Prefix orders as a general model of dynamics

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Prefix Orders as a General Model of Dynamics

P.J.L. Cuijpers

13/09

ISSN 0926-4515

Reports are available at:
http://library.tue.nl/catalog/TUEPublication.csp?Language=dut&Type=ComputerScienceReports&Sort=Author&level=1 and
http://library.tue.nl/catalog/TUEPublication.csp?Language=dut&Type=ComputerScienceReports&Sort=Year&Level=1

Computer Science Reports 13-09
Eindhoven, December 2013

Prefix Orders as a General Model of Dynamics

P.J.L. Cuijpers

December 20, 2013

Abstract

In this report we formalize and study the notion of prefix order on the executions of general dynamical systems and use basic category theory to show that appropriate structure preserving maps between such orders lead to the well-known notions of bisimulation, refinement, product, and union of behavior, without relying on a notion of 'next state'. Thus these notions are generalized to apply to arbitrary dynamical systems, including continuous and hybrid systems [1].

1 Introduction

When confronted with a new type of dynamical system, one of the the first questions that arises is: "how does this system behave?" Also, any book that studies the dynamics of a computational system, a control system, a physical system, a biological system, etc., starts by defining in some way "what the executions of such a system look like." As an example, in automata and process theory, executions are described as runs over a transition system [1], while in control theory, executions are usually functions of time to some variable-space [10]. In hybrid and cyber-physical systems theory, these two notions have been combined by defining time as a mix of continuous and discrete steps [4].

The notion of 'a set of executions' appears to be crucial in the study of dynamical systems, and while executions are often defined as functions of time, there is still much debate on what an appropriate notion of 'time' is. For this reason, we would like to characterize the essential properties of a set of executions without considering the notion of time. Admittedly, the word 'essential' here is biased towards process theory, for in this paper the notion of execution of a dynamical system is generalised in such a way that computer-science notions like implementation, refinement, specification, parallel composition and branching bisimulation are still defined in a natural way.

Using category theory as a compass, we start by formally defining our 'object of study' in the next section. We give axioms that characterize the idea of a 'prefix order' on executions, and use this idea as a basis throughout the remainder of the paper. In the subsequent sections, we propose different 'structure

[1] This report is an elaboration of the work presented at the DCM 2013 conference in Buenos Aires [3].

preserving maps' to characterize the different notions from computer science mentioned above. We start by considering refinements, or history preserving maps, and show the rich mathematical structure that arises from them. Then we descent to the substructure of future preserving refinements and show the relation with branching bisimulation equivalence as we know it from concurrency theory.

In this paper, we only develop a very basic theory of dynamics. Admittedly, this may raise more questions than it answers, and many possible continuations for research impose themselves immediately. In the concluding section, we sketch a number of directions for future research in which we expect the proposed generalization to be useful.

2 Prefix Orders

In [9, 5] the notion of branching bisimulation is studied on runs over a transition system, rather than on the transition systems themselves. The authors show that after unfolding a transition system into its set of executions, branching bisimulation can be characterized using forward- and backward- bisimulation relations. It was observed in [5] that the resulting definition of bisimulation only uses the notion of 'prefix' on the runs, rather than requiring a notion of 'silent-steps followed by a single observable step'. This observation becomes important when developing a notion of bisimulation that works for arbitrary types of execution, such as continuous execution, in which a notion of 'next step' does not always exist. Moreover, the subsequent sections show that just capturing the notion of prefix order on executions in an order theoretic fashion already gives us a very flexible general model of dynamics.

In literature, the notion of 'prefix' is often defined using some notion of time. An execution is then defined as a function $e : [0, t] \to X$ from some interval $[0, t]$ over time to a set X, and another execution $f : [0, t'] \to X$ is a *prefix* of e if $t' \leq t$ and for all $\tau \in [0, t']$ it holds that $e(\tau) = f(\tau)$. This notion of prefixing leads to an order relation on executions, which on closer inspection satisfies the following axioms.

Definition 1 (Prefix order) *A* prefix order $\langle \mathbb{U}, \preceq \rangle$ *consists of a* set of executions $\mathbb{U}$ *and a* prefix relation $\preceq \subseteq \mathbb{U} \times \mathbb{U}$ *that is:*

- *reflexive:* $\forall_{a \in \mathbb{U}}\ a \preceq a$*;*
- *transitive:* $\forall_{a,b \in \mathbb{U}}\ a \preceq b \ \wedge\ b \preceq c \ \Rightarrow\ a \preceq c$*;*
- *anti-symmetric:* $\forall_{a,b \in \mathbb{U}}\ a \preceq b \ \wedge\ b \preceq a \ \Rightarrow\ a = b$*;*
- *downward total:* $\forall_{a,b,c \in \mathbb{U}}\ (a \preceq c \ \wedge\ b \preceq c) \ \Rightarrow\ (a \preceq b \ \vee\ b \preceq a)$*;*

In this definition, only downward totality is special; the other three requirements simply say that prefixing is a *partial order*. Downward totality means that, although the future of a system may be branching from a given point of execution, the past is always totally ordered. In [6], this is called the *perfect recall property* of executions: at any point of execution the complete history of the system so far is remembered. Another way of looking at it, is saying that the set of executions behaves like a *tree* structure [7], except that it may be *dense* (in continuous systems there is no 'next' point of execution), there may be no *root* (in some systems history is infinite), and there may be multiple trees next to each other (for example because there are multiple initial states to consider). Typical examples of prefix orders are, of course, total orders like the natural numbers $\langle \mathbb{N}, \leq \rangle$ and the real numbers $\langle \mathbb{R}, \leq \rangle$. But also strings over an alphabet A under their natural prefix order $\langle A^*, \preceq \rangle$, and the aforementioned functions of time.

Two important notions on a prefix order are the *future* and the *history* of an execution.

Definition 2 (History and future) *Given a prefix order $\langle \mathbb{U}, \preceq \rangle$ and an execution $u \in \mathbb{U}$, the* history *and* future *of u are defined by*

- *history:* $u^- \triangleq \{v \in \mathbb{U} \mid v \preceq u\}$;
- *future:* $u^+ \triangleq \{v \in \mathbb{U} \mid u \preceq v\}$.

A map $f : \mathbb{U} \to \mathbb{V}$ between two prefix orders is then

- *order preserving if:* $\forall_{u,u' \in \mathbb{U}}\ u \preceq u' \ \Rightarrow\ f(u) \preceq f(u')$;
- *history preserving if:* $\forall_{u \in \mathbb{U}}\ f(u^-) = f(u)^-$;
- *future preserving if:* $\forall_{u \in \mathbb{U}}\ f(u^+) = f(u)^+$;

with the obvious lifting $f(A) \triangleq \{f(a) \mid a \in A\}$ of f to subsets $A \subseteq \mathbb{U}$.

Incidentally, the well-known idea of 'computation trees' as executions (see e.g. [9, 5]) is obtained by studying only prefix orders in which each history is a finite set, while the idea of 'initial states' at which a system is turned on is captured by studying only prefix orders in which each history has a minimum. Furthermore, one might verify that any history or future preserving function is an order preserving backward- or forward- simulation, respectively, and vice versa.

Lemma 1 (Order preservation) *Every history preserving function $f : \mathbb{U} \to \mathbb{V}$ is order preserving, and so is every future preserving function.*

Proof Let f be history preserving, and let $u, u' \in \mathbb{U}$ such that $u \preceq u'$. Then $u \in (u')^-$, and so $f(u) \in f((u')^-)$. By history preservation $f((u')^-) = f(u')^-$, so $f(u) \in f(u')^-$, from which we conclude $f(u) \preceq f(u')$. The proof for future preserving functions is dual to this. ⊠

Lemma 2 (Simulation) *An order preserving function $f : \mathbb{U} \to \mathbb{V}$ is history preserving if and only if for every $u \in \mathbb{U}$ and $v \in \mathbb{V}$ with $v \preceq f(u)$ there exists a $u' \preceq u$ such that $f(u') = v$. Similarly, it is future preserving if and only if for every $u \in \mathbb{U}$ and $v \in \mathbb{V}$ with $f(u) \preceq v$ there exists a $u \preceq u'$ such that $f(u') = v$.*

Proof Let f be history preserving with $v \preceq f(u)$, then $v \in f(u)^- = f(u^-)$. So there is a $u' \in u^-$ such that $f(u') = v$, and by construction $u' \preceq u$. Reversely, suppose that f is order preserving and that the condition in the lemma holds. Then for any u consider the sets $f(u^-)$ and $f(u)^-$. By order preservation we have $f(u^-) \subseteq f(u)^-$. Furthermore, for any $v \in f(u)^-$ we know $v \preceq f(u)$, and so by assumption there exists a $u' \preceq u$ with $f(u') = v$, and more importantly, $u' \in u^-$ so $v \in f(u^-)$, thus $f(u)^- \subseteq f(u^-)$, and therefore $f(u)^- = f(u)^-$. The proof for future preserving functions is again dual to this. ⊠

3 Refinements as history preserving maps

In this section, we study the category of prefix orders with history preserving maps as morphisms, henceforth referred to as $\mathbf{Pfx}^-$. History preserving maps reflect the property of downward totality, and as such put more emphasis preserving the past than on preserving the future. Even more strongly, the next theorem shows that the history of an execution in fact contains all information about that execution.

Theorem 1 (Histories) *Any prefix order* $\langle \mathbb{U}, \preceq \rangle$ *is isomorphic to its set of histories* $\langle \mathbb{U}^-, \subseteq \rangle$ *under the subset relation, where* $\mathbb{U}^- = \{u^- \mid u \in \mathbb{U}\}$.

Proof It is easy to verify that $\langle \mathbb{U}^-, \subseteq \rangle$ is indeed a prefix order, provided that $\mathbb{U}$ is a prefix order to start with. Furthermore, every element $H \in \mathbb{U}^-$ is a subset of $\mathbb{U}$ and, by construction, contains a maximum $h \in H$ in the ordering on $\mathbb{U}$. It is also easy to verify that taking this maximum gives us a bijection $\max : \mathbb{U}^- \to \mathbb{U}$ that is history preserving, hence by the previous lemma is an isomorphism. ⊠

In itself, this already justifies the study of prefix orders and history preserving maps as a category. But further justification can be found in the observation that a history preserving map $f : \mathbb{U} \to \mathbb{V}$ models how each execution of $\mathbb{U}$ maps to a (more abstract) execution in $\mathbb{V}$. At every point of execution in $\mathbb{U}$, f tells you exactly where the system is in $\mathbb{V}$, thus showing how $\mathbb{U}$ is a refined version of the behavior in $\mathbb{V}$. History preserving maps are suitable to describe *arbitrary implementations* of a given abstract description. As we will see later, surjective history and future preserving maps describe *correct refinements*, in which *all* abstract behavior is reflected in the implementation. In such a setting, the abstract behavior may be considered to be a *specification.* In the category of history preserving maps, notions like parallel composition, disjoint union, prefix closed subsets, arise naturally. The category of surjective history and future preserving maps turns out to be useful for understanding the notion of branching bisimulation equivalence on general dynamical systems.

