Models for Name-Passing Processes: Interleaving and Causal

Gian Luca Cattani* and Peter Sewell†
Computer Laboratory
University of Cambridge
England
{Luca.Cattani,Peter.Sewell}@cl.cam.ac.uk

September 2000

Abstract

We study syntax-free models for name-passing processes. For interleaving semantics, we identify the indexing structure required of an early labelled transition system to support the usual π-calculus operations, defining *Indexed Labelled Transition Systems*. For non-interleaving causal semantics we define *Indexed Labelled Asynchronous Transition Systems*, smoothly generalizing both our interleaving model and the standard Asynchronous Transition Systems model for CCS-like calculi. In each case we relate a denotational semantics to an operational view, for bisimulation and causal bisimulation respectively. We establish completeness properties of, and adjunctions between, categories of the two models. Alternative indexing structures and possible applications are also discussed. These are first steps towards a uniform understanding of the semantics and operations of name-passing calculi.

Contents

*supported by EPSRC grant GR/L62290: Calculi for Interactive Systems: Theory and Experiment
†supported by a Royal Society University Research Fellowship

1 Introduction

The study of concurrency has involved rich interplay between model-theoretic and syntactic approaches. The first takes a notion of behaviour – perhaps defined as some class of automata or labelled transition systems – as primary; the second focuses on some particular signature of process terms, perhaps giving it only an axiomatic semantics. It is now common to take an intermediate approach: to fix a signature of process terms and equip it with an operational semantics defining behaviour (e.g. transition relations) over those terms. This has been followed for almost all work on π-calculi, beginning with [40], in which an operational semantics defines transition relations with particular labels over π-terms. By contrast, in this paper we study purely model-theoretic notions of behaviour for π-calculi, with definitions that do not involve process syntax, to support the uniform development of metatheory for a range of calculi and semantics. For interleaving semantics we introduce *Indexed Labelled Transition Systems* with data specifying how transitions change under renaming – thus picking out the essential structure of a π early transition relation that is required for defining the normal operations and equivalences over π-terms. For non-interleaving causal semantics, we define *Indexed Labelled Asynchronous Transition Systems*, smoothly generalizing both our interleaving model and the standard Asynchronous Transition Systems model for CCS-like calculi [3, 55, 58]. In each case we give a denotational semantics of a π-calculus; we prove that the operational early and causal bisimulations [47, 6] coincide with model-theoretic notions. We also establish completeness properties of and adjunctions between categories of the two models, as first steps towards a uniform understanding of the semantics and operations of name-passing calculi. A number of alternative structures and applications of the models can be envisaged, including applications to model-checking and security reasoning; we briefly outline some possible directions.

Interleaving The standard notion of labelled transition system (LTS) for calculi without value-passing is straightforward. For example, given a set N of channel names (ranged over by $a, b, \ldots$) the CCS fragment

$$P ::= 0 \;\big|\; \overline{a}.P \;\big|\; a.P \;\big|\; P \mid Q \;\big|\; (\boldsymbol{\nu} c)P$$

can be given semantics in terms of LTSs

$$\langle S, \longrightarrow, i \rangle$$

where S is a set of states, $\longrightarrow \subseteq S \times \mathcal{L} \times S$ is a transition relation with labels $\mathcal{L} = \{\tau, a, \overline{a}, b, \overline{b}, \ldots\}$, and $i \in S$ is the initial state. Introducing value-passing, however, makes the situation more complex – particularly with scope extrusion. Consider the π-calculus fragment below, in which the 'c' in the input $bc.P$ and restriction $(\boldsymbol{\nu} c)P$ bind in the process P.

$$P ::= 0 \;\big|\; \overline{a}d.P \;\big|\; bc.P \;\big|\; P \mid Q \;\big|\; (\boldsymbol{\nu} c)P$$

Defining the behaviour of $bc.P$ involves substitution. For example, the communication of a free name

$$\overline{a}d.P \mid ac.Q \xrightarrow{\tau} P \mid \{{}^{d}\!/\!{}_{c}\}Q$$

is inferred in the 'early' semantics of [41, 47] with the rules below.

$$\textsc{Out}\,\frac{}{\overline{a}d.P \xrightarrow{\overline{a}d} P} \qquad \textsc{In}\,\frac{}{ac.Q \xrightarrow{ad} \{{}^{d}\!/\!{}_{c}\}Q}$$

$$\textsc{Com}\,\frac{P \xrightarrow{\overline{a}d} P' \quad Q \xrightarrow{ad} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

Note that d might or might not be in the free names of Q. Moreover, unlike in CCS, π-calculus τ-transitions can also involve scope extrusion:

$$((\boldsymbol{\nu} d)\overline{a}d.P) \mid ac.Q \xrightarrow{\tau} (\boldsymbol{\nu} d)(P \mid \{^d/_c\}Q) \quad \text{if } d \notin \mathrm{fn}(Q)$$

To define the τ-transitions of $P \mid Q$ compositionally, in terms of the transitions of P and Q, the semantics must distinguish between outputs of free and bound names, by taking transitions with labels $\overline{a}d$ and $\overline{a}(d)$ respectively. The τ-transition above can be inferred with the rules:

$$\textsc{Open}\ \frac{P \xrightarrow{\overline{a}d} P' \quad d \neq a}{(\boldsymbol{\nu} d)P \xrightarrow{\overline{a}(d)} P'}$$

$$\textsc{Close}\ \frac{P \xrightarrow{\overline{a}(d)} P' \quad Q \xrightarrow{ad} Q' \quad d \notin \mathrm{fn}(Q)}{P \mid Q \xrightarrow{\tau} (\boldsymbol{\nu} d)(P' \mid Q')}$$

The full semantics requires also the rules

$$\textsc{Res}\ \frac{P \xrightarrow{\ell} P' \quad d \notin \mathrm{fn}(\ell)}{(\boldsymbol{\nu} d)P \xrightarrow{\ell} (\boldsymbol{\nu} d)P'}$$

$$\textsc{Par}\ \frac{P \xrightarrow{\ell} P' \quad \mathrm{bn}(\ell) \cap \mathrm{fn}(Q) = \emptyset}{P \mid Q \xrightarrow{\ell} P' \mid Q}$$

(in which $\mathrm{bn}(\overline{a}(d)) = \{d\}$, and $\mathrm{bn}(\ell) = \emptyset$ for labels of other forms) for restricted transitions that do not involve scope extrusion and for parallel.

These SOS rules involve subtle conditions on the free names of process terms (relating them to names in labels), in addition to name substitution on process terms. To give a syntax-free notion of LTS that has enough structure to define the operations we must therefore consider states not simply to be elements of an arbitrary set but of a set indexed by finite sets of names – the 'free' names of the states – and add data specifying how states change under renaming. In Section 3 we will define an Indexed Labelled Transition System (or $\mathcal{N}$-LTS) to have data

$$\langle S : \mathcal{N} \to \mathbf{Set},\ \longrightarrow,\ \langle I, i \rangle \rangle$$

where S is a functor from an indexing category $\mathcal{N}$ of name-sets and renamings into **Set** (the category of all sets and functions), giving the set of states above each name-set; the transition relation is over the coproduct $\coprod_{A \in |\mathcal{N}|} S(A)$; and the initial state $\langle I, i \rangle$ is an element of that coproduct. Axioms must be imposed, enforcing:

1. the name-sets of the endpoints of a transition must be related to each other and to the label;
2. input transitions occur in families related by renaming of the result states;
3. (a) transitions are preserved by injective renaming, both of the names of states and of new names in labels;
   (b) inputs of new names above a name-set give rise to inputs of old names above larger name-sets; and
4. the transitions of an injective renaming of a state are determined by the transitions of the state.

We give the precise definition of $\mathcal{N}$-LTS in Section 3, following a description of the π-calculus we are using in Section 2. We also introduce categories $\mathcal{N}$-$\mathbf{LTS}_I$ (for each initial-name set I) of $\mathcal{N}$-LTSs and study their completeness properties. Many variant definitions of $\mathcal{N}$-LTS are possible; we discuss the alternatives in Section 7. In Section 4 we define constructions over $\mathcal{N}$-LTSs, giving a denotational semantics, and relate bisimulation over $\mathcal{N}$-LTSs with the bisimulation defined using the operational semantics.

Non-Interleaving models for process calculi have been much studied; they can support model-checking techniques that mitigate the state-explosion problem, and strong proof techniques. They are also required in cases where the desired properties of systems are most naturally stated in terms of causality or locality. Here again there are model-theoretic and syntactic approaches – the first is surveyed in [58]; the second is represented by various annotated operational semantics, e.g. [16, 8, 15, 32, 58]. The two seem to have been carried out almost independently – to our knowledge, the only works to make precise connections are [16, 8, 58]. Moreover, only the syntactic approach has been developed to address name-passing, in the annotated operational models of [6, 18]. There is also work that does not fit this categorisation, having both syntactic and model-theoretic aspects, with Petri nets and graph rewriting [9, 42].

Our goal in the second half of this paper is to develop the model-theoretic approach, and to make precise connections to the annotated operational notions. We develop a simple syntax-free non-interleaving model for name-passing that generalises both our interleaving model and the standard Asynchronous Transition Systems model for calculi without name-passing [3, 55, 58]. This is precisely related to causal bisimulation [6].

In CCS causal dependency arises from prefixing – in the behaviour of the process $\overline{x}.\overline{y}.0$ the y output causally depends on the x output. In π-calculus, name-binding introduces new dependencies, as thoroughly discussed in [18]. Transitions occurring in different parallel components of a process term, naively regarded as independent, may be forced to occur in a fixed order. For example, in the process $(\boldsymbol{\nu} y)(\overline{x}y \mid \overline{y}z)$ the transition $\overline{y}z$ can be observed only after $\overline{x}y$ – before this occurs the new-bound channel is not known to the environment. The two transitions of $(\boldsymbol{\nu} y)(\overline{x}y \mid \overline{z}y)$ are independent, however, despite the fact that the first to occur will be an output of a new name and the second will not. Further, an input of a previously-extruded name, e.g. $(\boldsymbol{\nu} y)(\overline{x}y \mid xw.0) \xrightarrow{\overline{x}y}\xrightarrow{xy} 0$, or output of a previously-input new name, e.g. $xw.\overline{x}w \xrightarrow{xy}\xrightarrow{\overline{x}y} 0$ (where y is new) involves dependency. Moreover, one can choose whether or not to distinguish between the prefix and name dependency, e.g. whether to identify $(\boldsymbol{\nu} y)(\overline{x}y.\overline{y}z)$ and $(\boldsymbol{\nu} y)(\overline{x}y \mid \overline{y}z)$.

