisimilarity in the entire presheaf category via spans of *surjective* open maps. This is the more robust definition, and indeed the one used in [8]; it applies even when the path category does not have an initial object, and open maps between rooted presheaves are necessarily surjective (see *e.g.* [23]).

Proposition 6

(i) Two synchronisation trees, over labelling set L, are L^*-bisimilar (i.e. strong bisimilar) iff their corresponding presheaves, under the canonical embedding, are related by a span of open maps in the full subcategory of rooted presheaves of $\widehat{L^*}$.

(ii) Two event structures, over labelling set L, are $\mathbf{Pom}_L$-bisimilar (i.e. strong history-preserving bisimilar) iff their corresponding presheaves, under the canonical embedding, are related by a span of open maps in the full subcategory of rooted presheaves of $\widehat{\mathbf{Pom}_L}$.

5 Presheafs as Transition Systems

Assume that a path category $\mathbf{P}$ has an initial object I, and that $\mathbf{P}_0$ is a subclass of morphisms of $\mathbf{P}$.

It will be helpful to think of a rooted presheaf over $\mathbf{P}$ as a transition system with labels taken from morphisms of $\mathbf{P}_0$:

Definition: Let X be a rooted presheaf over $\mathbf{P}$. Define its $\mathbf{P}_0$-*transition system*, denoted by $\mathcal{E}l_{\mathbf{P}_0}(X)$ to consist of:

> *states :* (P, p) where P is an object of $\mathbf{P}$ and $p \in X(P)$; take the unique member of $X(I)$ as the initial state;
>
> *labelling set:* $\mathbf{P}_0$;

transitions: $(P,p) \xrightarrow{m} (Q,q)$ whenever $m : P \to Q$ in $\mathbf{P}_0$ and $(Xm)(q) = p$.

Remark: The construction $\mathcal{E}l_{\mathbf{P}_0}(X)$ on a presheaf X is a slight generalisation of a well-known construction of a *category of elements* of a presheaf (see *e.g.* [10]).

Notice what the construction does on a presheaf X: it forms a transition system with "states" $p \in X(P)$ which by the Yoneda lemma correspond 1-1 with the computation paths from P (or strictly its image under the Yoneda embedding) into X.

Given a morphism of presheaves, *i.e.* a natural transformation between them, we obtain a morphism of transition systems; $\mathcal{E}l_{\mathbf{P}_0}$ extends to a functor on presheaves.

Definition: Suppose $f : X \to Y$ is a natural transformation between presheaves X and Y. Define $\mathcal{E}l_{\mathbf{P}_0}(f)$ to be the morphism of transition systems σ which acts on states so that $(P,p) \mapsto (P, f_P(p))$; thus the transition $(P,p) \xrightarrow{m} (Q,q)$ is sent to the transition $(P, f_P(p)) \xrightarrow{m} (Qf_Q(q))$.
(It takes a little checking that $\mathcal{E}l_{\mathbf{P}_0}(f)$ is indeed a morphism of transition sytems.)

So, thinking of categories of elements as transition systems, the associated functor is a label-preserving morphism of transition systems. More than this, provided $\mathbf{P}_0$ generates $\mathbf{P}$, a natural transformation f between presheaves is open iff $\mathcal{E}l_{\mathbf{P}_0}(f)$ is a zig-zag morphism between the associated transition systems (*cf.* Proposition 2).

Proposition 7 *Assume* $\mathbf{P}_0$ *generates* $\mathbf{P}$. *A morphism* $f : X \to Y$ *between rooted presheaves in* $\widehat{\mathbf{P}}$ *is open iff* $\mathcal{E}l_{\mathbf{P}_0}(f) : \mathcal{E}l_{\mathbf{P}_0}(X) \to \mathcal{E}l_{\mathbf{P}_0}(Y)$ *is a zig-zag morphism between transition systems with labelling sets* $\mathbf{P}_0$.