3.1 Parallel composition

In category theory, a product of two objects is usually an object that represents an arbitrary, or least-assuming, relation between the original two. In the category $\mathbf{Pfx}^-$, the product of two prefix orders $\mathbb{U}$ and $\mathbb{V}$ turns out to be the parallel composition $\mathbb{U} \parallel \mathbb{V}$, consisting of all synchronous and asynchronous interleavings of the executions of $\mathbb{U}$ and $\mathbb{V}$.

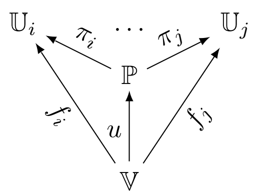

Figure 1: Product

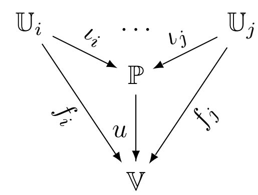

Figure 2: Co-product

Definition 3 (Product) *In category theory, a* product *of a family of objects* $\{\mathbb{U}_i \mid i \in I\}$ *is an object* $\mathbb{P}$ *together with a family of morphisms* $\{\pi_i : \mathbb{P} \to \mathbb{U} \mid i \in I\}$ *(called* projections*), such that for any other object* $\mathbb{S}$ *and family* $\{f_i : \mathbb{S} \to \mathbb{U} \mid i \in I\}$ *there exists a unique morphism* $u : \mathbb{S} \to \mathbb{P}$ *such that* $\pi_i \cdot u = f_i$ *for all* $i \in I$ *(i.e. there is a unique* u *such that the diagram in figure 1 commutes for all* i *and* j*).*

It is well-known from category theory that if a product exists it is unique upto isomorphism, which means that from now on we will often refer to it as *the* product [8].

Definition 4 (Parallel composition) *Given a family of prefix orders* $\{\langle \mathbb{U}_i, \preceq_i \rangle \mid i \in I\}$ *a* joint execution *is a* set *of tuples* $H \subseteq \prod_{i\in I} \mathbb{U}_i$*, modeling a history of concurrent points of execution (synchronous as well as interleaving), such that* $\exists_{h\in H} \forall_{i\in I}\, H_i = h_i^-$ *and* $\forall_{h,g\in H}\, (\forall_{i\in I}\, h_i \preceq_i g_i) \vee (\forall_{i\in I}\, g_i \preceq_i h_i)$*. (Here* $\prod$ *denotes the usual Cartesian product on sets,* h_i *denotes the* i*'th element in a tuple* h*, and* $H_i = \{h_i \mid h \in H\}$ *lifts this to sets of tuples.)*

The parallel composition *of this family, is the set* $\parallel_{i\in I} \mathbb{U}_i$ *of all joint executions, ordered by the relation* $\sqsubseteq$*, defined for all* $G, H \in \parallel_{i\in I} \mathbb{U}_i$ *by* $G \sqsubseteq H \Leftrightarrow G \subseteq H \wedge \forall_{h\in H} \forall_{g\in G} (\forall_{i\in I}\ h_i \preceq_i g_i) \Rightarrow (h \in G)$*. Together with the parallel composition, the family* $\{\pi_i : (\parallel_{j\in I} \mathbb{U}_j) \to \mathbb{U}_i\}$ *of* canonic projections *is given by* $\pi_i(H) = \max(H_i)$ *for every* $i \in I$ *and* $H \in \parallel_{j\in I} \mathbb{U}_j$*.*

Lemma 3 *For all* $G, H \in \parallel_{i\in I} \mathbb{U}_i$ *we find* $G \sqsubseteq H \Leftrightarrow \exists_{h\in H}\ G = \{g \in H \mid \forall_{i\in I}\, g(i) \preceq h(i)\}$

Proof The if-case follows from the maximum of G, using which we obtain an $h \in G$, hence $h \in H$, such that $G(i) = h(i)^-$, so for all $g \in G$ we have

$g(i) \preceq h(i)$. From now on, we denote this element h by $\max H$, since it has the character of a maximum after projection. The only if case is proven easily by taking $h \in H$ and $g \in G$ and observing that $h(i) \preceq g(i)$ for all i implies $h(i) \preceq (\max G)(i)$ hence $h \in G$. ⊠

Theorem 2 *The categorical product of a family of prefix orders is its parallel composition.*

Proof First we easily verify that $\sqsubseteq$ is reflexive, transitive and anti-symmetric. So the parallel composition is prefix ordered provided that $\sqsubseteq$ is also downward total. Take $F, G, H \in \|_{i\in I} \mathbb{U}_i$ and assume $F \sqsubseteq H$ and $G \sqsubseteq H$. If $F \subseteq G$ then it is easy to verify that $F \sqsubseteq G$, and similarly, for $G \subseteq F$ we find $G \sqsubseteq F$. Finally, we obtain a contradiction when we assume $F \not\subseteq G$ and $G \not\subseteq F$. Take $g \in G$ and $f \in F$ such that $g \notin F$ and $f \notin G$. Combining $g \notin F$ and $F \sqsubseteq G$ gives us $g(i) \preceq f(i)$ for all $i \in I$, but combining $f \notin G$ with $G \sqsubseteq F$ gives us $f(i) \preceq g(i)$ for all $i \in I$. From this we conclude $f(i) = g(i)$ using antisymmetry of $\preceq$, and finally the contradiction $f = g$ while $f \notin G$ and $g \in G$.

Next, we verify that the canonic projections π_i are history preserving. To see this, we use the previous lemma to observe that $\pi_i(H^-) = \{(\max G)(i) \mid G \sqsubseteq H\} = \{g(i) \mid g \in H\} = H(i) = \max(H(i))^- = \pi_i(H)^-$.

Finally, we verify that the canonic projections indeed provide a categorical product. For this, consider a family $\{f_i : \mathbb{P} \to \mathbb{U}_i\}$ of history preserving maps. We must now prove that there exists a unique map $u : \mathbb{P} \to (\|_{j\in I} \mathbb{U}_j)$ such that $f_i(p) = \pi_i(u(p))$ for all $i \in I$ and $p \in \mathbb{P}$.

As a candidate, define $u(p) = \{h \in \prod_{i\in I} \mathbb{U}_i \in\mid \exists_{t\preceq p} \forall_{i\in I}\ h(i) = f_i(t)\}$, and use the fact that all f_i are history preserving to verify that this function returns a joint execution for each $p \in \mathbb{P}$ and is indeed history preserving itself. By construction $f_i(p) = \pi_i(u(p))$.

To see that the choice of u is unique, consider any function $v : \mathbb{P} \to (\|_{j\in I} \mathbb{U}_j)$ satisfying $f_i(p) = \pi_i(v(p)) = \pi_i(u(p))$ for all p. Pick any $p \in \mathbb{P}$, pick $h \in v(p)$ and consider the prefix $h^- \cap v(p) \sqsubseteq v(p)$. By history preservation, there exists a $t \preceq p$ such that $v(t) = h^- \cap v(p)$. Furthermore, this prefix has h as maximum, so $\pi_i(v(t)) = h(i) = f_i(t)$ for all $i \in I$. By construction of u we find $h \in u(p)$, hence $v(p) \subseteq u(p)$. Reversely, pick $h \in u(p)$ and consider the prefix $h^- \cap u(p) \sqsubseteq u(p)$. By history preservation, we find $t \preceq p$ such that $u(t) = h^- \cap u(p)$, so $\pi_i(u(t)) = h(i) = f_i(t)$ for all $i \in I$. Hence $\pi_i(v(t)) = h(i)$ for all $i \in I$, hence $h \in v(t)$. Since $t \preceq p$, we find by order preservation $v(t) \sqsubseteq v(p)$, which implies $v(t) \subseteq v(p)$. We conclude $h \in v(p)$ and finally, $u(p) \subseteq v(p)$. Hence $u(p) = v(p)$ for all $p \in \mathbb{P}$. ⊠

The reader should be aware that, even though the parallel composition of prefix orders gives us the expected synchronous and asynchronous executions of computational and other familiar dynamical systems, there are pathological prefix orders of which the product is rather surprising. As an example, consider the set $-\Omega$ of ordinal numbers in reversed order, upto (but not including) the first uncountable ordinal. Furthermore, consider the set $-\mathbb{N}$ of natural numbers in reversed order. Obviously, both $-\Omega$ and $-\mathbb{N}$ do not have a minimum. Furthermore, it is well known from topology that any Ω is a (countably) compact space (see e.g. [12, p.69 ex43.8]), so any sequence in $-\Omega$ has a limit in $-\Omega$, which means that for any order preserving function $f : -\mathbb{N} \to -\Omega$ there is a $o \in -\Omega$ with $o \preceq f(n)$ for all $n \in \mathbb{N}$. From this we conclude that there is no $n \in \mathbb{N}$ with $f(n) \preceq o - 1$. As a result, $-\Omega$ and $-\mathbb{N}$ cannot have a joint execution, so, rather surprisingly, their parallel composition returns the empty set: $-\Omega \parallel -\mathbb{N} = \emptyset$. How to avoid these examples is a topic of ongoing research. The answer might be sought in fixing a common generating total order (a time-base so-to-speak) before composing two prefix orders.

3.2 Alternative composition

Dual to the product, a co-product of two objects is usually an object that represents an arbitrary union of the original two. It is most restricting, in the sense that it assumes the objects to have no interaction with each other whatsoever. In the category $\mathbf{Pfx}^-$, the co-product of two prefix orders $\mathbb{U}$ and $\mathbb{V}$ is called the alternative composition, and is simply obtained by taking the disjoint union $\mathbb{U} \uplus \mathbb{V}$ of all executions.