In Section 5 we define a relation of name-dependency between two labels (wrt. a name-set), and then an Indexed Labelled Asynchronous Transition System (or $\mathcal{N}$-LATS) to have data

$$\langle S : \mathcal{N} \to \mathbf{Set},\ \longrightarrow,\ \langle I, i \rangle,\ E,\ \mathcal{I} \rangle$$

where now transitions are annotated by elements of a set E of *events* and $\mathcal{I} \subseteq E \times E$ is an *independence relation* between events. We impose axioms requiring that one obtains an Indexed LTS when considering each $e \in E$ separately, and (roughly), that independent transitions can be permuted. As one would expect, name dependency is involved in the relationship between the transition and independence relations. We discuss how the constructions of Section 4 can be extended to $\mathcal{N}$-LATS, define history-preserving bisimulation and a name-dependency aware variant (respectively distinguishing and identifying the example two processes above), and prove correspondence results.

In Section 6 we continue the abstract study of the structures defined in the paper. We define categories $\mathcal{N}$-$\mathbf{LATS}_I$ of Indexed LATS (each for initial name-set I) and study their completeness properties as well as their relationship with the categories $\mathcal{N}$-$\mathbf{LTS}_I$. These are first steps towards an abstract understanding of the equivalences and constructions involved in the semantics of π-like process languages.

Further Motivation, Future Directions and Related Work Viewing models categorically has proven useful in study of the interleaving/non-interleaving and linear-time/branching-time distinctions [49]. Moreover, the categorical study of process calculi gives the possibility of obtaining general congruence results: in [58] categorical models of CCS-like processes are axiomatised and in [31] an abstract model-theoretic notion of bisimulation is introduced (via open maps); in [13, 14] these two are combined to give abstract congruence results for strong bisimulation over a wide range of models. It is our hope that the present work serves as a first step towards similar results for π-calculus-like process languages. In particular, we would like a categorical understanding of our operations for the two models, related by the results presented in Section 6. One could then use these results to address the problem of giving causal semantics to variants of the π-calculus, e.g. the box-π of [53, 54], for which an approximate notion of causality is used to state security properties. Preliminary discussion of this, and of other future directions, can be found in Section 8.

Among earlier models of π-processes, the name passing synchronisation trees of [27] and presheaves of [12] are the closest to our $\mathcal{N}$-LTS, though they employ a slightly different indexing structure (*cf.* Section 7). In [12] the models are defined using domain theoretical techniques similar to those employed in [56, 20], as the solutions to semantic equations. By contrast here we take a more concrete approach, with several advantages. Firstly, it is easy to conceive of minor modifications to our definitions to suit calculi such as the asynchronous π-calculus [7, 28]. In particular it should be quite straightforward to adapt the axioms of [50] to our models. It should also be easy to address the πI-calculus [48], in which only new names are communicated (though this can also be done domain-theoretically). Secondly, it supports a direct definition of weak bisimulation, something the domain model lacks completely and the presheaf model can, as far as we know, only achieve indirectly by means of a saturation construction [19].

It is also worth noticing that while the domain models are tailored for late bisimulation, our focus here is on early semantics, both to obtain a simpler notion of transition system, and because we have found the early style suits work on concurrent language semantics and on secure encapsulation [51, 53, 54, 52]. Presheaf models exist for both early and late notions [10]. Moreover we should add that, in contrast to [56, 20] (which have full-abstraction results wrt. strong bisimulation), we focus on intensional models, over which a number of equivalences can be defined (though we give results only for bisimulation). The literature contains also testing-based models [24, 4]. The precise relationships with these and other models defined in the literature, e.g. [42, 9, 30] requires further work.

More speculatively, we believe our structures may form a useful basis for π-calculus interleaving and partial-order model checking, via notions of finitely-generable $\mathcal{N}$-LTS and $\mathcal{N}$-LATS – the former of which may have interesting relationships with the HD-automata of [43], and wonder what the relationships are with the recent [22, 21, 26], where similar indexing structure is used in a λ-calculus setting.

Finally, notice that in this paper we introduce transition systems with indexed sets of states, but not indexed sets of transitions. This is because, as remarked above, when moving from a name-set to a larger one, transitions labelled with inputs of new names in the former give rise to input transitions of both new and old names in the latter – the correspondence between transitions is not functional, even for injective renamings. It may be possible to use more sophisticated indexing structures which allow transitions as well as states to be indexed; the pay-off for the extra complication being e.g. the possibility of using the notion of internal category to formally relate our Indexed Transition Systems with the standard ones.

This paper is a full and extended version of [11].

$$\textsc{Out}\ \frac{}{A \vdash \overline{x}v.P \xrightarrow{\overline{x}v} P} \qquad \textsc{In}\ \frac{}{A \vdash xp.P \xrightarrow{xv} \{^v/_p\}P}$$

$$\textsc{Tau}\ \frac{}{A \vdash \tau.P \xrightarrow{\tau} P} \qquad \textsc{Sum}\ \frac{A \vdash P \xrightarrow{\ell} P'}{A \vdash P + Q \xrightarrow{\ell} P'}$$

$$\textsc{Par}\ \frac{A \vdash P \xrightarrow{\ell} P'}{A \vdash P \mid Q \xrightarrow{\ell} P' \mid Q} \qquad \textsc{Com}\ \frac{A \vdash P \xrightarrow{\overline{x}v} P' \quad A \vdash Q \xrightarrow{xv} Q'}{A \vdash P \mid Q \xrightarrow{\tau} (\boldsymbol{\nu}\{v\} \setminus A)(P' \mid Q')}$$

$$\textsc{Res}\ \frac{A, x \vdash P \xrightarrow{\ell} P' \quad x \notin \mathrm{fn}(\ell)}{A \vdash (\boldsymbol{\nu}x)P \xrightarrow{\ell} (\boldsymbol{\nu}x)P'} \qquad \textsc{Open}\ \frac{A, x \vdash P \xrightarrow{\overline{y}x} P' \quad y \neq x}{A \vdash (\boldsymbol{\nu}x)P \xrightarrow{\overline{y}x} P'}$$

$$\textsc{Match}\ \frac{A \vdash P \xrightarrow{\ell} P'}{A \vdash [x = x]P \xrightarrow{\ell} P'} \qquad \textsc{Mismatch}\ \frac{A \vdash P \xrightarrow{\ell} P' \quad x \neq y}{A \vdash [x \neq y]P \xrightarrow{\ell} P'}$$

In all rules with conclusion of the form $A \vdash P \xrightarrow{\ell} Q$ there is an implicit side condition $\mathrm{fn}(P) \subseteq A$. Symmetric versions of Par, Com and Sum are elided.

Figure 1: π operational semantics

2 Background on the π-calculus

Many variant π-calculi have been studied in the literature since the original was introduced in [40]. Here, to show the wide applicability of our models, we take a rich set of primitives including summation, matching, mismatching and synchronous output. For notational simplicity, however, we treat only a monadic untyped calculus without basic values, and also omit replication. These could be easily added.

Syntax We take an infinite set N of *names* of channels, ranged over by a, b etc. The *process terms* are then those defined by the grammar

P, Q	$::=$	0	nil
		$P \mid Q$	parallel composition
		$P + Q$	choice
		$\tau.P$	internal action
		$\overline{a}d.P$	output d on channel a
		$ac.P$	input from channel a
		$(\boldsymbol{\nu}c)P$	new channel name creation
		$[a = b]P$	match
		$[a \neq b]P$	mismatch

Here the c in the input $bc.P$ and restriction $(\boldsymbol{\nu}c)P$ bind in the process P; we work up to alpha renaming of bound names. We write $\mathrm{fn}(P)$ for the set of free names of P, and $\{a/b\}P$ for the process term obtained from P by replacing all free occurrences of b by a.

Operational semantics We equip the calculus with a mild variant, explicitly-indexed, of the early labelled transition semantics of [47, 41], in which transitions are given for processes with respect to explicit supersets of their free name sets. This style simplifies the SOS rules, allowing sideconditions in Par and Close (here coalesced with Com) to be removed, gives a simple notion of trace, and supports subtype systems; it has been useful for work on concurrent language semantics and on secure encapsulation [51, 53, 54]. It is related to the original semantics at the end of this section. The labelled transition relation has the form

$$A \vdash P \xrightarrow{\ell} Q$$

where A is a finite set of names and $\mathrm{fn}(P) \subseteq A$; it should be read as 'in a state where the names A may be known by process P and by its environment, the process P can do ℓ to become Q. The labels Lab are $\{\tau\} \cup \{\, \overline{x}y \mid x, y \in N \,\} \cup \{\, xy \mid x, y \in N \,\}$. Note that we now have only one form of output label – a transition $A \vdash P \xrightarrow{\overline{x}v} Q$ is an output of a new name iff $v \notin A$. The transition relation is defined as the smallest relation satisfying the rules in Figure 1. The free names of a label are $\mathrm{fn}(\tau) = \{\}$, $\mathrm{fn}(\overline{x}v) = \mathrm{fn}(xv) = \{x, v\}$. We write A, x for $A \cup \{x\}$ where x is assumed not to be in A. If $A = \emptyset$ then $(\boldsymbol{\nu} A)P$ denotes P.

Note that the set of free names of a process can grow along transitions, for example $\{a\} \vdash (\boldsymbol{\nu} d)\overline{a}d.\overline{a}d \xrightarrow{\overline{a}d} \overline{a}d$, and that the rules depend in an essential way on alpha-conversion – the process $R = (\boldsymbol{\nu} d)\overline{a}d$ must be able to perform a bound output with label $\overline{a}\hat{d}$ for any $\hat{d} \neq a$; derivations of such transitions require use of the alpha-equivalence $R = (\boldsymbol{\nu}\hat{d})\overline{a}\hat{d}$. Note also that the SOS rules do not involve any structural congruence.

Example Properties We illustrate the SOS with some example transitions and properties – these will be special cases of the axioms imposed on $\mathcal{N}$-LTS in Section 3.