We can go further and characterise bisimulation on presheaves as a form of bisimulation on transition systems with labels in a generating class of morphisms $\mathbf{P}_0$.

Definition: Say two transition systems T_1, T_2 with a common label set are *back-and-forth bisimilar* iff there is a relation R between their states such that $i_1 R i_2$, so their initial states are related, and whenever $s_1 R s_2$, then

if $s_1 \xrightarrow{a} s_1'$ then $s_2 \xrightarrow{a} s_2'$ and $s_1' R s_2'$, for some state s_2' of T_2,

if $s_2 \xrightarrow{a} s_2'$ then $s_1 \xrightarrow{a} s_1'$ and $s_1' R s_2'$, for some state s_1' of T_1

if $s_1' \xrightarrow{a} s_1$ then $s_2' \xrightarrow{a} s_2$ and $s_1' R s_2'$, for some state s_2' of T_2, and if $s_2' \xrightarrow{a} s_2$ then $s_1' \xrightarrow{a} s_1$ and $s_1' R s_2'$, for some state s_1' of T_1.

Proposition 8 *Let X_1, X_2 be presheaves over $\mathbf{P}$. Assume $\mathbf{P}_0$ is a subclass of morphisms generating $\mathbf{P}$. The presheaves X_1, X_2 are $\mathbf{P}$-bisimilar iff their transition systems $\mathcal{E}l_{\mathbf{P}_0}(X_1), \mathcal{E}l_{\mathbf{P}_0}(X_2)$ are back-and-forth bisimilar.*

Remark: Though this result is presented in a different guise it consists essentially of Lemma 17 in [8] characterising bisimulation between rooted presheaves as strong path bisimulation.
Warning: This result should *not* be interpreted in the broader sense that we advocate back-and-forth bisimulation as the appropriate bisimulation on transition systems. In fact, the presheaves in $\widehat{L^*}$ of transition systems with labelling set L, obtained by the canonical functor from transition system to presheaves, will be bisimilar iff the original transition systems are strongly bisimilar in the sense of Park and Milner.

Example:
Paths as strings: When we specialise to the (partial order) category of strings L^*, the subcategory of rooted presheaves in $\widehat{L^*}$ is equivalent to the category of synchronisation trees. Bisimulation between rooted presheaves in $\widehat{L^*}$ is reduced to back-and-forth bisimulation based on extensions of strings by a single label. Thus bisimulation between rooted presheaves coincides with back-and-forth bisimulation on synchronisation trees, and as is well-known [13] this coincides with Park and Milner's strong bisimulation. As remarked above, the bisimulation on transition systems induced by the canonical functor to $\widehat{L^*}$ is strong bisimulation.
Paths as pomsets: Two subcategories of rooted presheaves are of interest, those over path categories $\mathbf{Pom}_L$ and $\mathbf{Pom}_L^s$.

In the case of bisimulation between presheaves over $\mathbf{Pom}_L$ it suffices to consider "atomic" prefix and augmentation morphisms. Presheaves of event structures under the canonical embedding are bisimilar iff the event structures are strong history-preserving bisimilar (see [8] for the proof).

Just for the moment, consider the full subcategory of event structures, over labelling set L, where morphisms $\eta : E \to E'$ are further constrained to satisfy:

> if x is a configuration of E, then ηx is a configuration of E', and the restriction of η from x to ηx is an isomorphism of pomsets.
> ((–Here we identify a configuration of an event structure E with its pomset structure induced by E.–))

We call such morphisms *prefix* morphisms because they generalise their namesakes on pomsets. The canonical functor from the category of event structures with prefix morphisms, to rooted presheaves in $\widehat{\mathbf{Pom}_L^s}$ is full and faithful (because the category of pomsets with prefix morphisms is dense in the category of event structures with prefix morphisms). Under it two event structures give rise to bisimilar presheaves iff they are strong history-preserving bisimilar. This is essentially because if we look at the transition system of the presheaf obtained from an event structure, its states will correspond to configurations of the event structure.