Definition 5 (Co-product) *In category theory, a* co-product *of a family of objects $\{\mathbb{U}_i \mid i \in I\}$ is an object $\mathbb{P}$ together with a family of morphisms $\{\iota_i : \mathbb{U} \to \mathbb{P} \mid i \in I\}$ (called* insertions*), such that for any other object $\mathbb{S}$ and family $\{f_i : \mathbb{U} \to \mathbb{S} \mid i \in I\}$ there exists a unique morphism $u : \mathbb{P} \to \mathbb{S}$ such that $u \cdot \iota_i = f_i$ for all $i \in I$ (i.e. there is a unique u such that the diagram in figure 2 commutes for all i and j).*

Definition 6 (Alternative composition) *Given a family $\{\langle \mathbb{U}_i, \preceq_i \rangle \mid i \in I\}$ of prefix orders, the* alternative composition *or* disjoint union *is the disjoint union on sets: $\biguplus_{i\in I} \mathbb{U}_i = \{(i,u) \mid i \in I \ \wedge \ u \in \mathbb{U}_i\}$. This set is ordered by the relation $\sqsubseteq$, defined by $(i,u) \sqsubseteq (j,v) \ \Leftrightarrow \ i = j \ \wedge \ u \preceq v$, and equipped with a family $\{\iota_i : \mathbb{U}_i \to \biguplus_{j\in I} \mathbb{U}_j\}$ of* canonic insertions *given by $\iota_i(u) = (i,u)$ for all $i \in I$ and $u \in \mathbb{U}_i$.*

Theorem 3 *The categorical co-product of a family of prefix orders is its alternative composition.*

Proof To verify that $\biguplus_{i\in I} \mathbb{U}_i$ is prefix ordered and that its canonic insertions are history preserving is trivial. Furthermore, if we consider a family $\{f_i : \mathbb{U}_i \to \mathbb{P}\}$ of history preserving maps, it is straightforward to verify that $u(i,x_i) = f_i(x_i)$ defines a unique history preserving map $u : \biguplus_{i\in I} \mathbb{U}_i \to \mathbb{P}$ such that $u(\iota_i(x_i)) = f_i(x_i)$ for all $i \in I$ and $x_i \in \mathbb{U}_i$. Therefore, the disjoint union is a categorical co-product. ⊠

3.3 Subobjects

In category theory, the notion of *monomorphism* is often used to model that one object is a subobject of another. In the category **Pfx**$^-$ subobjects turn out to be isomorphic to the prefix-closed subsets of a prefix order. We show this by proving that every prefix-closed subset has a natural monomorphism associated with it, and that every source of a monomorphism is isomorphic to a prefix-closed subset.

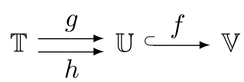

Figure 3: Monomorphism

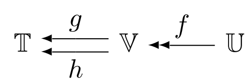

Figure 4: Epimorphism

Definition 7 (Monomorphism) *In category theory, a morphism* $f : \mathbb{U} \to \mathbb{V}$ *is called a* monomorphism*, denoted with a hooked arrow, and is a witness for* $\mathbb{U}$ *being a subobject of* $\mathbb{V}$*, if for every two morphisms* $g, h : \mathbb{T} \to \mathbb{U}$ *in that category we have* $f \cdot g = f \cdot h \;\Rightarrow\; g = h$*. I.e. the diagram in figure 3 only commutes when* $g = h$*.*

Definition 8 (Prefix-closed subset) *Given a prefix order* $\{\langle \mathbb{U}, \preceq\}$ *a subset* $X \subseteq \mathbb{U}$ *is* prefix-closed *if for all* $x \in X$ *we find* $x^- \subseteq X$*.*

Theorem 4 *A history preserving map is a monomorphism if and only if it is an injection.*

Proof From set theory we know that any injective function is a monomorphism, and that monomorphisms carry over to subcategories. So any injective history preserving map is a monomorphism.

For the reverse direction, assume that f is a monomorphism, and pick $x, y \in \mathbb{X}$ such that $f(x) = f(y)$. Firstly, if $x \preceq y$, then take $\mathbb{Z} = y^-$, take $h(t) = t$ for all $t \in \mathbb{T}$, and take $g(t) = t$ for $t \preceq x$ while $g(t) = x$ for $x \preceq t$. By construction $f \circ h = f \circ g$, so by monicity $h = g$, and therefore $x = y$. From this we conclude that if $f(x) = f(y)$, then x and y are not related by $\preceq$. Secondly, now that we have established that any u and v for which $f(u) = f(v)$ are unrelated, we construct the set $\mathbb{T} = \{(u, v) \mid u \leq x \;\wedge\; v \leq y \;\wedge\; f(u) = f(v)\}$, order it by $(u, v) \sqsubseteq (u', v') \;\Leftrightarrow\; u \leq u' \;\wedge\; v \leq v'$, and construct two functions $h, g : \mathbb{T} \to \mathbb{U}$ defined as $h(u, v) = u$ and $g(u, v) = v$. We verify that $\langle \mathbb{T}, \sqsubseteq \rangle$ is indeed a prefix order as follows. That it is a partial order is trivial, but that it is downward total is not. For this, consider $(u, v) \sqsubseteq (u'', v'')$ and $(u', v') \sqsubseteq (u'', v'')$ while (u, v) and (u', v') are unrelated. This leaves us with two possibilities, namely $u \preceq u'$ and $v' \preceq v$, or $u' \preceq u$ and $v \preceq v'$. But we also know that $f(u) = f(v)$ and $f(u') = f(v')$, so by order preservation either $f(u) \preceq f(u') = f(v') \preceq f(v) = f(u)$ or $f(u') \preceq f(u) = f(v) \preceq f(v') = f(u')$

from which we conclude $f(u) = f(u')$ and similarly $f(v) = f(v')$. We use the first observation to conclude $u = u'$ and $v = v'$, thus $(u, v) = (u', v')$, contradicting the assumption that (u, v) and (u', v') were unrelated. Finally, by construction the functions g and h are history preserving and satisfy $g \circ f = h \circ f$, so $g = h$ and in particular $x = h(x, y) = g(x, y) = y$, from which we conclude that f is an injection. ⊠

Accidentally, the reader may also verify that (in contrast to the category of partial orders) monomorphisms are exactly the embeddings, i.e. those functions f satisfying $x \preceq y \quad \Leftrightarrow \quad f(x) \preceq f(y)$ for all x and y. Next, we show that subobjects are exactly the prefix-closed subsets.

Theorem 5 *Every morphism $f : \mathbb{U} \to \mathbb{V}$ selects a prefix-closed subset $f(\mathbb{U}) = \{f(u) \mid u \in \mathbb{U}\}$.*

Proof This follows directly from lemma 2. ⊠

Theorem 6 *Given a prefix-closed subset, $\mathbb{U} \subseteq \mathbb{V}$, the natural injection $\iota : \mathbb{U} \to \mathbb{V}$ defined by $\iota(u) = u$ is a monomorphism.*

Proof Trivial. ⊠

Theorem 7 *Given monomorphisms $f : \mathbb{U} \to \mathbb{V}$ and $g : \mathbb{W} \to \mathbb{V}$ such that $f(\mathbb{U}) = g(\mathbb{W})$, there exists an isomorphism $h : \mathbb{U} \to \mathbb{W}$ such that $g \circ h = f$.*

Proof As g and f are injections and $f(\mathbb{U}) = g(\mathbb{W})$, we find for every $u \in \mathbb{U}$ a unique $w \in \mathbb{W}$ such $f(u) = g(w)$. Picking $h(u) = w$ it is straightforward to verify that this function is a bijection by construction, and history preserving because f and g are history preserving, hence it is an isomorphism. ⊠

3.4 Quotients

The dual of a monomorphism is an *epimorphism*. Epimorphisms are often used to model that one object is a quotient of another. In the category $\mathbf{Pfx}^-$ quotients turn out to be isomorphic to the equivalence classes of a particular kind of backward bisimulation relations, which we will call *order contracting backward bisimulation equivalences.*

Definition 9 (Epimorphism) *In category theory, a morphism $f : \mathbb{U} \to \mathbb{V}$ is called an* epimorphism*, denoted with a double arrow, and is a witness for $\mathbb{V}$ being a quotient of $\mathbb{U}$, if for every two morphisms $g, h : \mathbb{V} \to \mathbb{T}$ in that category we find $g \cdot f = h \cdot f \quad \Rightarrow \quad g = h$. I.e. the diagram in figure 4 only commutes when $g = h$.*

Definition 10 (Order contracting backward bisimulation equivalence) *Given a prefix order* $\mathbb{U}$*, a relation* $\sim\subseteq \mathbb{U}\times\mathbb{U}$ *is a* backward bisimulation *if:*

- $\forall_{u,v,u'\in\mathbb{U}}\, u\sim v \;\wedge\; u'\preceq u \;\Rightarrow\; \exists_{v'\in\mathbb{U}}\; v'\preceq v \;\wedge\; u'\sim v'$*; and*
- $\forall_{u,v,v'\in\mathbb{U}}\, u\sim v \;\wedge\; v'\preceq v \;\Rightarrow\; \exists_{u'\in\mathbb{U}}\; u'\preceq u \;\wedge\; u'\sim v'$.

Furthermore, a relation is an equivalence *if:*

- $\forall_{u\in\mathbb{U}}\; u\sim u$*; and*
- $\forall_{u,v\in\mathbb{U}}\; u\sim v\Rightarrow v\sim u$*; and*
- $\forall_{u,v,w\in\mathbb{U}}\; u\sim v \;\wedge\; v\sim w\Rightarrow u\sim w$.

Given an equivalence $\sim$*, we write* $[u]_\sim=\{v\in\mathbb{U}\mid u\sim v\}$ *for the* equivalence class *of* $u\in\mathbb{U}$.
An equivalence relation is order contracting *if*

- $\forall_{u,v,w,x\in\mathbb{U}}\; u\preceq v \;\wedge\; v\sim w \;\wedge\; w\preceq x \;\wedge\; x\sim u \;\Rightarrow u\sim v$.