1. If $A \vdash P \xrightarrow{\overline{x}z} Q$ then $x \in A$. We might have z new, i.e. $z \notin A$ or not, i.e. $z \in A$. In either case, Q has free names contained in $A \cup \{x, z\}$. The same holds for input transitions.
2. A transition $A \vdash P \xrightarrow{xz} Q$ must arise from an input prefix in P, which must therefore be able to input any other name (new or old). Moreover, the resulting states can all be obtained by substitution from the resulting state after a new name is input.
3. (a) If $A \vdash P \xrightarrow{\overline{x}z} Q$ and $z \in A$ then for any injective substitution, say $f : A \rightarrow_{\mathrm{inj}} B$, there is a transition $B \vdash fP \xrightarrow{\overline{fx}fz} fQ$. For output of a new name, i.e. $z \notin A$, the value z can also be renamed to any $\hat{z} \notin B$, giving $B \vdash fP \xrightarrow{\overline{fx}\hat{z}} (f + [\hat{z}/z])Q$. The same holds for input transitions.
   (b) A derivation of an input $A \vdash P \xrightarrow{xz} Q$ of a new name $z \notin A$ is preserved by extending the name-set – so P above (A, z) has an input of an old name $A, z \vdash P \xrightarrow{xz} Q$.
4. Non-injective renaming can enable and (with mismatch) disable transitions, but the behaviour of an injective renaming of P is determined by that of P.

Operational Equivalences The normal notion of early bisimulation can be easily adapted to the explicitly-indexed setting. Take *bisimulation* $\dot{\sim}$ to be the largest family of relations indexed by finite sets of names such that each $\dot{\sim}_A$ is a symmetric relation over $\{\, P \mid \mathrm{fn}(P) \subseteq A \,\}$ and for all $P \mathrel{\dot{\sim}_A} Q$,

- if $A \vdash P \xrightarrow{\ell} P'$ then $\exists Q' \,.\, A \vdash Q \xrightarrow{\ell} Q' \wedge P' \mathrel{\dot{\sim}_{A \cup \mathrm{fn}(\ell)}} Q'$.

We do not develop other equivalences in this paper, but linear-time notions can also be defined straightforwardly. For example, for partial traces write

$$A_1 \vdash P_1 \xrightarrow{\ell_1} \ldots \xrightarrow{\ell_n} P_{n+1}$$

to mean $\exists P_2, \ldots, P_n, A_2, \ldots, A_n \;.\; \forall i \in 1..n \;.\; A_{i+1} = A_i \cup \mathrm{fn}(\ell_i) \wedge A_i \vdash P_i \xrightarrow{\ell_i} P_{i+1}$. If $\mathrm{fn}(P) \subseteq A$ then the partial A-traces of P are simply $\{\, \ell_1 \mathrel{..} \ell_n \mid \exists P' \;.\; A \vdash P \xrightarrow{\ell_1} \ldots \xrightarrow{\ell_n} P' \,\}$.

$$\textsc{Out}\ \frac{}{\overline{x}v.P \xrightarrow{\overline{x}v} P} \qquad \textsc{In}\ \frac{}{xp.P \xrightarrow{xv} \{^v/_p\}P}$$

$$\textsc{Tau}\ \frac{}{\tau.P \xrightarrow{\tau} P} \qquad \textsc{Sum}\ \frac{P \xrightarrow{\mu} P'}{P+Q \xrightarrow{\mu} P'}$$

$$\textsc{Par}\ \frac{P \xrightarrow{\mu} P' \quad \mathrm{bn}(\mu) \cap \mathrm{fn}(Q) = \emptyset}{P \mid Q \xrightarrow{\mu} P' \mid Q} \qquad \textsc{Com}\ \frac{P \xrightarrow{\overline{x}v} P' \quad Q \xrightarrow{xv} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\textsc{Close}\ \frac{P \xrightarrow{\overline{x}(v)} P' \quad Q \xrightarrow{xv} Q' \quad v \notin \mathrm{fn}(Q)}{P \mid Q \xrightarrow{\tau} (\boldsymbol{\nu} v)(P' \mid Q')}$$

$$\textsc{Res}\ \frac{P \xrightarrow{\mu} P' \quad x \notin \mathrm{n}(\mu)}{(\boldsymbol{\nu} x)P \xrightarrow{\mu} (\boldsymbol{\nu} x)P'} \qquad \textsc{Open}\ \frac{P \xrightarrow{\overline{y}x} P' \quad x \neq y}{(\boldsymbol{\nu} x)P \xrightarrow{\overline{y}(x)} P'}$$

$$\textsc{Match}\ \frac{P \xrightarrow{\mu} P'}{[x=x]P \xrightarrow{\mu} P'} \qquad \textsc{Mismatch}\ \frac{P \xrightarrow{\mu} P' \quad x \neq y}{[x \neq y]P \xrightarrow{\mu} P'}$$

Symmetric versions of Par, Com, Close and Sum are elided.

Figure 2: π conventional early operational semantics

Conventional Early Semantics

We conclude this section by recalling the 'conventional' operational semantics, without explicit indexing sets, in the original style of [47, 41], and showing that the two early bisimulation relations coincide. The conventional labelled transition relation has the form

$$P \xrightarrow{\mu} Q$$

where the label μ is taken from

$$\mu ::= \overline{x}v \;\big|\; \overline{x}(v) \;\big|\; xv \;\big|\; \tau$$

In the absence of an explicit index set outputs of old and new names must be distinguished in the label – as $\overline{x}v$ and $\overline{x}(v)$ respectively. The transitions are defined in Figure 2, which uses the following auxiliary functions for the free, 'bound' and all names of a label. Note that (despite the $\mathrm{bn}(\_)$ notation) the semantics does not involve alpha conversion of labels.

μ	$\mathrm{fn}(\mu)$	$\mathrm{bn}(\mu)$	$\mathrm{n}(\mu)$
$\overline{x}v$	$\{x,v\}$	$\{\}$	$\{x,v\}$
$\overline{x}(v)$	$\{x\}$	$\{v\}$	$\{x,v\}$
xv	$\{x,v\}$	$\{\}$	$\{x,v\}$
τ	$\{\}$	$\{\}$	$\{\}$

Again following [47, 41], take *conventional early bisimulation* $\dot{\sim}_{\mathrm{con}}$ to be the largest symmetric relation over terms such that each for all $P \mathrel{\dot{\sim}_{\mathrm{con}}} Q$,

- if $P \xrightarrow{\mu} P'$ and $\mathrm{bn}(\mu) \cap \mathrm{fn}(P,Q) = \emptyset$ then $\exists Q' \,.\, Q \xrightarrow{\mu} Q' \wedge P' \mathrel{\dot{\sim}_{\mathrm{con}}} Q'$.

Note that the conventional LTS includes pathological transitions such as $\overline{x}v.0 + (\boldsymbol{\nu} v)\overline{x}v.P \xrightarrow{\overline{x}(v)} P$, in which a name v both occurs free in the left hand side and is output as new. The condition $\mathrm{bn}(\mu) \cap \mathrm{fn}(P,Q) = \emptyset$ in the definition of bisimulation disregards such transitions, among others.

Lemma 2.1 *If* $P \xrightarrow{\mu} Q$ *then* $\mathrm{fn}(Q) \subseteq \mathrm{fn}(P) \cup \mathrm{n}(\mu)$.

Lemma 2.2 *If* $P \dot{\sim}_A P'$ *then* $P \dot{\sim}_{\mathrm{fn}(P,P')} P'$.

Lemma 2.3 *If* $A \supseteq \mathrm{fn}(P)$ *then*

1. $A \vdash P \xrightarrow{\overline{x}v} Q \wedge v \in A$ *iff* $P \xrightarrow{\overline{x}v} Q$
2. $A \vdash P \xrightarrow{\overline{x}v} Q \wedge v \notin A$ *iff* $P \xrightarrow{\overline{x}(v)} Q \wedge v \notin A$
3. $A \vdash P \xrightarrow{xv} Q$ *iff* $P \xrightarrow{xv} Q$
4. $A \vdash P \xrightarrow{\tau} Q$ *iff* $P \xrightarrow{\tau} Q$

Theorem 2.4 *If* $A \supseteq \mathrm{fn}(P,Q)$ *then* $P \dot{\sim}_A Q$ *iff* $P \dot{\sim}_{\mathrm{con}} Q$.

Proof: We check $\mathcal{R} = \{\, P, P' \mid \exists A \,.\, A \supseteq \mathrm{fn}(P,P') \wedge P \dot{\sim}_A P' \,\}$ and $\mathcal{R}_A = \{\, P, P' \mid A \supseteq \mathrm{fn}(P,P') \wedge P \dot{\sim}_{\mathrm{con}} P' \,\}$ are bisimulations of the two forms. The first is straightforward using Lemmas 2.3 and 2.2. The second is straightforward using Lemmas 2.3 and 2.1.

□

3 $\mathcal{N}$-LTS

In this section we introduce Indexed Labelled Transition Systems. To account for name substitution of π-terms, we take an indexing structure of name-sets and renaming functions on the set of states. We then axiomatize the key properties of the transition relation with respect to this indexing structure. We have also considered other choices of indexing structure, as briefly discussed in Section 7.

Definition 3.1 *Take* $\mathcal{N}$ *to be the category with objects finite subsets of* N *and arrows functions* $f : A \to B$ *between them.*

As before, given the fixed name set N, we define the set of π-labels as follows:

Definition 3.2 *Define* Lab *to be the set* $\{\tau\} \cup \{xy \mid x,y \in N\} \cup \{\overline{x}y \mid x,y \in N\}$

NOTATION:

- If $f : A \to B$ and $g : A' \to B'$ are two functions we write $f + g$ for the obvious function $A \uplus A' \to B \uplus B'$. If $f : A \to B$ and $g : A' \to B$ we write $[f,g]$ for the obvious copairing function $A \uplus A' \to B$.
- Given two names x and y, write $[y/x]$ for the unique function $\{x\} \to \{y\}$. If $f : A \to B$ is a function with $x \notin A$ and $y \in B$, by abuse of notation, we write $[f,[y/x]]$ for the obvious function $A, x \to B$.