Thus, strong history-preserving bisimulation of event structures coincides with bisimulation of the canonical presheaves (obtained by the canonical embedding) in the presheaves $\mathbf{Pom}_L$, and also with bisimulation between the canonical presheaves over just $\mathbf{Pom}_L^p$, where we restrict to simply prefix morphisms of pomsets and event structures. In investigating the bisimilarity of event structures it suffices to consider just "atomic" prefix morphisms in $\mathbf{Pom}_L^s$ where a single new event is adjoined.

6 Logic and Game Corollaries

By characterising bisimulation on presheaves as back-and-forth bisimulation on their associated transition systems we can connect with logic and game characterisations of bisimulation of the kind discussed in [12] (for logic) and [19] (for games and logic).

6.1 A specification logic

Assume the path category $\mathbf{P}$ is a small subcategory with initial object I. Let $\mathbf{P}_0$ be a subclass of morphisms of $\mathbf{P}$.

Define $\mathbf{P}_0$-*assertions* by:

$$A ::= \overline{\langle m \rangle} A \mid \langle m \rangle A \mid \neg A \mid \bigwedge_{j \in J} A_j$$

where m is a morphism in $\mathbf{P}_0$, and J is an indexing set, possibly empty and *not* restricted to being finite. The modality $\overline{\langle m \rangle}$ is an "backwards" modality, while $\langle m \rangle$ is a "forwards" modality. We define the semantics with respect to a transition system with labelling set $\mathbf{P}_0$:

- $s \models \langle m \rangle A$ iff $\exists s'.\ s \xrightarrow{m} s'$ and $s' \models A$
- $s \models \overline{\langle m \rangle} A$ iff $\exists s'.\ s' \xrightarrow{m} s$ and $s' \models A$
- the boolean operations receive their expected meanings.

The logic is but a step away from Hennessy-Milner logic, well-known to be characteristic for strong bisimulation, and the proof is virtually the same (see [12, 8]).

Theorem 9 *Let $\mathbf{P}_0$ generate $\mathbf{P}$. Two rooted presheaves in $\widehat{\mathbf{P}}$ are bisimilar iff their $\mathbf{P}_0$-transition systems satisfy the same assertions.*

Example: We determine a satisfaction relation for synchronisation trees and event structures via their canonical embeddings in presheaf categories $\widehat{L^*}$, $\widehat{\mathbf{Pom}_L}$ and $\widehat{\mathbf{Pom}^s_L}$; for a more direct definition of the satisfaction relation for these concrete models, based on their paths see [8].
Paths as strings: Traditional Hennessy-Milner logic arises by reducing the seemingly richer logic based on all extension morphisms in L^*. Firstly, as remarked earlier we can restrict to just the forwards modalities; for synchronisation trees back-and-forth bisimulation amounts to strong bisimulation. Because extensions by a single symbol are enough to generate the category of strings L^*, it suffices in getting a logic characteristic for bisimulation on synchronisation trees to restrict to forward modal assertions of the form $\langle b \rangle A$ where b is a single label; specifying the label b together with either the domain of the morphism is enough to determine the morphism in the path category.
Paths as pomsets: Bisimulation between rooted presheaves over $\mathbf{Pom}_L$ or $\mathbf{Pom}^s_L$ is characterised by satisfaction of assertions with modalities labelled by "atomic" morphisms. The category of event structures, with labelling set L, with prefix morphisms embeds canonically in $\widehat{\mathbf{Pom}^s_L}$. So strong history-preserving bisimulation of event structures is characterised by logic with forwards and backwards modalities labelled by "atomic" prefix morphisms. In the case where the event structures have no autoconcurrency (*i.e.* no concurrent events with the same label) the labels associated with the modalities can be simplified to single labels—see [14].

6.2 Games on presheaves

Assume again that the path category $\mathbf{P}$ is a small subcategory with initial object I, and that $\mathbf{P}_0$ be a subclass of morphisms of $\mathbf{P}$.