Theorem 8 *A history preserving map is an epimorphism if and only if it is a surjection.*

Proof Assume that f is a surjective morphism and take $g,h:\mathbb{V}\to\mathbb{T}$ with $g\neq h$, then there exists a $v\in\mathbb{V}$ with $g(v)\neq h(v)$. Surjection gives us a $u\in\mathbb{U}$ with $f(u)=v$ so $g(f(u))\neq h(f(u))$, from which we conclude that f is an epimorphism.
Reversely, assume that f is an epimorphism but not surjective. We show this leads to a contradiction by picking any $v\in\mathbb{V}$ for which there is no $u\in\mathbb{U}$ with $f(u)=v$. Observe that lemma 2 now ensures that there is also no $u\in\mathbb{U}$ with $f(u)\in v^+$, since every history of v^+ goes through v. (This is a crucial observation. In comparison, in the category $\mathbf{Pfx}^+$ surjection is *not* a necessary condition for an epimorphism and also in the category of partial orders it is not.) If v is minimal we construct $g,h:\mathbb{V}\to\{a,b\}$, with a and b unordered, by defining $g(w)=a$ for all $w\in\mathbb{V}$ while $h(w)=a$ for any $w\neq v$ and $h(v)=b$. It is easily verified that g and h are history preserving, and by construction $g\cdot f=g\cdot h$, which contradicts the assumption that f is an epimorphism. Similarly, if v is not minimal we construct $g,h:\mathbb{V}\to\{0,1\}$ by defining $g(w)=0$ for all $w\in\mathbb{V}$ while $h(w)=0$ for any $w\notin v^+$ and $h(w)=1$ for $w\in v^+$. Again it is easily verified that g is history preserving. To see that h is history preserving we observe that $h(w^-)=\{0\}=0^-$ for any $w\notin v^+$. Furthermore, if $w\in v^+$ then we use the fact that v is not minimal to find a $w'\prec v$ with (by antisymmetry) $w'\notin v^+$ to conclude $h(w^-)=\{0,1\}=1^-=h(w)^-$. By construction we get $g\cdot f=g\cdot h$, which once more contradicts the assumption that f is an epimorphism. ⊠

Theorem 9 *Every morphism* $f:\mathbb{U}\to\mathbb{V}$ *defines an order contracting backward bisimulation equivalence on* $\mathbb{U}$ *given by* $u\sim_f u' \;\Leftrightarrow\; f(u)=f(u')$.

Proof It is easy to see that $\sim_f$ is an equivalence, and that it is a backward bisimulation follows straightforwardly from lemma 2. Finally, assume $u \sim_f u' \preceq u'' \sim_f u''' \preceq u$, then by order preservation $f(u) = f(u') \preceq f(u'') = f(u''') \preceq f(u)$, and by anti-symmetry, $f(u) = f(u'')$, so $u \sim_f u''$. Therefore $\sim_f$ is order contracting. ⊠

Theorem 10 *Given an order contracting backward bisimulation equivalence $\sim$ on $\mathbb{U}$, the projection $[\_]_\sim : \mathbb{U} \to \mathbb{U}/\sim$ of executions onto their equivalence classes is an epimorphism.*

Proof Let $\mathbb{U}/\sim$ denote the set of equivalence classes of $\sim$, and let $\sqsubseteq$ denote the ordering on equivalence classes defined by $[u]_\sim \sqsubseteq [u']_\sim$ iff $\exists_{w,w' \in \mathbb{U}} u \sim w \preceq w' \sim u'$. We will prove that $\sqsubseteq$ is a prefix order, and that the projection $[\_]$ is an epimorphism.

Firstly, observe that $u \sim u \preceq u \sim u$, so $[u]_\sim \sqsubseteq [u]_\sim$, so $\sqsubseteq$ is reflexive. Secondly, assume $[u]_\sim \sqsubseteq [u']_\sim$ and $[u']_\sim \sqsubseteq [u'']_\sim$, then there exist w, w', w'', w''' such that $u \sim w \preceq w' \sim u' \sim w'' \preceq w''' \sim u''$. By transitivity and backward bisimulation of $\sim$ we find a v' such that $u \sim v'$ and $v' \preceq w'' \preceq w'''$, so $u \sim v' \preceq w''' \sim u''$, from which we conclude that $[u]_\sim \sqsubseteq [u'']_\sim$, hence $\sqsubseteq$ is transitive. Thirdly, assume $[u]_\sim \sqsubseteq [u']_\sim$ and $[u']_\sim \sqsubseteq [u]_\sim$, then there exist w, w', w'', w''' such that $u \sim w \preceq w' \sim u' \sim w'' \preceq w''' \sim u$. By transitivity and backward bisimulation we find a v' and v'' such that $u \sim v' \preceq u' \sim v'' \preceq u$, and by order contraction we find $u \sim u'$, hence $[u]_\sim = [u']_\sim$, hence $\sqsubseteq$ is anti-symmetric. Fourthly, assume $[u]_\sim \sqsubseteq [u']_\sim$ and $[u'']_\sim \sqsubseteq [u]_\sim$, then there exist w, w', w'', w''' such that $u \sim w \preceq w' \sim u'$ and $u'' \sim w'' \preceq w''' \sim u'$. Using backward bisimulation we find v' and v'' such that $u \sim v' \preceq u'$ and $u'' \sim v'' \preceq u'$. Using downward totality we derive $v' \preceq v''$ or $v'' \preceq v'$. Hence $u \sim v' \preceq v'' \sim u''$ or $u'' \sim v'' \preceq v' \sim u$, from which we conclude $[u]_\sim \sqsubseteq [u'']_\sim$ or $[u'']_\sim \sqsubseteq [u]_\sim$, from which we conclude that $\sqsubseteq$ is downward total.

Finally, the projection $[\_]$ is surjective by construction. Also by construction, if $u \preceq u'$ then $[u]_\sim \sqsubseteq [u']_\sim$, so it is order preserving. Finally, if $[u]_\sim \sqsubseteq [u']_\sim$ then there exists w, w' such that $u \sim w \preceq w' \sim u'$, and by backward bisimulation of $\sim$ we find an element v such that $u \sim v \preceq u'$, so there exists a $v \preceq u'$ such that $[v]_\sim = [u]_\sim$, and by lemma 2, the projection is history preserving. From this we conclude that it is an epimorphism. ⊠

Theorem 11 *Given epimorphisms $f : \mathbb{U} \to \mathbb{V}$ and $g : \mathbb{U} \to \mathbb{W}$ such that $\sim_f = \sim_g$, there exists an isomorphism $h : \mathbb{V} \to \mathbb{W}$ such that $h \circ f = g$.*

Proof Consider the epimorphism $f : \mathbb{U} \to \mathbb{V}$, and let $\mathbb{U}/\sim_f$ denote the set of equivalence classes of $\sim_f$. Next, verify that $\mathbb{U}/\sim_f$ is prefix ordered if we define $[u]_{\sim_f} \sqsubseteq [u']_{\sim_f}$ iff $\exists_{w,w' \in \mathbb{U}} u \sim_f w \preceq w' \sim_f u'$. Furthermore, since the elements of each equivalence class coincide on f, we can lift this map to $f' : \mathbb{U}/\sim_f \to \mathbb{V}$ by stating $f'([u]) = f(u)$. This map is an injection by construction and a history preserving surjection because f is an epimorphism. In

Pfx$^-$ every bijective morphism is an isomorphism, and so is f. Finally, since $\sim_f = \sim_g$, combining the isomorphisms $f' : \mathbb{U}/\sim_f \to \mathbb{V}$ and $g' : \mathbb{U}/\sim_g \to \mathbb{W}$ gives us an isomorphism $g' \circ (f')^{-1} : \mathbb{V} \to \mathbb{W}$ which by construction satisfies: $g' \circ (f')^{-1} \circ f = g' \circ (f')^{-1} \circ f' \circ [\_] = g' \circ [\_] = g$. ⊠

3.5 Small limits and co-limits

Given that we no know how to construct arbitrary products, co-products, subobjects and quotients, we can continue to construct any limit or co-limit of a commuting diagram as the subobject of a parallel composition, or the equivalence classes of an alternative composition, respectively. This very general construction will be of interest to those familiar with category theory, but not so much to those interested in dynamical systems. Therefore, we just pose the claim here and leave the details to the interested reader. Note that further-on we use the limit construction to create push-outs and pull-backs when we discuss the notion of branching bisimulation between prefix orders.

Definition 11 (Diagram) *Given a category $\mathcal{D}$ with a* set *of objects (i.e. a so-called* small *category), a* diagram in **Pfx**$^-$ *is a functor $F : \mathcal{D} \to$* **Pfx**$^-$ *that maps the objects of $\mathcal{D}$ to prefix orders and the morphisms of $\mathcal{D}$ to history preserving maps, such that $F(f \circ g) = F(f) \circ F(g)$ for all morphisms in $\mathcal{D}$.*

Definition 12 (Diagram limits) *Given a diagram $F : \mathcal{D} \to$* **Pfx**$^-$ *its limit $\varinjlim F$ is a prefix order with morphisms $\pi_d : \varinjlim F \to F(d)$ for each object $d \in \mathcal{D}$, such that $F(f) \circ \pi_d = \pi_e$ for any morphism $f : d \to e$ in $\mathcal{D}$, and furthermore, for any prefix order $\mathbb{V}$ with morphisms π'_d that satisfy this condition there is a unique $u : \mathbb{V} \to \mathbb{U}$ such that $\pi_d \circ u = \pi'_d$ for all $d \in \mathcal{D}$.*

Theorem 12 (Small limit theorem) *Given a diagram $F : \mathcal{D} \to$* **Pfx**$^-$ *its limit is given by the largest prefix-closed subset $\mathbb{L} \subseteq \|_{d \in \mathcal{D}} F(d)$ such that $H \in \mathbb{L} \Leftrightarrow \forall_{(f:d \to e) \in \mathcal{D}} \forall_{G \sqsubseteq H} F(f)(\pi_{F(d)}(G)) = \pi_{F(e)}(G)$.*

Definition 13 (Diagram co-limits) *Given a diagram $F : \mathcal{D} \to$* **Pfx**$^-$ *its co-limit $\varprojlim F$ is a prefix order with morphisms $\iota_d : F(d) \to \varprojlim F$ for each object $d \in \mathcal{D}$, such that $\iota_e \circ F(f) = \iota_d$ for any morphism $f : d \to e$ in $\mathcal{D}$, and furthermore, for any prefix order $\mathbb{V}$ with morphisms ι'_d that satisfy this condition there is a unique $u : \mathbb{U} \to \mathbb{V}$ such that $\iota'_d = u \cdot \iota_d$ for all $d \in \mathcal{D}$.*

Theorem 13 (Small co-limit theorem) *Given a diagram $F : \mathcal{D} \to$* **Pfx**$^-$ *its co-limit is given by the smallest order contracting equivalence $\sim$ on $\biguplus_{d \in \mathcal{D}} F(d)$ such that $\forall_{(f:d \to e) \in \mathcal{D}} \forall_{x \in F(d)} \iota_{F(e)}(F(f)(x)) \sim \iota_{F(d)}(x)$.*

4 Bisimulation through history and future preserving surjections

In this section, the previously mentioned result of [9, 5], capturing branching bisimulation using futures and histories, is generalized to prefix orders. However, in contrast to [9, 5], we do not use a relational definition of branching bisimulation here, but a more categorical definition using spans (as proposed by [13]) and an alternative one using co-spans.