- If $S : \mathcal{N} \to \mathbf{Set}$ is a functor and if $\coprod_{A\in|\mathcal{N}|} S(A)$ is the disjoint union of the sets $S(A)$ for objects A of $\mathcal{N}$, write $\langle A, s\rangle$ for the element $s \in S(A)$ as an element of the disjoint union and S for the set $\coprod_{A\in|\mathcal{N}|} S(A)$ itself.
- If $\longrightarrow \subseteq \mathsf{S} \times \mathrm{Lab} \times \mathsf{S}$ is a (transition) relation, we will write $A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t}$ to mean that there exists an $s \in S(A)$, a set B and a $t \in S(B)$ such that $\mathsf{s} = \langle A, s\rangle$, $\mathsf{t} = \langle B, t\rangle$ and $\mathsf{s} \xrightarrow{\ell} \mathsf{t}$. Sometimes we want to make explicit the existence of B and write $A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t} \dashv B$ to this purpose. Also if $f : A \to B$ is a function, write $f\mathsf{s}$ for $\langle B, S(f)(s)\rangle$.
- If $S, S' : \mathcal{N} \to \mathbf{Set}$ are two functors, $\mathsf{s} = \langle A, s\rangle \in \mathsf{S}$ and $\alpha : S \Longrightarrow S'$ is a natural transformation, we write $\alpha\mathsf{s}$ for $\langle A, \alpha_A s\rangle \in \mathsf{S}'$.

Definition 3.3 *For any label* $\ell \in \mathrm{Lab}$, *define the* channel names *of* ℓ, $\mathrm{chan}(\ell)$ *and the* value names *of* ℓ, $\mathrm{val}(\ell)$ *as follows:*

$$\begin{array}{rcl@{\qquad}rcl} \mathrm{chan}(\tau) & = & \emptyset & \mathrm{val}(\tau) & = & \emptyset \\ \mathrm{chan}(\overline{x}y) & = & \{x\} & \mathrm{val}(\overline{x}y) & = & \{y\} \\ \mathrm{chan}(xy) & = & \{x\} & \mathrm{val}(xy) & = & \{y\} \end{array}$$

Definition 3.4 *Define an Indexed Labelled Transition System ($\mathcal{N}$-LTS) to be a structure*

$$T = \langle S : \mathcal{N} \to \mathbf{Set},\ \longrightarrow,\ \mathsf{i}\rangle$$

where $\mathsf{i} = \langle I, i\rangle \in \mathsf{S}$, $\longrightarrow \subseteq \mathsf{S} \times \mathrm{Lab} \times \mathsf{S}$ *and the following conditions hold.*

1. *(Naming)* $A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t} \dashv B \implies \mathrm{chan}(\ell) \subseteq A \wedge B = A \cup \mathrm{fn}(\ell)$
2. (a) *(Input – new)* $A \vdash \mathsf{s} \xrightarrow{xy} \mathsf{t} \dashv A, y \implies \forall z \in A.\, A \vdash \mathsf{s} \xrightarrow{xz} [1_A, [z/y]]\mathsf{t}$
   (b) *(Input – old)* $A \vdash \mathsf{s} \xrightarrow{xy} \mathsf{t} \dashv A \implies \forall z \notin A \exists \mathsf{t_z}.\, A \vdash \mathsf{s} \xrightarrow{xz} \mathsf{t_z} \dashv A, z \wedge \mathsf{t} = [1_A, [y/z]]\mathsf{t_z}$
3. (a) *(Injective substitution)*
   For $f : A \to_{inj} B$, $A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t} \wedge g : (\mathrm{fn}(\ell) \setminus A) \to_{bij} \hat{B} \wedge \hat{B} \cap B = \emptyset \implies f\mathsf{s} \xrightarrow{(f+g)\ell} (f+g)\mathsf{t}$
   (b) *(Shifting)*
   $A \vdash \mathsf{s} \xrightarrow{xy} \mathsf{t} \dashv A, y \implies A, y \vdash \iota\mathsf{s} \xrightarrow{xy} \mathsf{t}$, *where* $\iota : A \hookrightarrow A, y$ *is the set inclusion function*
4. *For* $f : A \to_{inj} B$, *if* $f\mathsf{s} \xrightarrow{\ell'} \mathsf{t}'$ *then one of the following two cases applies*
   (a) *there exist* ℓ, t, $g : \mathrm{fn}(\ell) \setminus A \to_{bij} \hat{B}$ *such that* $\hat{B} \cap B = \emptyset$ *and* $\ell' = (f+g)(\ell)$ *and* $\mathsf{s} \xrightarrow{\ell} \mathsf{t}$ *and* $\mathsf{t}' = (f+g)\mathsf{t}$
   (b) *there exist* $x \in A$, $y \notin A$, $z \in B \setminus ran(f)$ *and* t *such that* $\ell' = f(x)z$ *and* $A \vdash \mathsf{s} \xrightarrow{xy} \mathsf{t} \dashv A, y$ *and* $\mathsf{t}' = [f, [z/y]]\mathsf{t}$

Condition 1 ensures that communication with the environment occurs only along publicly known channels and that the knowledge of such channels is correctly propagated from one state to another when a transition occurs. Conditions 2 ensure that if a name can be received as input along a specific channel, then any other name can be received as well. Condition 3a asserts that transitions are preserved along injective renamings, while condition 3b shows how inputs of new names generate inputs of "old" names when moving from a name set to a larger one. Finally, condition 4 ensures that the transitions out of a state which has been injectively renamed are determined by those of the state itself (cf. the example properties of Section 2). Clause 4b differs from that stated in [11] by requiring $z \notin \mathrm{ran}(f)$ – in the presence of the other axioms the two are equivalent, but this version is more elegant and supports the definition of $\mathcal{N}_{\mathrm{inj}}$-LTS in Section 7.

In fact the definition contains some redundancy:

Proposition 3.5 *Condition 3b 'shifting' is implied by conditions 2a 'input-new' and 3a 'injective substitution'.*

Proof: Suppose that $A \vdash \mathsf{s} \xrightarrow{xy} \mathsf{t} \dashv A, y$, then by condition 3a, $A, y \vdash \iota s \xrightarrow{xz} (\iota{+}[z/y])\mathsf{t} \dashv A, y, z$, for any $z \notin A, y$. Thus by condition 2a we deduce that $A, y \vdash \iota\mathsf{s} \xrightarrow{xy} [1_{A,y}, [y/z]](\iota{+}[z/y])\mathsf{t} \dashv A, y$. But $[1_{A,y}, [y/z]](\iota + [z/y]) = [1_{A,y}\iota, [y/z][z/y]] = [\iota, 1_{\{y\}}] = 1_{A,y}$ and therefore $A, y \vdash \iota\mathsf{s} \xrightarrow{xy} \mathsf{t}$.

□

Despite this we keep condition 3b, for two reasons. Firstly, we regard the condition as conceptually important, thus we did not want to omit it from the main definition. Secondly, conditions 2a and 2b, introduced to ensure uniform behaviour of input transitions, can be argued to be unnecessary from the model-theoretic point of view (just as their analogues are neglected in the reduction of value-passing CCS to pure CCS [39]). When 2a and 2b are omitted, 3b becomes essential.

For illustrative purposes we list now a few simple consequences of Definition 3.4. Analogous properties of π-terms are often established as lemmas, e.g. to prove correspondence between labelled and reduction semantics (see [53, 52] for explicitly-indexed developments).

Proposition 3.6 (Weakening) *If* $A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t}$ *and* $x \notin A \cup \mathrm{fn}(\ell)$ *then* $\iota\mathsf{s} \xrightarrow{\ell} \jmath\mathsf{t}$*, where* $\iota : A \hookrightarrow A, x$ *and* $\jmath : A \cup \mathrm{fn}(\ell) \hookrightarrow (A \cup \mathrm{fn}(\ell)), x$.

Proof: Suppose that $A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t}$ and $x \notin A \cup \mathrm{fn}(\ell)$ for some states s and t. Observe, first of all that by condition 3a $A, x \vdash \iota\mathsf{s} \xrightarrow{\ell} (\iota + g)\mathsf{t}$, for any $g : \mathrm{fn}(\ell) \setminus A \to_{\mathrm{bij}} B$, with $B \cap (A, x) = \emptyset$. Choose then g to be $1_{\mathrm{fn}(\ell)\setminus A}$. Then $\jmath = \iota + g$ and so the required property is satisfied.

□

Proposition 3.7 (Strengthening) *If* $A, x \vdash \iota\mathsf{s} \xrightarrow{\ell} \mathsf{t}'$*, and* $x \notin \mathrm{fn}(\ell)$*, where* $\iota : A \hookrightarrow A, x$*, then there exists* t *such that* $A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t}$ *and* $\mathsf{t}' = \jmath\mathsf{t}$*, where* $\jmath : A \cup \mathrm{fn}(\ell) \hookrightarrow (A \cup \mathrm{fn}(\ell)), x$.

Proof: Suppose that $A, x \vdash \iota\mathsf{s} \xrightarrow{\ell} \mathsf{t}'$ and $x \notin \mathrm{fn}(\ell)$ for some states s and t'. By condition 4, there can be two possibilities. It might be that there exist $\bar{\ell}$, t and $g : \mathrm{fn}(\bar{\ell}) \setminus A \to_{\mathrm{bij}} B$ such that $B \cap A, x = \emptyset$, $\ell = (\iota + g)(\bar{\ell})$, $\mathsf{s} \xrightarrow{\bar{\ell}} \mathsf{t}$ and $\mathsf{t}' = (\iota + g)\mathsf{t}$. Because of the properties above and by condition 3a, $\mathsf{s} \xrightarrow{\ell} (1_A + g)\mathsf{t}$. Moreover $\jmath = (\iota + g)(1_A + g^{-1})$, thus $\jmath(1_A + g)\mathsf{t} = (\iota + g)(1_A + g^{-1})(1_A + g)\mathsf{t} = (\iota + g)\mathsf{t} = \mathsf{t}'$.

Otherwise, there exist $x' \in A$, $y \notin A$, $z \in A, x$ and $\bar{\mathsf{t}}$ such that $\ell = x'z$, $A \vdash \mathsf{s} \xrightarrow{x'y} \bar{\mathsf{t}}$ and $\mathsf{t}' = [\iota, [z, y]]\bar{t}$. Now, since $x \notin \mathrm{fn}(\ell)$, then it must be that $z \in A$. So by condition 2a, $A \vdash \mathsf{s} \xrightarrow{x'z} [1_A, [z/y]]\bar{\mathsf{t}}$. Moreover $\jmath = \iota$ and $[\iota, [z/y]] = \iota[1_A, [z/y]]$. Thus, taking $\mathsf{t} = [1_A, [z/y]]\bar{\mathsf{t}}$, gives us $\jmath\mathsf{t} = \mathsf{t}'$.