Viewing presheaves as transition systems, we may also lift existing notions of games for transition systems to presheaves. As an example we adopt here a back-and-forth version of the games for transition systems defined by e.g. [19], well known to be characteristic for strong bisimulation, to .

Let $T_0 = (S_0, i_0, L_0, tran_0)$ and $T_1 = (S_1, i_1, L_1, tran_1)$ be two transition systems. The game $G(T_0, T_1)$ played by two players (I and II) is defined as follows. The configurations of the game consists of pairs of states ($s_0 \in S_0, s_1 \in S_1$) with (i_0, i_1) as the starting configuration. A *play* consists of a sequence of alternating moves by the two players (Player I making the first move), where a move consists of a choice of a transition from one of the systems, according to the following game rules:

At configuration (s_0, s_1)

- either Player I chooses a transition $s_0 \xrightarrow{a} s_0'$, after which Player II chooses a transition $s_1 \xrightarrow{a} s_1'$, and the game continues at configuration (s_0', s_1'),

- or Player I chooses a transition $s_1 \xrightarrow{a} s_1'$, after which Player II chooses a transition $s_0 \xrightarrow{a} s_0'$, and the game continues at configuration (s_0', s_1')

- or Player I chooses a transition $s_0' \xrightarrow{a} s_0$, after which Player II chooses a transistion $s_1' \xrightarrow{a} s_1$, and the game continues at configuration (s_0', s_1')

- or Player I chooses a transition $s_1' \xrightarrow{a} s_1$, after which Player II chooses a transition $s_0' \xrightarrow{a} s_0$, and the game continues at configuration (s_0', s_1').

Player I wins a play if Player II gets stuck, i.e. at some point cannot match a move by Player I according to the rules of the game. All other plays are won by Player II, i.e. all infinite plays and plays where Player at some point cannot make a move. A (history-free) *strategy* for a player is a set of rules which for each configuration tells the player how to proceed, i.e. for Player II a rule will associate to each configurations and a choice of back or forth transition in one of the systems by Player I, a set of matching transitions in the other system. A strategy is *winning* for a player, if he or she wins every play played according to the strategy.

Intuitively, the two players have different goals in game $G(T_0, T_1)$: Player I wants to show that the two transition systems are distinguishable, Player II that they are not. Viewing presheaves as transition systems notion of distinguishablity is determined by:

Theorem 10 *Let* $\mathbf{P}_0$ *generate* $\mathbf{P}$*. Two rooted presheaves in* $\widehat{\mathbf{P}}$ *are bisimilar iff Player II has a winning strategy in the game defined by their two* $\mathbf{P}_0$*-transition systems.*

This theorem follows from Theorem 8 by essentially the proof of the corresponding theorem for transition systems from [19].

Example: Games for synchronization trees and event structures are obtained from their cannonical embeddings in presheaf categories.
Paths as strings: We obtain the original Stirling games characteristic for synchronization trees in the same way we obtained the original Hennessy-Milner logic above. First of all, from [13] we can restrict the games to only forwards moves, i.e. transitions labelled by extension morphisms. Secondly, from the theorem above we may restrict games to allow only moves involving extension with a single symbol, and finally such a morphism in the path

category is determined by its domain and the label of the extended single symbol.

Paths as pomsets: Bisimulation between rooted presheaves over $\mathbf{Pom}_L$ or $\mathbf{Pom}^s_L$ is characterised by games with moves restricted to transitions labeled by "atomic" morphisms. We may obtain games for event structures via their canonical embedding in $\widehat{\mathbf{Pom}^s_L}$, and hence we get that games with moves restricted to forwards and backwards transitions labelled by "atomic" prefix morphisms are characteristic for strong history-preserving bisimulation of event structures.