We first establish the fact that both these alternative definitions capture the original notion of bisimulation on labeled transition systems. Next, we will show that the definition using spans does in general *not* lead to an equivalence on prefix orders, while the definition using co-spans does. Furthermore, the definition using co-spans turns out to be a congruence for the parallel composition and alternative composition.

4.1 Bisimulation on labeled transition systems

Definition 14 (Labeled transition system) *A* labeled transition system *is a tuple $\langle X, A, i, \to\rangle$, consisting of a set of states X, a set of observables A, an initial state $i \in A$, and a transition relation $\to\subseteq X \times (A \cup \{\tau\}) \times X$ with the unobservable $\tau \notin A$. Given $a \in A \cup \{\tau\}$ I write $x \stackrel{a}{\to} x'$ for $(x, a, x') \in\to$ and $x_0 \stackrel{a}{\twoheadrightarrow} x_{n+1}$ whenever there exists a sequence $x_0 \dots x_{n+1}$ such that $x_i \stackrel{\tau}{\to} x_{i+1}$ for every $i < n$ and $x_n \stackrel{a}{\to} x_{n+1}$.*

Definition 15 (Run) *A* run *over a labeled transition system $\langle X, A, i, \to\rangle$ is a sequence $\rho \in ((A \cup \{\tau\}) \times X)^*$ such that, if ρ is not empty, it holds that $i \stackrel{\rho_1(0)}{\to} \rho_2(0)$ and $\rho_2(n) \stackrel{\rho_1(n+1)}{\to} \rho_2(n+1)$ for all $n+1 \in dom(\rho)$. The set of all runs is denoted $\mathcal{R}(\to)$, is prefix ordered in the usual way, and is observed by a function $\pi_{A^*} : \mathcal{R}(\to) \to A^*$ defined recursively as $\pi_{A^*}(\epsilon) = \epsilon$, $\pi_{A^*}(\rho \cdot \tau) = \pi_{A^*}(\rho)$, and $\pi_{A^*}(\rho \cdot a) = \pi_{A^*}(\rho) \cdot a$, for $a \in A$.*

Definition 16 (Branching Bisimulation) *Two labeled transition systems $\langle X, A, i, \to_1\rangle$ and $\langle Y, A, j, \to_2\rangle$ are* branching bisimilar *if there exists a relation $\mathcal{R} \subseteq X \times Y$ such that $i \,\mathcal{R}\, j$ and*

- *if $x\mathcal{R}y$, and $x \stackrel{a}{\to}_1 x'$, then either $a = \tau$ and $x'\mathcal{R}y$, or there exist y', y'' such that $y \stackrel{\tau}{\twoheadrightarrow}_2 y'$ and $y' \stackrel{a}{\to}_2 y''$ and $x\mathcal{R}y'$ and $x'\mathcal{R}y''$;*
- *if $x\mathcal{R}y$, and $y \stackrel{a}{\to}_2 y'$, then either $a = \tau$ and $x\mathcal{R}y'$, or there exist x', x'' such that $x \stackrel{\tau}{\twoheadrightarrow}_1 x'$ and $x' \stackrel{a}{\to}_1 x''$ and $x'\mathcal{R}y$ and $x''\mathcal{R}y'$.*

Theorem 14 *Two labeled transition systems $\langle X, A, i, \to_1\rangle$ and $\langle Y, A, j, \to_2\rangle$ are branching bisimilar if and only if there exists a prefix order $\langle \mathbb{U}, \preceq\rangle$ and a* span *(f, g) of history and future preserving (surjective) maps $f : \mathbb{U} \to Runs(\to_1)$ and $g : \mathbb{U} \to Runs(\to_2)$ such that $\pi_{A^*}(g(u)) = \pi_{A^*}(f(u))$ for every $u \in \mathbb{U}$.*

Proof To check one direction, assume a span of future and history preserving surjections $f : \mathbb{U} \to \text{Runs}(\to_1)$ and $g : \mathbb{U} \to \text{Runs}(\to_2)$ such that $\pi_{A^*}(g(u)) = \pi_{A^*}(f(u))$ for every $u \in \mathbb{U}$. Then create the relation

$$x\mathcal{R}y \text{ iff } \exists_{u\in\mathbb{U}}\ \pi_X(u) = x \ \wedge\ \pi_Y(u) = y,$$

where $\pi_X(u)$ denotes the mapping of the joint execution u to the current state in $\langle X, A, i, \rightarrow_1\rangle$, and similarly for $\pi_Y(u)$.

We now continue to prove that this is in fact a branching bisimulation relation between $\langle X, A, i, \rightarrow_1\rangle$ and $\langle Y, A, j, \rightarrow_2\rangle$. To see this, note that history preservation forces the minimum of $\mathbb{U}$ to map onto the empty strings over X and Y, respectively, thus witnessing $i\mathcal{R}j$.

Next, consider $x, x' \in X$ and $y \in Y$ such that $x \stackrel{a}{\rightarrow} x'$ and $x\mathcal{R}y$. By construction, there exists a $u \in \mathbb{U}$ such that $\pi_X(u) = x$ and $\pi_Y(u) = y$, so from u we can derive a run $f(u)$ over X leading to x, and we can lengthen this execution with an a-step to x'. Future preservation of f then gives us a $u \preceq u'$ such that $f(u')$ ends in x' and $\pi_{A^*}(u') = \pi_{A^*}(u) \cdot a$ in case $a \neq \tau$ and $\pi_{A^*}(u') = \pi_{A^*}(u)$ otherwise. Furthermore observe that there are no points in between $f(u)$ and $f(u')$. From this, order preservation of g gives us a run $g(u)$ upto y and a run $g(u')$ to some further point y''' at which a has been observed. Thus we know there exist y' and y'' such that $y \stackrel{\tau}{\twoheadrightarrow}_2 y'$, $y' \stackrel{a}{\rightarrow}_2 y''$ and $y'' \stackrel{\tau}{\twoheadrightarrow}_2 y'''$ (possibly $y' = y''$ if $a = \tau$). By history preservation of g we can then find $u_{y'}$ and $u_{y''}$ such that $g(u_{y'})$ gives the run upto y' and $g(u_{y''})$ gives the run upto y'', and $u \preceq u_{y'} \preceq u_{y''} \preceq u'$. By order preservation, we find $f(u) \preceq f(u_{y'}) \preceq f(u_{y''}) \preceq f(u')$, and because observations π_{A^*} are preserved and there are no points between $f(u)$ and $f(u')$ we find $f(u) = f(u_{y'})$ and $f(u_{y''}) = f(u')$ hence $x\mathcal{R}y'$ and $x'\mathcal{R}y''$. The proof is symmetric when starting from a transition $y \stackrel{a}{\rightarrow} y'$.

To check the other direction, assume that there exists a bisimulation relation $\mathcal{R} \subseteq X \times Y$ between $\langle X, A, i, \rightarrow_1\rangle$ and $\langle Y, A, j, \rightarrow_2\rangle$. We can now use the categorical product discussed further-on in this paper to create the parallel composition of runs $\mathrm{Runs}(\rightarrow_1) \parallel \mathrm{Runs}(\rightarrow_2)$, and from it select the subobject that agrees on the bisimulation relation: $\mathbb{U} = \{H \in \mathrm{Runs}(\rightarrow_1) \parallel \mathrm{Runs}(\rightarrow_2) \mid \forall_{H' \sqsubseteq H}\, \pi_X(H')\, \mathcal{R}\, \pi_Y(H')\}$.

This set is by construction prefix-closed, and using the canonical projections of the product, it maps to $\mathrm{Runs}(\rightarrow_1)$ and $\mathrm{Runs}(\rightarrow_2)$ in the obvious history preserving way.

In general these maps may not be surjective. However, in the particular case of runs products can be proven to have surjective and future preserving canonical projections. We leave this to the reader to verify. Given that the projections of the product are surjective and future preserving, it is straightforward to use the definition of branching bisimulation to show that the projections from the subobject $\mathbb{U}$ satisfy the conditions of lemma 2, and hence are future preserving as well. Finally, because all initial states are mapped to, future preservation implies surjection for those maps as well. ⊠

In the previous section, we argued that history preserving maps model refinements of a specification. As a consequence, the above theorem may be interpreted as: *two specifications are branching bisimilar if and only if they have a common refinement*. In other words, branching bisimulation is a way to define that two specifications are 'consistent' with each other. But there is a second alternative to defining bisimulation, which is by saying that *two implementations are branching bisimilar if and only if they have a common specification.* That coincides with the following theorem.

Theorem 15 *Two labeled transition systems $\langle X, A, i, \rightarrow_1\rangle$ and $\langle Y, A, j, \rightarrow_2\rangle$ are branching bisimilar if and only if there exists a prefix order $\langle \mathbb{U}, \preceq\rangle$, a* co-span *$(f, g)$ of history and future preserving (surjective) maps $f : Runs(\rightarrow_1) \rightarrow \mathbb{U}$ and $g : Runs(\rightarrow_2) \rightarrow \mathbb{U}$, and a function $\pi_{A^*} : \mathbb{U} \rightarrow A^*$, such that $\pi_{A^*}(x) = \pi_{A^*}(f(x))$ and $\pi_{A^*}(y) = \pi_{A^*}(g(y))$, for all $x \in X$ and $y \in Y$.*

Proof To check one direction, assume a co-span of future and history preserving surjections $f : \text{Runs}(\rightarrow_1) \rightarrow \mathbb{U}$ and $g : \text{Runs}(\rightarrow_2) \rightarrow \mathbb{U}$ as in the theorem and create the relation:
$x\mathcal{R}y$ iff $\exists_{r\in\text{Runs}(\rightarrow_1), r'\in\text{Runs}(\rightarrow_2)}\ \pi_X(r) = x\ \wedge\ \pi_Y(r') = y\ \wedge\ f(r) = g(r')$.