□

Proposition 3.8 (Converse of Shifting) *If* $A, y \vdash \iota\mathsf{s} \xrightarrow{xy} \mathsf{t}$*, where* $\iota : A \hookrightarrow A, y$*, then* $A \vdash \mathsf{s} \xrightarrow{xy} \mathsf{t}$.

Proof: Suppose $A, y \vdash \iota \mathsf{s} \xrightarrow{xy} \mathsf{t}$, then either condition 4a or condition 4b must apply. Since y is not in A, 4a does not apply. Thus, by condition 4b there exist $\hat{y} \notin A$ and $\hat{\mathsf{t}}$, such that $A \vdash \mathsf{s} \xrightarrow{x\hat{y}} \hat{\mathsf{t}}$ and $\mathsf{t} = [\iota, [y/\hat{y}]]\hat{\mathsf{t}}$. By condition 3a, we then have that $A \vdash \mathsf{s} \xrightarrow{xy} (1_A + [y/\hat{y}])\hat{\mathsf{t}}$. But $1_A + [y/\hat{y}] = [\iota, [y/\hat{y}]]$, hence $A \vdash \mathsf{s} \xrightarrow{xy} \mathsf{t}$.

□

The terms of the π-calculus can be easily structured into an indexed sets of states, leading to the interpretation of π-terms as $\mathcal{N}$-LTSs all having the same set of states and transition relation, but different initial state.

Definition 3.9 *For every π-term P with free names in A and function $f : A \to B$, write fP for the π-term obtained by simultaneously substituting (avoiding capturing) $f(x)$ for x in P, for every $x \in A$.*

Definition 3.10 *Define $\pi : \mathcal{N} \to \mathbf{Set}$ to be the following functor:*

$$\begin{array}{rcl} \pi(A) & = & \{P \mid P \textit{ is a } \pi\textit{-term and } \mathrm{fn}(P) \subseteq A\} \\ \pi(f)(P) & = & fP \ . \end{array}$$

Definition 3.11 *For every π-term P with free names in I, define*

$$(\!|P|\!)_I = \langle \pi, \longrightarrow, \langle I, P \rangle \rangle$$

where $\langle A, Q \rangle \xrightarrow{\ell} \langle B, R \rangle$ if and only if $A \vdash Q \xrightarrow{\ell} R$ according to the operational semantics of Figure 1 and $B = A \cup \mathrm{fn}(\ell)$.

It is simple to verify using the operational semantics that $(\!|P|\!)_I$ is a $\mathcal{N}$-LTS.

In the next section we will provide a compositional semantics of π-terms as $\mathcal{N}$-LTSs. Before doing so, we conclude this section by structuring the class of $\mathcal{N}$-LTSs into a category and prove some of its main completeness and structural properties.

Definition 3.12 *Define a morphism $\alpha : T_1 \to T_2$ between $\mathcal{N}$-LTSs, with initial state over the same name-set I, to be a natural transformation $\alpha : S_1 \Longrightarrow S_2$ such that*

1. $\alpha \mathsf{i}_1 = \mathsf{i}_2$
2. $\mathsf{s} \xrightarrow{\ell} \mathsf{s}'$ *implies* $\alpha\mathsf{s} \xrightarrow{\ell} \alpha\mathsf{s}'$.

Define $\mathcal{N}$-$\mathbf{LTS}_I$ to be the category of $\mathcal{N}$-LTSs with initial name-set I and these morphisms.

Theorem 3.13 *For every name-set I, the category $\mathcal{N}$-$\mathbf{LTS}_I$ is complete and cocomplete.*

Proof:
Completeness By a well-known result of category theory [36], it suffices to show that $\mathcal{N}$-$\mathbf{LTS}_I$ has equalisers and small products.

<u>Equalisers</u>: If $T_1 \underset{\beta}{\overset{\alpha}{\rightrightarrows}} T_2$ are two morphisms in $\mathcal{N}$-$\mathbf{LTS}_I$, define $T_0 = \langle S_0, \longrightarrow_0, \mathsf{i}_1 \rangle$ and $\gamma : T_0 \hookrightarrow T_1$ as follows:

- $S_0 \xrightarrow{\gamma} S_1 \underset{\beta}{\overset{\alpha}{\rightrightarrows}} S_2$ is the equaliser in $\mathbf{Set}^{\mathcal{N}}$, defined as $S_0(A) = \{s \in S_1(A) \mid \alpha_A(s) = \beta_A(s)\}$, while γ_A is the inclusion function.

- $\longrightarrow_0$ is the largest subset of $\mathsf{S}_0 \times \mathrm{Lab} \times \mathsf{S}_0$ which satisfies conditions 2b and 4 of Definition 3.4 and is such that
$$A \vdash \mathsf{s} \xrightarrow{\ell}_0 \mathsf{t} \dashv B \implies A \vdash \gamma_A \mathsf{s} \xrightarrow{\ell}_1 \gamma_B \mathsf{t} \dashv B.$$

It is a matter of easy verification to check that $\longrightarrow_0$ is well-defined, i.e. that such a largest subset exists as the partial order of transition relations in $\mathsf{S}_0 \times \mathrm{Lab} \times \mathsf{S}_0$ satisfying conditions 2b and 4 is a complete lattice. Moreover all the other conditions of Definition 3.4 are met as a consequence of the fact that $\longrightarrow_0$ is essentially a subrelation of $\longrightarrow_1$. The universal property follows from the fact that the diagram $S_0 \xrightarrow{\gamma} S_1 \overset{\alpha}{\underset{\beta}{\rightrightarrows}} S_2$ is an equaliser in $\mathbf{Set}^{\mathcal{N}}$.

<u>Products</u>: Let $(T_k)_{k\in K}$ be a family of transition systems. Define their product

$$T = \langle S, \longrightarrow, \langle I, \langle i_k \rangle_{k\in K}\rangle\rangle$$

and $(\pi_k : T \to T_k)_{k\in K}$ as follows:

- $(\pi_k : S \to S_k)_{k\in K}$ is the product in $\mathbf{Set}^{\mathcal{N}}$ of the S_k's. That is, $S(A) = \{\langle s_k\rangle_{k\in K} \mid s_k \in S_k(A)\}$, with $S(f)$ and π_k defined in the obvious way.
- The transition relation is defined as $A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t}$ iff for every k, $A \vdash \pi_k \mathsf{s} \xrightarrow{\ell} \pi_k \mathsf{t}$.

It is very easy to verify that $(\pi_k : T \to T_k)_{k\in K}$ satisfies the universal property of products.

The terminal object, i.e. the empty product is given by, $\langle \mathbf{1}, \longrightarrow, \langle I, \star\rangle\rangle$, where for every A, $\mathbf{1}(A) = \{\star\}$ and for every A and ℓ, $\langle A, \star\rangle \xrightarrow{\ell} \langle A \cup \mathrm{fn}(\ell), \star\rangle$.

Cocompleteness Similarly, it suffices to show that $\mathcal{N}$-$\mathbf{LTS}_I$ has coequalisers and small coproducts.

<u>Coequalisers</u>: If $T_1 \overset{\alpha}{\underset{\beta}{\rightrightarrows}} T_2$ are two morphisms in $\mathcal{N}$-$\mathbf{LTS}_I$, define $T_0 = \langle S_0, \longrightarrow_0, \mathsf{i}_0\rangle$ and $\gamma : T_2 \to T_0$ as follows:

- $S_1 \overset{\alpha}{\underset{\beta}{\rightrightarrows}} S_2 \xrightarrow{\gamma} S_0$ is a coequaliser in $\mathbf{Set}^{\mathcal{N}}$.
- $\longrightarrow_0$ is defined as $\mathsf{s} \xrightarrow{\ell}_0 \mathsf{t}$ if there exists $\bar{\mathsf{s}}$ and $\bar{\mathsf{t}}$ such that $\gamma\bar{\mathsf{s}} = \mathsf{s}$, $\gamma\bar{\mathsf{t}} = \mathsf{t}$ and $\bar{\mathsf{s}} \xrightarrow{\ell}_2 \bar{\mathsf{t}}$.
- $\mathsf{i}_0 \stackrel{\text{def}}{=} \gamma \mathsf{i}_2$.

Naturality of γ ensures that T_0 is an $\mathcal{N}$-LTS and by the definition of $\longrightarrow_0$, we immediately also have that γ is a morphism in $\mathcal{N}$-$\mathbf{LTS}_I$. The universal property follows from that of $S_1 \overset{\alpha}{\underset{\beta}{\rightrightarrows}} S_2 \xrightarrow{\gamma} S_0$.

<u>Coproducts</u>: The initial object is $0_I = \langle \mathcal{N}(I, -), \emptyset, \langle I, 1_I\rangle\rangle$, where $\mathcal{N}(I, -)$ is the *representable* functor [37], $\mathcal{N}(I,-)(A) \stackrel{\text{def}}{=} \mathcal{N}(I, A) = \{f \mid f : I \to A\}$, $\mathcal{N}(I,-)(g)(f) \stackrel{\text{def}}{=} \mathcal{N}(I, g)(f) = gf$, for any $g : A \to B$ and $f : I \to A$ in $\mathcal{N}$. Observe in fact that for any functor $S : \mathcal{N} \to \mathbf{Set}$ and element $i \in S(I)$, there exists a unique natural transformation $0_S : \mathcal{N}(I, -) \to S$ such that $0_{S,I}(1_I) = i$.

Suppose now that $(T_k)_{k\in K}$ is a family of $\mathcal{N}$-LTS's. Define $T = \langle S, \longrightarrow, \mathsf{i}\rangle$ and $(\mathrm{in}_k : T_k \to T)_{k\in K}$ to be

- $(\mathrm{in}_k : S_k \to S)_{k\in K}$ a colimit in $\mathbf{Set}^{\mathcal{N}}$ of the wide-span diagram of vertex 0_I and edges $(0_{S_k})_{k\in K}$

- $\mathsf{i} = \mathrm{in}_k \mathsf{i}_k$ (notice that it does not matter which k is chosen) and $\mathsf{s} \xrightarrow{\ell} \mathsf{t}$ if there exists k, $\bar{\mathsf{s}}$ and $\bar{\mathsf{t}}$ such that $\mathrm{in}_k \bar{\mathsf{s}} = \mathsf{s}$, $\mathrm{in}_k \bar{\mathsf{t}} = \mathsf{t}$ and $\bar{\mathsf{s}} \xrightarrow{\ell}_k \bar{\mathsf{t}}$.