7 Concluding Remarks

So are presheaves *just* transition systems? No, they are really much more. While it can provide helpful intuition to think of presheaves as transition systems, presheaves possess a great deal of mathematical structure, which has already proved useful, or is potentially useful. For instance, there are results like that of [5] showing that constructions obtained from certain left Kan extensions automatically preserve open maps, and observations like that of [23], that moving to the category of *profunctors*, essentially presheaves acting as morphisms, we can begin to tackle higher-order features like process-passing; in recent work there appear to be technical advantages in viewing profunctors as transition systems. More speculatively, we can hope that the fact that presheaves form a topos will become helpful.

References

[1] Barr, M., and Wells, C., Category theory for Computer Science. Prentice Hall, 1990.

[2] Bednarczyk, M., Hereditary history preserving bisimulation or what is the power of the future perfect in program logics. Technical report, Polish Academy of Sciences, Gdansk, 1991.

[3] Benson, D.B., and Ben-Shachar, O., Bisimulation of automata. Information and Computation, 79, pp. 60–83, 1988.

[4] Castellani, I., Bisimulation and abstraction homomorphisms. Proc. of CAAP 85, Springer Lecture Notes in Computer Science, 1985.

[5] Cattani, G-L. and Winskel, G., Presheaf models for concurrency. To appear in the proc. of CSL'96, Utrecht, August 1996.

[6] Van Glabeek, R.J., and Goltz, U., Equivalence notions for concurrent systems and refinement of actions. Proc of MFCS, Springer Lecture Notes in Computer Science 379, 1989.

[7] Joyal, A., and Moerdijk, I., A completeness theorem for open maps. In Annals of Pure and Applied Logic 70, 51-86, 1994.

[8] Joyal, A., Nielsen, M. and Winskell, G., Bisimulation from Open Maps. Information and Computation, 127, no. 2, pp. 164–185, 1996.

[9] MacLane, S., Categories for the Working Mathematician. Graduate Texts in Mathematics, Springer, 1971.

[10] MacLane, S., and Moerdijk, I., *Sheaves in geometry and logic: a first introduction to topos theory*, Springer, 1992.

[11] Milner,A.J.R.G.,Calculus of communicating systems. Springer Lecture Notes in Computer Science 92, 1980.

[12] Milner,A.J.R.G., *Communication and concurrency.* Prentice Hall, 1989.

[13] De Nicola, R., Montanari, U., and Vaandrager, F., Back and Forth Bisimulations. Proceedings of CONCUR'90, Springer Lecture Notes in Computer Science, vol.458, pp.152–165, 1990.

[14] Nielsen, M., and Clausen, C., Bisimulations, Games, and Logic. Proceedings of CONCUR'94, Springer Lecture Notes in Computer Science, 836, pp. 385–400, 1994.

[15] Nielsen, M., and Winskel, G., Petri nets and bisimulations. To appear in Theoretical Computer Science. It also appears as BRICS Research Report RS-95-4, Computer Science Dept., Aarhus University, 1995.

[16] Pratt, V.R., Modelling concurrency with partial orders, *International Journal of Parallel Programming, 15,1*, pp. 33–71, 1986.

[17] Rabinovitch, A., and Traktenbrot, B., Behaviour structures and nets. Fundamenta Informatica, 11(4), pp. 357–404, 1988.

[18] Segerberg, K., Decidability of S4.1, Theoria 34, pp.7-20, 1968.

[19] Stirling, C., Model Checking and Other Games. Notes for Mathfit Workshop on Finite Model Theory, University of Wales Swansea, 1996.

[20] Van Benthem, J., Correspondence theory. In the Handbook of Philosophical Logic, Vol.II, ed. Gabbay and Guenther, Reidel, pp. 167–247, 1984

[21] Winskel,G., Event structures. Springer Lecture Notes in Computer Science 255, 1987.

[22] Winskel, G., and Nielsen, M., Models for concurrency. In the Handbook of Logic in Computer Science, vol. IV, ed. Abramsky, Gabbay and Maibaum, Oxford University Press, 1995.

[23] Winskel, G., A presheaf semantics of value-passing processes. To appear in the proc. of CONCUR'96, Pisa, August 1996.