We now continue to prove that this is in fact a branching bisimulation relation between $\langle X, A, i, \rightarrow_1\rangle$ and $\langle Y, A, j, \rightarrow_2\rangle$. For this, consider the empty run ϵ over X. We have $\pi_X(\epsilon) = i$, and we know that $f(\epsilon)$ is a minimal element of $\mathbb{U}$. Furthermore, since g is surjective, there is a run over Y that maps to this minimal element, and by history preservation, this run is minimal, hence it is the empty run. So $f(\epsilon) = g(\epsilon)$ and $\pi_X = i$ and $\pi_Y = j$, hence $i\mathcal{R}j$.

Next, consider $x, x' \in X$ and $y \in Y$ such that $x \xrightarrow{a} x'$ and $x\mathcal{R}y$. By construction of $\mathcal{R}$ we find that x is reachable by some run r and y is reachable by some run r' such that $f(r) = g(r')$. Furthermore, we can extend r with an a-step to x', thus forming a run r_0 with $r \prec r_0$ and $\pi_{A^*}(r_0) = \pi_{A^*}(r)\cdot a$. Observe that there are no points between r and r_0, and by order preservation of f we find $g(r') = f(r) \preceq f(r_0)$. Furthermore, by history preservation we find that there are no points between $f(r)$ and $f(r_0)$ either (since there are no points between r and r_0 that could map to these). Then, by future preservation of g there exists a run r_1 that extends r' and observes a at a state y'''. Deconstructing this run gives us y', y'' with $y \overset{\tau}{\twoheadrightarrow}_2 y'$, $y' \xrightarrow{a}_2 y''$ and $y'' \overset{\tau}{\twoheadrightarrow}_2 y'''$ (possibly $y' = y''$ if $a = \tau$). Let $r_{y'}$, $r_{y''}$ be the runs upto these intermediate points respectively, then order preservation of g, the fact that there are no points between $g(r)$ and $g(r_1)$, and preservation of the observation gives us $g(r) = g(r_{y'})$ and $g(r_{y''}) = g(r_1)$, and so $x\mathcal{R}y'$ and $x'\mathcal{R}y''$. The proof when starting from a transition $y \xrightarrow{a} y'$ is symmetric to this.

To check the other direction, assume that there exists a bisimulation relation $\mathcal{R} \subseteq X \times Y$ between $\langle X, A, i, \rightarrow_1\rangle$ and $\langle Y, A, j, \rightarrow_2\rangle$. We now just construct the quotient transition system $\langle \mathcal{R}, A, (i, j), \rightarrow_{1,2}\rangle$. The runs of the original transition systems map to the runs of this quotient system in a natural way, which upon closer inspection (using the properties of branching bisimulation relations) turn out to be history and future preserving surjections. ⊠

4.2 Bisimulation through spans versus co-spans

Comparing the proof-sketches of the two theorems in the previous section, one may notice that the first relies on the notion of categorical product, while the second one does not. Furthermore, we have previously seen that some products of general prefix orders behave unexpectedly. Indeed, while the idea of using a span or co-span of maps to define bisimulation coincides for labeled transition systems, there are differences when we apply them to prefix orders in general.

Next, we show that on prefix orders the definition using co-spans is to be preferred over the definition using spans, mainly because the latter does not yield an equivalence in general. To see why this is so, we first have to define the general notion of bisimulation using spans and co-spans more precisely.

Definition 17 *A* labeled prefix order $\langle \mathbb{U}, \preceq, A, \alpha \rangle$ *is a prefix order* $\langle \mathbb{U}, \preceq \rangle$ *together with a labeling function* $\alpha : \mathbb{U} \to A$.

Definition 18 *Two labeled prefix orders* $\langle \mathbb{U}, \preceq, A, \alpha \rangle$ *and* $\langle \mathbb{V}, \preceq, A, \beta \rangle$ *are* span bisimilar*, denoted* $\alpha \leftrightarroweq_s \beta$*, if there exists a prefix order* $\langle \mathbb{W}, \preceq \rangle$ *and future and history preserving surjections* $u : \mathbb{W} \to \mathbb{U}$ *and* $v : \mathbb{W} \to \mathbb{V}$ *such that* $\alpha \circ u = \beta \circ v$.

Definition 19 *Two labeled prefix orders* $\langle \mathbb{U}, \preceq, A, \alpha \rangle$ *and* $\langle \mathbb{V}, \preceq, A, \beta \rangle$ *are* co-span bisimilar*, denoted* $\alpha \leftrightarroweq_c \beta$*, if there exists a third labeled prefix order* $\langle \mathbb{W}, \preceq, A, \gamma \rangle$ *and future and history preserving surjections* $u : \mathbb{U} \to \mathbb{W}$ *and* $v : \mathbb{V} \to \mathbb{W}$ *such that* $\gamma \circ u = \alpha$ *and* $\gamma \circ v = \beta$.

To see why span bisimilarity is not an equivalence, we only need to label the prefix orders $-\Omega$ and $-\mathbb{N}$ using a constant labeling $\alpha(\omega) = 1$ for all $\omega \in -\Omega$ and $\beta(n) = 1$ for all $n \in -\mathbb{N}$. Recall from section 3.1 that the product of $-\Omega$ and $-\mathbb{N}$ is the empty set, so the only span $u : \mathbb{W} \to -\Omega$ and $v : \mathbb{W} \to -\mathbb{N}$ that can exist between the two (even without considering the labeling) has $\mathbb{W} = \emptyset$. Obviously, this span is not surjective, hence $-\Omega$ and $-\mathbb{N}$ are not bisimilar ($-\Omega \not\leftrightarroweq_s -\mathbb{N}$). But on the other hand, it is easy to verify that $-\Omega$ labeled as above is bisimilar to the singleton prefix order labeled by 1, and $-\mathbb{N}$ is also bisimilar to 1 (witnessed by the span of the identity and the only possible function from each of the prefix orders to 1). From this we conclude that span bisimilarity $\leftrightarroweq_s$ is not transitive, hence not an equivalence.

In the remainder of this section, we will focus on co-span bisimilarity, and show that it is an equivalence and a congruence for parallel composition and alternative composition.

Theorem 16 (Equivalence) *The relation* $\leftrightarroweq_c$ *on labeled prefix orders is an equivalence.*

Proof That co-span bisimilarity is symmetric is trivial. To see that $\leftrightarroweq_c$ is reflexive one only has to realize that the identity is a history and future preserving surjection. To see that $\leftrightarroweq_c$ is transitive, assume labeled prefix orders $\alpha : \mathbb{U}_1 \to A$, $\beta : \mathbb{U}_2 \to A$ and $\gamma : \mathbb{U}_3 \to A$ such that $\alpha \leftrightarroweq_c \beta$ and $\beta \leftrightarroweq_c \gamma$. Then by definition there exist labeled prefix orders $\mu : \mathbb{X}_{12} \to A$ and $\nu : \mathbb{X}_{23} \to A$ and future and history preserving surjections $f : \mathbb{U}_1 \to \mathbb{X}_{12}$, $g : \mathbb{U}_2 \to \mathbb{X}_{12}$, $h : \mathbb{U}_2 \to \mathbb{X}_{23}$, and $i : \mathbb{U}_3 \to \mathbb{X}_{23}$, such that the solid arrows in the the diagram in figure 5 commute. If we then determine the limit (also called *pushout*) of the maps g and h in this diagram, we obtain the prefix order $\mathbb{Y}$ with maps

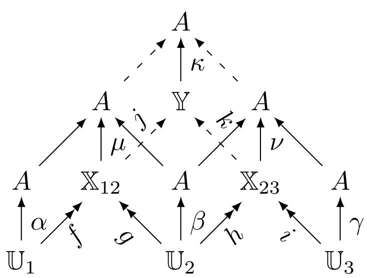

Figure 5: Co-span bisimulation is transitive. Unlabeled arrows represent identities.

$j : \mathbb{X}_{12} \to \mathbb{Y}$ and $k : \mathbb{X}_{23} \to \mathbb{Y}$. Furthermore, $\mathbb{Y}$ is generated from an order contracting backward bisimulation $\sim$ on $\mathbb{X}_{12} \uplus \mathbb{X}_{23}$, and because g and h are surjections every $y \in \mathbb{Y}$ represents an equivalence class in both $\mathbb{X}_{12}$ and $\mathbb{X}_{23}$, and the maps i and j are surjective. All the elements in these equivalence classes map to the same point in A, hence we obtain a labeling $\kappa : \mathbb{Y} \to A$ for which the diagram commutes. Finally, pick $x \in \mathbb{X}_{12}$ and $y \in \mathbb{Y}$ such that $j(x) \sqsubseteq y$. This means that there exists $x', x'' \in \mathbb{X}_{12}$ such that $x \sim x' \preceq x'' \sim y$. Backtracking these points along g gives us $u, u', u'' \in \mathbb{U}$ such that $g(u) = g(u')$ and $u' \preceq u''$. Furthermore, because g is future preserving, we find u''' such that $u \preceq u'''$ and $g(u''') = g(u'')$. By order preservation, this gives us $x \preceq g(u''')$ and using the commuting diagram we find $g(u''') \sim u'' \sim y$, so $j(g(u''')) = y$, from which we conclude that j is future preserving. In a similar vein k is future preserving, and so are $f \circ j$ and $k \circ i$, thus witnessing $\alpha \leftrightarrows_c \gamma$. ⊠

Next, we lift the notion of parallel composition and alternative composition to labeled prefix orders.

Definition 20 (Labeled compositions) *Given two labeled prefix orders $\alpha : \mathbb{U} \to A$, $\beta : \mathbb{V} \to B$ we define their parallel composition $\alpha \parallel \beta : \mathbb{U} \parallel \mathbb{W} \to A \times B$ by $(\alpha \times \beta)(H) = (\alpha(\pi_1(H)), \beta(\pi_2(H)))$ for all $H \in \mathbb{U} \parallel \mathbb{V}$, and we define their alternative composition $\alpha \uplus \beta : \mathbb{U} \uplus \mathbb{V} \to A \uplus B$ by $(\alpha \uplus \beta)(x) = \alpha(x)$ for $x \in \mathbb{U}$ and $(\alpha \uplus \beta)(x) = \beta(x)$ for $x \in \mathbb{V}$.*

It is easy to verify that $\alpha \parallel \beta$ and $\beta \parallel \alpha$ are isomorphic, hence bisimilar, and so are $\alpha \uplus \beta$ and $\beta \uplus \alpha$. But more importantly, bisimulation using co-spans turns out to be a congruence for these operators.