□

Every reindexing function $f : A \to B$ induces an obvious functor $R(f) : \mathcal{N}\text{-}\mathbf{LTS}_A \to \mathcal{N}\text{-}\mathbf{LTS}_B$ that reindexes the initial state according to f.

Definition 3.14 *Define the reindexing functor $R : \mathcal{N} \to \mathbf{CAT}$ as follows: for every name-set A, $R(A) = \mathcal{N}\text{-}\mathbf{LTS}_A$, while for every function $f : A \to B$, $R(f)$ is the functor*

$$\begin{aligned} R(f)(\langle S, \longrightarrow, \mathsf{i}\rangle) &= \langle S, \longrightarrow, f\mathsf{i}\rangle \\ R(f)(\alpha) &= \alpha \ . \end{aligned}$$

This allows us to employ the Grothendieck construction [29] to produce a category $\mathcal{N}\text{-}\mathbf{LTS}$ cofibred over $\mathcal{N}$. In more concrete terms we have the following:

Definition 3.15 *Define $\mathcal{N}\text{-}\mathbf{LTS}$ to be the following category:*

Objects: *$\mathcal{N}$-LTSs*

Arrows: *$\langle \alpha, f\rangle : \langle S, \longrightarrow, \langle A, i\rangle\rangle \to \langle S', \longrightarrow', \langle B, i'\rangle\rangle$ is an arrow if $f : A \to B$ is a function and $\alpha : \langle S, \longrightarrow, \langle B, S(f)(i)\rangle\rangle \to \langle S', \longrightarrow', \langle B, i'\rangle\rangle$ is a morphism in $\mathcal{N}\text{-}\mathbf{LTS}_B$ (see Definition 3.12).*

The composition is defined using the reindexing functor (that in this case have the only effect of providing the right type for composing arrows), i.e.

$$\langle \beta, g\rangle \langle \alpha, f\rangle \stackrel{\text{def}}{=} \langle \beta R(f)(\alpha), gf\rangle \ .$$

By construction the obvious projection functor $\mathcal{N}\text{-}\mathbf{LTS} \to \mathcal{N}$ which sends a $\mathcal{N}$-LTS to its initial name-set and any arrow $\langle \alpha, f\rangle$ to f is a cofibration [29]. The category $\mathcal{N}$ is equivalent to the category of finite sets, which is known to have all finite colimits. By a well-known result of fibred category theory [29] we can then immediately deduce that $\mathcal{N}\text{-}\mathbf{LTS}$ has finite colimits as well.

4 Denotational semantics

We describe now operations on $\mathcal{N}$-LTS that we will use in giving a compositional semantics to the π-calculus. It will be straightforward to turn the operations below into functors, in fact into ω-continuous functors. As we have shown, for every name-set I, the category $\mathcal{N}\text{-}\mathbf{LTS}_I$ is cocomplete and therefore has colimits of ω-chains. Thus a semantics of recursively defined processes, such as replicated ones, can be obtained using initial algebras in the usual way [58].

The most interesting operations are deadlock, which to obtain initiality has what may be a slightly surprising definition, and restriction and parallel composition. For restriction an equivalence relation, a semantic analogue of α-conversion, needs to be imposed on states – just as in the operational semantics a transition of $(\boldsymbol{\nu} x)P$ may be derived from a transition of $(\boldsymbol{\nu}\hat{x})\{^{\hat{x}}/_{x}\}P$ for any $\hat{x} \notin (\mathrm{fn}(P) \setminus x)$. For parallel, in the operational semantics states reachable by transitions from $P \mid Q$ may involve restriction of $P' \mid Q'$ for P', Q' reachable from P, Q. The construction over the model involves a similar quotienting as for restriction. The equivalence relation used in both cases is defined as follows.

Definition 4.1 *If $S : \mathcal{N} \to \mathbf{Set}$ is a functor and A is a finite subset of N, take $\leftrightarrow_A$ to be the equivalence relation on (possibly subsets of) the set $\coprod_{B \supseteq A} S(B)$ defined by $\langle B_1, s_1\rangle \leftrightarrow_A \langle B_2, s_2\rangle$ if there exists a bijection $b : B_1 \to_{bij} B_2$, such that for every $x \in A$, $b(x) = x$ and such that $S(b)(s_1) = s_2$.*

Observe that elements of $S(A)$ can only be related to themselves, i.e. their equivalence class is a singleton. For this reason, when no confusion arises, we will write s for $[\langle A,s\rangle]_{\leftrightarrow_A}$. Notice also that, because of naturality, any arrow $\alpha : T \to T'$ in $\mathcal{N}$-$\mathbf{LTS}_I$ preserves $\leftrightarrow_A$, i.e.

$$\langle B_1, s_1\rangle \leftrightarrow_A \langle B_2, s_2\rangle \text{ implies } \langle B_1, \alpha_{B_1}(s_1)\rangle \leftrightarrow_A \langle B_2, \alpha_{B_1}(s_2)\rangle ,$$

for any pair of states of T. Thus for any set A, the function $\alpha_{\leftrightarrow_A} : \coprod_{B\supseteq A} S(B) \to \coprod_{B\supseteq A} S'(B)$ given by $\alpha_{\leftrightarrow_A}([\langle B,s\rangle]_{\leftrightarrow_A}) \stackrel{\text{def}}{=} [\langle B, \alpha_B(s)\rangle]_{\leftrightarrow_A}$ is well-defined.

In the constructions below we shall often extend a transition system with new initial state over a chosen name set (say I), but now all of its reindexings must also be added. This can be expressed using the *representable* functor $\mathcal{N}(I,-)$ which we met in the proof of Theorem 3.13 and that provides the state space of the initial object of $\mathcal{N}$-$\mathbf{LTS}_I$. Notice that we write e.g. $S + \mathcal{N}(I,-)$ for the coproduct of functors which is given by the pointwise disjoint union of sets.

NOTATION: If U and V are two sets and $U \uplus V$ their disjoint union and no confusion arises, we will write $\mathsf{l} : U \to U \uplus V$ and $\mathsf{r} : V \to U \uplus V$ for the obvious left and right injections. If $\mathsf{s} = \langle A, u\rangle$, write ls for $\langle A, \mathsf{l}u\rangle$ and similarly for r. In what follows, unless otherwise stated we suppose that $T = \langle S : \mathcal{N} \to \mathbf{Set},\ \longrightarrow,\ \mathsf{i}\rangle$ and $T_k = \langle S_k : \mathcal{N} \to \mathbf{Set},\ \longrightarrow_k,\ \mathsf{i}_k\rangle$ (for $k = 1,2$) are $\mathcal{N}$-LTSs, with $\mathsf{i} = \langle I, i\rangle$ and $\mathsf{i}_k = \langle I, i_k\rangle$. Note that the initial name-sets I coincide.

Restriction If $T = \langle S : \mathcal{N} \to \mathbf{Set},\ \longrightarrow,\ \langle (I,x), i\rangle\rangle$ define the restriction $\nu_{x\in(I,x)}(T)$ to be

$$\langle S' : \mathcal{N} \to \mathbf{Set},\ \longrightarrow',\ \langle I, \mathsf{r}[\langle (I,x), i\rangle]_{\leftrightarrow_I}\rangle\rangle ,$$

where

- $S'(A) = S(A) \uplus (\coprod_{y\notin A} S(A,y))/\leftrightarrow_A$
- $\longrightarrow'$ is defined by the following three rules:

$$\frac{A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t}}{A \vdash \mathsf{ls} \xrightarrow{\ell}{}' \mathsf{lt}}$$

$$\frac{A, z \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t}}{A \vdash \mathsf{r}[\mathsf{s}]_{\leftrightarrow_A} \xrightarrow{\ell}{}' \mathsf{r}[\mathsf{t}]_{\leftrightarrow_{A\cup \mathrm{fn}(\ell)}}}\ z \notin \mathrm{fn}(\ell)$$

$$\frac{A, z \vdash \mathsf{s} \xrightarrow{\overline{x}z} \mathsf{t}}{A \vdash \mathsf{r}[\mathsf{s}]_{\leftrightarrow_A} \xrightarrow{\overline{x}z}{}' \mathsf{lt}}\ x \neq z$$

If $\alpha : T_1 \to T_2$ is an arrow in $\mathcal{N}$-$\mathbf{LTS}_{I,x}$, define $\nu_{x\in(I,x)}(\alpha)$ to be $\alpha + \alpha_{\leftrightarrow_{-}}$, where for every A, $\alpha_{\leftrightarrow_A}$ is restricted in this case to operate only on equivalence classes $[\langle B, s\rangle]$ where $B = A, y$ for some $y \notin A$.

Output and τ prefix If $x, y \in I$, define $\overline{x}y(T)$ to be

$$\langle S + \mathcal{N}(I,-) : \mathcal{N} \to \mathbf{Set},\ \longrightarrow',\ \langle I, \mathsf{r}1_I\rangle\rangle ,$$

where $\longrightarrow'$ is defined by the following rules:

$$\frac{f : I \to A}{\langle A, \mathsf{r}f\rangle \xrightarrow{\overline{f(x)}f(y)}{}' \mathsf{l}(f\mathsf{i})}$$

$$\frac{\mathsf{s} \xrightarrow{\ell} \mathsf{t}}{\mathsf{ls} \xrightarrow{\ell}{}' \mathsf{lt}}$$

Define $\tau(T)$ similarly by labelling the transition in the first rule τ rather than $\overline{f(x)}f(y)$.

If $\alpha : T_1 \to T_2$ is an arrow in $\mathcal{N}$-$\mathbf{LTS}_I$, define $\overline{x}y(\alpha)$ (and equally $\tau(\alpha)$)

Input prefix If $T = \langle S : \mathcal{N} \to \mathbf{Set},\ \longrightarrow,\ \langle (I, y), i\rangle\rangle$ is a transition system and $y \neq x \in I$, define $xy(T)$ to be

$$\langle S + \mathcal{N}(I, -) : \mathcal{N} \to \mathbf{Set},\ \longrightarrow',\ \langle I, \mathsf{r}1_I\rangle\rangle\ ,$$

where

$$\frac{f : I \to A \qquad \iota : A \hookrightarrow A \cup \{z\}}{\langle A, \mathsf{r}f\rangle \xrightarrow{f(x)z}{}' \langle A \cup \{z\}, \mathsf{l}S([\iota f, [z/y]])(i)\rangle}$$

$$\frac{\mathsf{s} \xrightarrow{\ell} \mathsf{t}}{\mathsf{ls} \xrightarrow{\ell}{}' \mathsf{lt}}$$

Deadlock at I For every set of names I, define the deadlock $\mathcal{N}$-LTS with free names in I as $0_I = \langle \mathcal{N}(I, -), \emptyset, \langle I, 1_I\rangle\rangle$. Recall, from the proof Theorem 3.13 that 0_I is the initial object of the category $\mathcal{N}$-$\mathbf{LTS}_I$.