Theorem 17 (Congruence) *Given three labeled prefix orders $\alpha : \mathbb{U} \to A$, $\beta : \mathbb{V} \to A$ and $\gamma : \mathbb{W} \to B$ such that $\alpha \leftrightarrows_c \beta$, we find $\alpha \parallel \gamma \leftrightarrows_c \beta \times \gamma$ and $\alpha \uplus \gamma \leftrightarrows_c \beta \uplus \gamma$.*

Proof From the fact that $\alpha \leftrightarrows_c \beta$ we can construct $\kappa : \mathbb{X} \to A$ and future and history preserving surjections $f : \mathbb{U} \to \mathbb{X}$ and $g : \mathbb{U} \to \mathbb{X}$ such that $\kappa \circ f = \alpha$

and $\kappa \circ g = \beta$. Now we construct $\kappa \parallel \gamma$ and define $f' : \mathbb{U} \parallel \mathbb{W} \to \mathbb{X} \parallel \mathbb{W}$ and $g' : \mathbb{V} \parallel \mathbb{W} \to \mathbb{X} \parallel \mathbb{W}$ by $f'(H) = \{(f(u), w) \mid (u, w) \in H\}$ and $g'(H) = \{(g(v), w) \mid (v, w) \in H\}$. It is straightforward to verify that f' and g' are history and future preserving surjections, thus witnessing $\alpha \parallel \gamma \leftrightarroweq_c \beta \parallel \gamma$. Dually, we define construct $\kappa \uplus \gamma$ and define $f'' : \mathbb{U} \uplus \mathbb{W} \to \mathbb{X} \uplus \mathbb{W}$ and $g'' : \mathbb{V} \uplus \mathbb{W} \to \mathbb{X} \uplus \mathbb{W}$ by $f''(t) = f(t)$ if $t \in \mathbb{U}$ and $f''(t) = t$ if $t \in \mathbb{W}$, and $g''(t) = g(t)$ if $t \in \mathbb{V}$ and $g''(t) = t$ if $t \in \mathbb{W}$. Also for f'' and g'' it is straightforward to verify that they are history and future preserving surjections, witnessing $\alpha \uplus \gamma \leftrightarroweq_c \beta \uplus \gamma$. $\boxtimes$

5 Discussion and concluding remarks

We have shown that dynamical systems can be modeled as a set of executions under their natural prefix ordering, and that history preserving maps represent the refinement of a specification, thus allowing refinements between various types of dynamics in one unified framework. Furthermore, if refinements are complete in the sense that all and only specified behavior is refined, then the corresponding maps are surjective and future preserving.

One of the next steps, is to deal with structured operational semantics in a categorical fashion. Is it possible to create maps from any operation defined using structured operational semantics to the components it depends on? In general, the composition of two systems does not lead to a refinement, so there will not simply be a history preserving map. For example, the system $\mathbb{X} \biguplus \mathbb{Y}$ does not have natural maps back to $\mathbb{X}$ and $\mathbb{Y}$. However, there are natural *partial* history preserving maps from $\mathbb{X} \biguplus \mathbb{Y}$ to $\mathbb{X}$ and $\mathbb{Y}$. From the point of view of $\mathbb{X}$, the *composition* $\mathbb{X} \biguplus \mathbb{Y}$ is a combination of *refinement* and *specification*. The newly specified part is therefore undefined in the map to $\mathbb{X}$, while the refinement is mapped in a history preserving way. For the study of operational semantics in a category theoretic way, I therefore expect that partial history preserving maps may be helpful. A first exploration of using partial history preserving maps can be found in [2].

Another possible step, is to add more structure to the notion of prefix order, thus becoming less general but more applicable. Prefix orders really only model the dynamical properties of a system. If one would like to study timing, continuity, energy, or other properties, an observation map (like the one used in section 4.2) is needed. Incidentally, the map used in section 4 is itself a history preserving map, but other types of maps are conceivable as well. For example, if $\pi : \mathbb{U} \to Q$ and $\pi : \mathbb{V} \to Q$ map the executions of two systems to some (partially ordered) quantity Q, one could define that $\mathbb{U}$ is an *over-approximation* of $\mathbb{V}$ if there is a history and future preserving surjection $f : \mathbb{U} \to \mathbb{V}$ such that $\pi(f(u)) \leq \pi(u)$ for every $u \in \mathbb{U}$. Furthermore, the idea of *prefixing* is intimately coupled with the notion of *concatenation*, since prefixing is also often defined as: $x \preceq z$ iff $\exists_y \; x \cdot y = z$. It seems therefore reasonable to also study which *semigroups* $\langle \mathbb{U}, \cdot \rangle$ admit a natural prefix order. Finally, one could also study probabilistic systems by imposing a measure on the anti-chains of the prefix order, or one could study continuous systems by making using of the natural interval topology on prefix orders, and consider continuous maps between a prefix order and some physical variable.

When the observations on a prefix order change, the notion of bisimulation should change with it. Luckily, the definition of bisimulation through co-spans is very flexible in this respect. What is a bit disturbing, is that the definition through spans does not yield an equivalence. This may upset the general consensus in process algebra that any implementation can also be seen as a specification and vice versa. In the field of coalgebra, it was already known that there are alternative definitions of (strong) bisimulation that do not always coincide outside the domain of labeled transition systems (see e.g. [11]). Part of the added value of this paper is that we can now also do it for branching bisimulation, and part of the value is that the interpretation of morphisms as refinements gives us an explanation in 'natural language' of why the difference is there. If we consider specifications that have a common implementation this does not yield an equivalence, because transitivity only occurs when three equivalent systems also share a single common implementation. If we consider implementations that have a common specification we do obtain an equivalence, because two common implementations turn out to always allow a third, even more abstract, implementation.

In conclusion, adding observations in order to study different types of dynamical systems is reminiscent of the definition of executions as functions of time. Looking back, perhaps we did not succeed in eliminating the notion of time from our modeling paradigm after all. In stead, we did perhaps succeed in capturing, in an order theoretic way, the notion of a dynamical system as a function of *branching time*.

Acknowledgements go to Harsh Beohar, Erik de Vink and Ruurd Kuiper for their continuing and much appreciated support, and to MEALS for the opportunity to travel to Buenos Aires.

References

[1] J.C.M. Baeten, T. Basten & M.A. Reniers (2010): *Process Algebra: Equational Theories of Communicating Processes*. Cambridge Tracts in Theoretical Computer Science, Cambridge University Press.

[2] P.J.L. Cuijpers (2013): *The categorical limit of a sequence of dynamical systems*. In: *EXPRESS/SOS 2013*, *EPTCS* 120, pp. 78–92, doi:10.4204/EPTCS.120.7.

[3] P.J.L. Cuijpers (2013): *Prefix Orders as a General Model of Dynamics*. In: *DCM 2013*, Pre-proceedings of the 9th international workshop on developments in computational models (DCM 2013)., pp. 25–29.

[4] J.M. Davoren & P. Tabuada (2007): *On Simulations and Bisimulations of General Flow Systems*. In Alberto Bemporad, Antonio Bicchi & Giorgio Buttazzo, editors: *Hybrid Systems: Computation and Control*, *Lecture*

Notes in Computer Science 4416, Springer Berlin Heidelberg, pp. 145–158, doi:10.1007/978-3-540-71493-4_14.

[5] R.J. van Glabbeek (2001): *Current Trends in Theoretical Computer Science; Entering the 21st Century*, chapter What is Branching Time Semantics and Why to Use It?, pp. 469–479. World Scientific.

[6] J.Y. Halpern & K.R. O'Neill (2008): *Secrecy in Multiagent Systems. ACM Trans. Inf. Syst. Secur.* 12, pp. 5:1–5:47, doi:10.1145/1410234.1410239.

[7] K. Kunen (1988): *Set Theory: An Introduction to Independence Proofs*, third edition. *Studies In Logic and the Foundations of Mathematics* 102, Elsevier Science B.V.

[8] S. MacLane (1971): *Categories for the Working Mathematician.* Springer-Verlag.

[9] R. Nicola, U. Montanari & F. Vaandrager (1990): *Back and forth bisimulations.* In J.C.M. Baeten & J.W. Klop, editors: *CONCUR '90 Theories of Concurrency: Unification and Extension*, *Lecture Notes in Computer Science* 458, Springer Berlin Heidelberg, pp. 152–165, doi:10.1007/BFb0039058.

[10] J.W. Polderman & J.C. Willems (1998): *Introduction to Mathematical Systems Theory: A Behavioural Approach. Texts in Applied Mathematics* 26, Springer-Verlag.

[11] S. Staton (2009): *Relating Coalgebraic Notions of Bisimulation.* In A. Kurz, M. Lenisa & A. Tarlecki, editors: *Algebra and Coalgebra in Computer Science*, *Lecture Notes in Computer Science* 5728, Springer Berlin Heidelberg, pp. 191–205, doi:10.1007/978-3-642-03741-2_14.

[12] L.A. Steen & J.A. Seebach (1970): *Counterexamples in topology.*

[13] G. Winskel & M. Nielsen (1995): *Handbook of logic in computer science (vol. 4).* chapter Models for concurrency, Oxford University Press, Oxford, UK, pp. 1–148. Available at http://portal.acm.org/citation.cfm?id=218623.218630.

Science Reports | **Department of Mathematics and Computer Science**
Technische Universiteit Eindhoven

If you want to receive reports, send an email to: wsinsan@tue.nl (we cannot guarantee the availability of the requested reports).