Matching and Mismatching If $x, y \in I$, define $[x = y](T)$ to be

$$\langle S + \mathcal{N}(I, -) : \mathcal{N} \to \mathbf{Set},\ \longrightarrow',\ \langle I, \mathsf{r}1_I\rangle\rangle\ ,$$

where $\longrightarrow'$ is defined by the following rules:

$$\frac{f : I \to A \qquad f\mathsf{i} \xrightarrow{\ell} \mathsf{s} \qquad f(x) = f(y)}{\langle A, \mathsf{r}f\rangle \xrightarrow{\ell}{}' \mathsf{ls}}$$

$$\frac{\mathsf{s} \xrightarrow{\ell} \mathsf{t}}{\mathsf{ls} \xrightarrow{\ell}{}' \mathsf{lt}}$$

Define $[x \neq y](T)$ similarly by requiring $f(x) \neq f(y)$ in the first rule.

Sum Define the sum,

$$T_1 \oplus T_2 = \langle (S_1 + \mathcal{N}(I, -) + S_2) : \mathcal{N} \to \mathbf{Set},\ \longrightarrow_0,\ \langle I, \mathsf{m}1_I\rangle\rangle$$

where $\longrightarrow_0$ is defined by the following rules:

$$\frac{f : I \to A \quad \langle A, S_k(f)(i_1)\rangle \xrightarrow{\ell}_1 \mathsf{s}}{\langle A, \mathsf{m}f\rangle \xrightarrow{\ell}_0 \mathsf{ls}} \text{(and sym. for } 2, \mathsf{r})$$

$$\frac{\mathsf{s} \xrightarrow{\ell}_1 \mathsf{t}}{\mathsf{ls} \xrightarrow{\ell}_0 \mathsf{lt}} \qquad\qquad \frac{\mathsf{s} \xrightarrow{\ell}_2 \mathsf{t}}{\mathsf{rs} \xrightarrow{\ell}_0 \mathsf{rt}}$$

where we now use $\mathsf{l}, \mathsf{m}, \mathsf{r}$ rather than just l and r. If $\alpha : T_1 \to T_1'$ and $\beta : T_2 \to T_2'$ are two arrows, define $\alpha \oplus \beta = \alpha + 1_{\mathcal{N}(I,-)} + \beta$.

Remark: If we assume the $\mathcal{N}$-LTSs to be non-restarting, i.e. with no loops involving the initial state or any of its reindexings, we can simply use the categorical coproduct to model non-deterministic sum.

Parallel composition

NOTATION: If S_1 and S_2 are two functors $\mathcal{N} \to \mathbf{Set}$, and if $\leftrightarrow_A$ is the equivalence relation on $\coprod_{B \supseteq A}(S_1 \times S_2)(B) = \coprod_{B \supseteq A} S_1(B) \times S_2(B)$ defined as in Definition 4.1, and if $\mathsf{s}_1 = \langle B, s_1 \rangle$ and $\mathsf{s}_2 = \langle B, s_2 \rangle$ write $\mathsf{s}_1|_A\mathsf{s}_2$ for the equivalence class $[(B, s_1, s_2)]_{\leftrightarrow_A}$.

If T_1 and T_2 are two $\mathcal{N}$-LTSs as before, define their parallel composition,

$$T_1|T_2 = \langle S_0 : \mathcal{N} \to \mathbf{Set},\ \longrightarrow_0,\ \mathsf{i}_1|_I\mathsf{i}_2 \rangle$$

where

- $S_0(A) = (\coprod_{B \supseteq A}(S_1 \times S_2)(B))/\leftrightarrow_A$, while $S_0(f : A \to A')([(B, s_1, s_2)]_{\leftrightarrow_A}) = [(B', t_1, t_2)]_{\leftrightarrow_{A'}}$, where $t_k = S(f + g)(s_k)$, for $k = 1, 2$ and $g : B \setminus A \to_{\mathrm{bij}} B' \setminus A'$ is a bijection
- $\longrightarrow_0$ is defined by the following three rules (and symmetric versions of the first two):

$$\frac{A, A' \vdash \mathsf{s}_1 \xrightarrow{\ell}_1 \mathsf{t}_1 \qquad \iota : A, A' \hookrightarrow A \cup \mathrm{fn}(\ell), A'}{A \vdash \mathsf{s}_1|_A\mathsf{s}_2 \xrightarrow{\ell}_0 \mathsf{t}_1|_{A \cup \mathrm{fn}(\ell)}\iota\mathsf{s}_2}$$

$$\frac{A, A' \vdash \mathsf{s}_1 \xrightarrow{\overline{x}y}_1 \mathsf{t}_1 \qquad A, A' \vdash \mathsf{s}_2 \xrightarrow{xy}_2 \mathsf{t}_2}{A \vdash \mathsf{s}_1|_A\mathsf{s}_2 \xrightarrow{\tau}_0 \mathsf{t}_1|_A\mathsf{t}_2}$$

$$\frac{A, y \vdash \mathsf{s}_1|_{A,y}\mathsf{s}_2 \xrightarrow{\overline{x}y}_0 \mathsf{t}_1|_{A,y}\mathsf{t}_2}{A \vdash \mathsf{s}_1|_A\mathsf{s}_2 \xrightarrow{\overline{x}y}_0 \mathsf{t}_1|_{A,y}\mathsf{t}_2}$$

It is routine to verify that the functor S_0 is well defined, i.e. that the definition of $S_0(f)$ is independent of the choice of representatives and of the choice of the functions g.

If $\alpha : T_1 \to T_1'$ and $\beta : T_2 \to T_2'$ are two arrows, define $(\alpha|\beta)_A([\langle B, s_1, s_2 \rangle]_{\leftrightarrow_A}) = [\langle B, \alpha_B(s_1), \beta_B(s_2) \rangle]_{\leftrightarrow_A}$.

Compositional semantics to π-terms is given using the operations defined above in the usual way. For a process term P, with free names in I, we write $[\![P]\!]_I$ for the corresponding $\mathcal{N}$-LTS. So in particular we have

- $[\![0]\!]_I = 0_I$
- $[\![P \mid Q]\!]_I = [\![P]\!]_I \mid [\![Q]\!]_I$
- $[\![P + Q]\!]_I = [\![P]\!]_I \oplus [\![Q]\!]_I$
- $[\![\tau.P]\!]_I = \tau([\![P]\!]_I)$
- $[\![\overline{x}y.P]\!]_I = \overline{x}y([\![P]\!]_I)$
- $[\![xy.P]\!]_I = xy([\![P]\!]_I)$
- $[\![\nu x.P]\!]_I = (\boldsymbol{\nu} x)([\![P]\!]_{I,x})$
- $[\![[x = y]P]\!]_I = [x = y]([\![P]\!]_I)$
- $[\![[x \neq y]P]\!]_I = [x \neq y]([\![P]\!]_I)$

Bisimilarity is defined in the usual way, but thanks to the indexing, we can also define directly in the model the closure under name substitutions, which for the π-calculus characterises the largest congruence included in bisimilarity.

Definition 4.2 *Define two $\mathcal{N}$-LTSs T_1 and T_2 to be*

- *strongly bisimilar if the LTS $\langle \mathsf{S}_1, \longrightarrow_1, \mathsf{i}_1\rangle$ and $\langle \mathsf{S}_2, \longrightarrow_2, \mathsf{i}_2\rangle$ are bisimilar in the usual sense of Milner [39].*
- *strongly equivalent if, for every $f : I \to A$, the $\mathcal{N}$-LTSs $\langle S_1, \longrightarrow_1, f\mathsf{i}_1\rangle$ and $\langle S_2, \longrightarrow_2, f\mathsf{i}_2\rangle$ are strongly bisimilar.*

Weak bisimilarity and equivalence are defined similarly.

By taking full account of the indexing structure of the state space we naturally characterise 'open bisimilarity' [47].

Definition 4.3 *If $S_1, S_2 : \mathcal{N} \to \mathbf{Set}$ are functors, define a relation between S_1 and S_2 to be a subobject $R \hookrightarrow S_1 \times S_2$ of their product.*

In other words (cf. [37]) a relation R is an indexed set of pairs $(R(A))_{A\in|\mathcal{N}|}$ such that, for every A,

$$R(A) \subseteq S_1(A) \times S_2(A)$$

and for every $f : A \to B$,

$$\langle s_1, s_2\rangle \in R(A) \text{ implies } \langle S_1(f)(s_1), S_2(f)(s_2)\rangle \in R(B) \ .$$

Any relation $R \hookrightarrow S_1 \times S_2$, induces a relation

$$\mathsf{R} \subseteq \mathsf{S}_1 \times \mathsf{S}_2 \ ,$$

defined as $\mathsf{R} \stackrel{\text{def}}{=} \{\langle \mathsf{s}, \mathsf{t}\rangle \mid \exists A,\, s,\, t.\, \langle s, t\rangle \in R(A) \wedge \mathsf{s} = \langle A, s\rangle \wedge \mathsf{t} = \langle A, t\rangle\}$.

Definition 4.4 *If T_1 and T_2 are two $\mathcal{N}$-LTSs with indexed sets of states S_1 and S_2, respectively, define a relation $R \hookrightarrow S_1 \times S_2$ to be an open bisimulation if the relation*

$$\mathsf{R} \subseteq \mathsf{S}_1 \times \mathsf{S}_2$$

is a strong bisimulation for the transition systems $\langle \mathsf{S}_1, \longrightarrow_1, \mathsf{i}_1\rangle$ and $\langle \mathsf{S}_2, \longrightarrow_2, \mathsf{i}_2\rangle$.

Clearly open bisimilarity is the finest among all the equivalence relations defined above.

We conclude this section with the result which relates bisimulation in the model with bisimulation in the operational semantics.