In this series appeared (from 2009):

09/01	Wil M.P. van der Aalst, Kees M. van Hee, Peter Massuthe, Natalia Sidorova and Jan Martijn van der Werf	Compositional Service Trees
09/02	P.J.l. Cuijpers, F.A.J. Koenders, M.G.P. Pustjens, B.A.G. Senders, P.J.A. van Tilburg, P. Verduin	Queue merge: a Binary Operator for Modeling Queueing Behavior
09/03	Maarten G. Meulen, Frank P.M. Stappers and Tim A.C. Willemse	Breadth-Bounded Model Checking
09/04	Muhammad Atif and MohammadReza Mousavi	Formal Specification and Analysis of Accelerated Heartbeat Protocols
09/05	Michael Franssen	Placeholder Calculus for First-Order logic
09/06	Daniel Trivellato, Fred Spiessens, Nicola Zannone and Sandro Etalle	POLIPO: Policies & OntoLogies for the Interoperability, Portability, and autOnomy
09/07	Marco Zapletal, Wil M.P. van der Aalst, Nick Russell, Philipp Liegl and Hannes Werthner	Pattern-based Analysis of Windows Workflow
09/08	Mike Holenderski, Reinder J. Bril and Johan J. Lukkien	Swift mode changes in memory constrained real-time systems
09/09	Dragan Bošnački, Aad Mathijssen and Yaroslav S. Usenko	Behavioural analysis of an I²C Linux Driver
09/10	Ugur Keskin	In-Vehicle Communication Networks: A Literature Survey
09/11	Bas Ploeger	Analysis of ACS using mCRL2
09/12	Wolfgang Boehmer, Christoph Brandt and Jan Friso Groote	Evaluation of a Business Continuity Plan using Process Algebra and Modal Logic
09/13	Luca Aceto, Anna Ingolfsdottir, MohammadReza Mousavi and Michel A. Reniers	A Rule Format for Unit Elements
09/14	Maja Pešić, Dragan Bošnački and Wil M.P. van der Aalst	Enacting Declarative Languages using LTL: Avoiding Errors and Improving Performance
09/15	MohammadReza Mousavi and Emil Sekerinski, Editors	Proceedings of Formal Methods 2009 Doctoral Symposium
09/16	Muhammad Atif	Formal Analysis of Consensus Protocols in Asynchronous Distributed Systems
09/17	Jeroen Keiren and Tim A.C. Willemse	Bisimulation Minimisations for Boolean Equation Systems
09/18	Kees van Hee, Jan Hidders, Geert-Jan Houben, Jan Paredaens, Philippe Thiran	On-the-fly Auditing of Business Processes
10/01	Ammar Osaiweran, Marcel Boosten, MohammadReza Mousavi	Analytical Software Design: Introduction and Industrial Experience Report
10/02	F.E.J. Kruseman Aretz	Design and correctness proof of an emulation of the floating-point operations of the Electrologica X8. A case study

10/03	Luca Aceto, Matteo Cimini, Anna Ingolfsdottir, MohammadReza Mousavi and Michel A. Reniers	On Rule Formats for Zero and Unit Elements
10/04	Hamid Reza Asaadi, Ramtin Khosravi, MohammadReza Mousavi, Neda Noroozi	Towards Model-Based Testing of Electronic Funds Transfer Systems
10/05	Reinder J. Bril, Uğur Keskin, Moris Behnam, Thomas Nolte	Schedulability analysis of synchronization protocols based on overrun without payback for hierarchical scheduling frameworks revisited
10/06	Zvezdan Protić	Locally unique labeling of model elements for state-based model differences
10/07	C.G.U. Okwudire and R.J. Bril	Converting existing analysis to the EDP resource model
10/08	Muhammed Atif, Sjoerd Cranen, MohammadReza Mousavi	Reconstruction and verification of group membership protocols
10/09	Sjoerd Cranen, Jan Friso Groote, Michel Reniers	A linear translation from LTL to the first-order modal μ-calculus
10/10	Mike Holenderski, Wim Cools Reinder J. Bril, Johan J. Lukkien	Extending an Open-source Real-time Operating System with Hierarchical Scheduling
10/11	Eric van Wyk and Steffen Zschaler	1st Doctoral Symposium of the International Conference on Software Language Engineering (SLE)
10/12	Pre-Proceedings	3rd International Software Language Engineering Conference
10/13	Faisal Kamiran, Toon Calders and Mykola Pechenizkiy	Discrimination Aware Decision Tree Learning
10/14	J.F. Groote, T.W.D.M. Kouters and A.A.H. Osaiweran	Specification Guidelines to avoid the State Space Explosion Problem
10/15	Daniel Trivellato, Nicola Zannone and Sandro Etalle	GEM: a Distributed Goal Evaluation Algorithm for Trust Management
10/16	L. Aceto, M. Cimini, A.Ingolfsdottir, M.R. Mousavi and M. A. Reniers	Rule Formats for Distributivity
10/17	L. Aceto, A. Birgisson, A. Ingolfsdottir, and M.R. Mousavi	Decompositional Reasoning about the History of Parallel Processes
10/18	P.D. Mosses, M.R. Mousavi and M.A. Reniers	Robustness os Behavioral Equivalence on Open Terms
10/19	Harsh Beohar and Pieter Cuijpers	Desynchronisability of (partial) closed loop systems
11/01	Kees M. van Hee, Natalia Sidorova and Jan Martijn van der Werf	Refinement of Synchronizable Places with Multi-workflow Nets - Weak termination preserved!
11/02	M.F. van Amstel, M.G.J. van den Brand and L.J.P. Engelen	Using a DSL and Fine-grained Model Transformations to Explore the boundaries of Model Verification
11/03	H.R. Mahrooghi and M.R. Mousavi	Reconciling Operational and Epistemic Approaches to the Formal Analysis of Crypto-Based Security Protocols
11/04	J.F. Groote, A.A.H. Osaiweran and J.H. Wesselius	Benefits of Applying Formal Methods to Industrial Control Software
11/05	Jan Friso Groote and Jan Lanik	Semantics, bisimulation and congruence results for a general stochastic process operator
11/06	P.J.L. Cuijpers	Moore-Smith theory for Uniform Spaces through Asymptotic Equivalence
11/07	F.P.M. Stappers, M.A. Reniers and S. Weber	Transforming SOS Specifications to Linear Processes
11/08	Debjyoti Bera, Kees M. van Hee, Michiel van Osch and Jan Martijn van der Werf	A Component Framework where Port Compatibility Implies Weak Termination
11/09	Tseesuren Batsuuri, Reinder J. Bril and Johan Lukkien	Model, analysis, and improvements for inter-vehicle communication using one-hop periodic broadcasting based on the 802.11p protocol

11/10	Neda Noroozi, Ramtin Khosravi, MohammadReza Mousavi and Tim A.C. Willemse	Synchronizing Asynchronous Conformance Testing
11/11	Jeroen J.A. Keiren and Michel A. Reniers	Type checking mCRL2
11/12	Muhammad Atif, MohammadReza Mousavi and Ammar Osaiweran	Formal Verification of Unreliable Failure Detectors in Partially Synchronous Systems
11/13	J.F. Groote, A.A.H. Osaiweran and J.H. Wesselius	Experience report on developing the Front-end Client unit under the control of formal methods
11/14	J.F. Groote, A.A.H. Osaiweran and J.H. Wesselius	Ananlyzing a Controller of a Power Distribution Unit Using Formal Methods
11/15	John Businge, Alexander Serebrenik and Mark van den Brand	Eclipse API Usage: The Good and The Bad
11/16	J.F. Groote, A.A.H. Osaiweran, M.T.W. Schuts and J.H. Wesselius	Investigating the Effects of Designing Control Software using Push and Poll Strategies
11/17	M.F. van Amstel, A. Serebrenik And M.G.J. van den Brand	Visualizing Traceability in Model Transformation Compositions
11/18	F.P.M. Stappers, M.A. Reniers, J.F. Groote and S. Weber	Dogfooding the Structural Operational Semantics of mCRL2
12/01	S. Cranen	Model checking the FlexRay startup phase
12/02	U. Khadim and P.J.L. Cuijpers	Appendix C / G of the paper: Repairing Time-Determinism in the Process Algebra for Hybrid Systems ACP
12/03	M.M.H.P. van den Heuvel, P.J.L. Cuijpers, J.J. Lukkien and N.W. Fisher	Revised budget allocations for fixed-priority-scheduled periodic resources
12/04	Ammar Osaiweran, Tom Fransen, Jan Friso Groote and Bart van Rijnsoever	Experience Report on Designing and Developing Control Components using Formal Methods
12/05	Sjoerd Cranen, Jeroen J.A. Keiren and Tim A.C. Willemse	A cure for stuttering parity games
12/06	A.P. van der Meer	CIF MSOS type system
12/07	Dirk Fahland and Robert Prüfer	Data and Abstraction for Scenario-Based Modeling with Petri Nets
12/08	Luc Engelen and Anton Wijs	Checking Property Preservation of Refining Transformations for Model-Driven Development
12/09	M.M.H.P. van den Heuvel, M. Behnam, R.J. Bril, J.J. Lukkien and T. Nolte	Opaque analysis for resource-sharing components in hierarchical real-time systems - extended version –
12/10	Milosh Stolikj, Pieter J. L. Cuijpers and Johan J. Lukkien	Efficient reprogramming of sensor networks using incremental updates and data compression
12/11	John Businge, Alexander Serebrenik and Mark van den Brand	Survival of Eclipse Third-party Plug-ins
12/12	Jeroen J.A. Keiren and Martijn D. Klabbers	Modelling and verifying IEEE Std 11073-20601 session setup using mCRL2
12/13	Ammar Osaiweran, Jan Friso Groote, Mathijs Schuts, Jozef Hooman and Bart van Rijnsoever	Evaluating the Effect of Formal Techniques in Industry
12/14	Ammar Osaiweran, Mathijs Schuts, and Jozef Hooman	Incorporating Formal Techniques into Industrial Practice
13/01	S. Cranen, M.W. Gazda, J.W. Wesselink and T.A.C. Willemse	Abstraction in Parameterised Boolean Equation Systems
13/02	Neda Noroozi, Mohammad Reza Mousavi and Tim A.C. Willemse	Decomposability in Formal Conformance Testing

13/03	D. Bera, K.M. van Hee and N. Sidorova	Discrete Timed Petri nets
13/04	A. Kota Gopalakrishna, T. Ozcelebi, A. Liotta and J.J. Lukkien	Relevance as a Metric for Evaluating Machine Learning Algorithms
13/05	T. Ozcelebi, A. Weffers-Albu and J.J. Lukkien	Proceedings of the 2012 Workshop on Ambient Intelligence Infrastructures (WAmIi)
13/06	Lotfi ben Othmane, Pelin Angin, Harold Weffers and Bharat Bhargava	Extending the Agile Development Process to Develop Acceptably Secure Software
13/07	R.H. Mak	Resource-aware Life Cycle Models for Service-oriented Applications managed by a Component Framework
13/08	Mark van den Brand and Jan Friso Groote	Software Engineering: Redundancy is Key
13/09	P.J.L. Cuijpers	Prefix Orders as a General Model of Dynamics