Theorem 4.5 *Let P and Q be two π-terms with free names in I. Then $P \dot{\sim}_I Q$ if and only if $[\![P]\!]_I$ is bisimilar to $[\![Q]\!]_I$.*

To prove this theorem we first need the following lemma.

Lemma 4.6 *For every π-term P, with free names in I, $(\![P]\!)_I$ is open bisimilar to $[\![P]\!]_I$.*

Proof: The proof goes by structural induction on P. We show how to inductively define open bisimulations. It will be generally trivial to verify that the induced relations on the states of the transition systems are bisimulations. Thus we omit the verification in all but a few less trivially clear cases.

- $P = 0$ Define $\mathcal{B}(A) = \{\langle 0, f\rangle \mid f : I \to A\}$. For any $g : A \to B$, $\mathcal{B}(g)\langle 0, f\rangle = \langle 0, gf\rangle$. Since there are no transitions to match, this induces trivially a bisimulation.

- $P = \overline{x}y.Q$ Suppose $\mathcal{B} \subseteq (\!|Q|\!)_I \times [\![Q]\!]_I$ is an open bisimulation. Define

$$(\overline{x}y.\mathcal{B})(A) = \{\langle R, \mathsf{l}s\rangle \mid \langle R, s\rangle \in \mathcal{B}(A)\} + \{\langle \overline{(fx}fy.fQ), \mathsf{r}f\rangle \mid f : I \to A\}$$

If $g : A \to B$, $(\overline{x}y.\mathcal{B})(g)\langle R, \mathsf{l}s\rangle = \langle gR, \mathsf{l}[\![Q]\!]_I(g)(s)\rangle$, while $(\overline{x}y.\mathcal{B})(g)\langle \overline{(fx}fy.fQ), \mathsf{r}f\rangle = \langle \overline{((gf)x}(gf)y.(gf)Q), \mathsf{r}(gf)\rangle$.

- $P = \tau.Q$ and $P = xy.Q$ These cases are dealt with similarly to the output case.
- $P = [x = y]Q$ and $P = [x \neq y]Q$ Suppose $\mathcal{B} \subseteq (\!|Q|\!)_I \times [\![Q]\!]_I$ is an open bisimulation. Define

$$([x = y]\mathcal{B})(A) = \{\langle R, \mathsf{l}s\rangle \mid \langle R, s\rangle \in \mathcal{B}(A)\} + \{\langle R, \mathsf{r}f\rangle \mid f : I \to A \& R = ([fx = fy]fQ)\}$$

The case of mismatching has $R = ([fx \neq fy]fQ)\}$ in the definition of the second set above. The action on renaming functions is defined as in the case of the prefixes.

- $P = (\boldsymbol{\nu} x).Q$ Suppose $\mathcal{B} \subseteq (\!|Q|\!)_{I,x} \times [\![Q]\!]_{I,x}$ is an open bisimulation. Define

$$((\boldsymbol{\nu} x).\mathcal{B})(A) = \{\langle R, \mathsf{l}s\rangle \mid \langle R, s\rangle \in \mathcal{B}(A)\} + \bigcup_{y \in \mathcal{N}\setminus A} \{\langle (\boldsymbol{\nu} y).R, \mathsf{r}[s]_{\leftrightarrow A}\rangle \mid \langle R, s\rangle \in \mathcal{B}(A, y)\}$$

- $P = P_1 \mid P_2$ Suppose that $\mathcal{B} \subseteq (\!|P_1|\!)_I \times [\![P_1]\!]_I$ and $\mathcal{C} \subseteq (\!|P_2|\!)_I \times [\![P_2]\!]_I$ are open bisimulations. Define

$$(\mathcal{B} \mid \mathcal{C})(A) = \{\langle \nu\vec{x}(Q_1 \mid Q_2), \mathsf{s}_1 \mid_A \mathsf{s}_2\rangle \mid \langle Q_1, s_1\rangle \in \mathcal{B}(A, \vec{x}) \& \langle Q_2, s_2\rangle \in \mathcal{C}(A, \vec{x})\}$$

with the obvious action on renaming functions. Let us see in some detail why the relation on states induced by $\mathcal{B} \mid \mathcal{C}$ is a bisimulation. Suppose for instance that

$$\langle \nu\vec{x}(Q_1 \mid Q_2), \mathsf{s}_1 \mid_A \mathsf{s}_2\rangle \in (\mathcal{B} \mid \mathcal{C})(A)$$

with, without loss of generality, $\langle Q_1, s_1\rangle \in \mathcal{B}(A, \vec{x})$ and $\langle Q_2, s_2\rangle \in \mathcal{C}(A, \vec{x})$, and that $A \vdash \nu\vec{x}(Q_1 \mid Q_2) \xrightarrow{\ell} Q$. Then one of the following three cases (or one of their symmetric versions) must hold:

1. There exist $\vec{y}$, $\overline{Q}_1$ and $\overline{Q}_2$ such that
   - $\nu\vec{x}(Q_1 \mid Q_2) = \nu\vec{y}(\overline{Q}_1 \mid \overline{Q}_2)$
   - $\mathrm{fn}(\ell) \notin \vec{y}$
   - $A, \vec{y} \vdash \overline{Q}_1 \xrightarrow{\ell} Q_1'$
   - $Q = \nu\vec{y}(Q_1' \mid \overline{Q}_2)$.

   Then there exists a bijection $b : A, \vec{x} \to_{\mathrm{bij}} A, \vec{y}$ such that $b(a) = a$ for every $a \in A$, $bQ_1 = \overline{Q}_1$ and $bQ_2 = \overline{Q}_2$. Thus $\langle \overline{Q}_1, S_1(b)(s_1)\rangle \in \mathcal{B}(A, \vec{y})$ and $\langle \overline{Q}_2, S_2(b)(s_2)\rangle \in \mathcal{C}(A, \vec{y})$. So, by inductive hypothesis $A, \vec{y} \vdash b\mathsf{s}_1 \xrightarrow{\ell} t_1$ with $\langle Q_1', t_1\rangle \in \mathcal{B}(A \cup \mathrm{fn}(\ell), \vec{y})$. Hence

   $$A \vdash \mathsf{s}_1 \mid_A \mathsf{s}_2 = b\mathsf{s}_1 \mid_A b\mathsf{s}_2 \xrightarrow{\ell} \mathsf{t}_1 \mid_{A\cup\mathrm{fn}(\ell)} \iota b\mathsf{s}_2\ ,$$

   with $\langle Q, \mathsf{t}_1 \mid_{A\cup\mathrm{fn}(\ell)} \iota b\mathsf{s}_2\rangle \in (\mathcal{B} \mid \mathcal{C})(A \cup \mathrm{fn}(\ell))$.
2. There exist $\vec{y}$, $\overline{Q}_1$ and $\overline{Q}_2$ such that
   - $\nu\vec{x}(Q_1 \mid Q_2) = \nu\vec{y}(\overline{Q}_1 \mid \overline{Q}_2)$
   - $\ell = \tau$

– $A,\vec{y} \vdash \overline{Q}_1 \xrightarrow{\overline{w}z} Q_1'$, $A,\vec{y} \vdash \overline{Q}_2 \xrightarrow{wz} Q_2'$
– $Q = \nu\vec{y}(Q_1' \mid Q_2')$, if $z \in A,\vec{y}$
– $Q = \nu\vec{y}\nu z(Q_1' \mid Q_2')$, if $z \notin A,\vec{y}$

We then have a bijection b with the same properties as above. Therefore it is the case that there exists states t_1 and t_2 such that $A,\vec{y} \vdash b\mathsf{s}_1 \xrightarrow{\overline{w}z} t_1$ and $A,\vec{y} \vdash b\mathsf{s}_2 \xrightarrow{wz} t_2$, with $\langle Q_1', t_1\rangle \in \mathcal{B}(\{z\} \cup (A,\vec{y}))$ and $\langle Q_2', t_2\rangle \in \mathcal{C}(\{z\} \cup (A,\vec{y}))$. Thus

$$A \vdash \mathsf{s}_1 \,|_A\, \mathsf{s}_2 = b\mathsf{s}_1 \,|_A\, b\mathsf{s}_2 \xrightarrow{\tau} \mathsf{t}_1 \,|_A\, \mathsf{t}_2 \ ,$$

with $\langle Q, \mathsf{t}_1 \,|_A\, \mathsf{t}_2\rangle \in (\mathcal{B} \mid \mathcal{C})(A)$.

3. There exist $\vec{y}$, $\overline{Q}_1$ and $\overline{Q}_2$ such that
   – $\nu\vec{x}(Q_1 \mid Q_2) = \nu\vec{y}(\overline{Q}_1 \mid \overline{Q}_2)$
   – $\ell = \overline{w}z$, with $z \in \vec{y}$ and $w \in A$
   – $A,\vec{y} \vdash \overline{Q}_1 \xrightarrow{\overline{w}z} Q_1'$
   – $Q = \nu\vec{y} \setminus \{z\}(Q_1' \mid \overline{Q}_2)$

   Again we have a bijection b with the same properties as above and thus there exists t_1 such that $A,\vec{y} \vdash b\mathsf{s}_1 \xrightarrow{\overline{w}z} \mathsf{t}_1$ with $\langle Q_1', t_1\rangle \in \mathcal{B}(A,\vec{y} \setminus \{z\})$. Therefore we also have that
$$A \vdash \mathsf{s}_1 \,|_A\, \mathsf{s}_2 = b\mathsf{s}_1 \,|_A\, b\mathsf{s}_2 \xrightarrow{\overline{w}z} \mathsf{t}_1 \,|_{A,z}\, \iota b\mathsf{s}_2 \ .$$

The matching of transitions of $[\![P_1 \mid P_2]\!]_I$ by transitions of $(\![P_1 \mid P_2]\!)_I$ is proven with a similar analysis (cf. the rules that define the parallel composition of two $\mathcal{N}$-LTSs).

□

Using the lemma above it is now easy to prove the Theorem 4.5:

Proof:[of Theorem 4.5] Suppose P and Q are two π-terms with free names in I. Then by definition, $P \dot\sim_I Q$ if and only if $(\![P]\!)_I$ is bisimilar to $(\![Q]\!)_I$. By Lemma 4.6, then $[\![P]\!]_I$ is bisimilar to $[\![Q]\!]_I$. Conversely, if $[\![P]\!]_I$ is bisimilar to $[\![Q]\!]_I$, then, by Lemma 4.6, $(\![P]\!)_I$ is bisimilar to $(\![Q]\!)_I$ and thus $P \dot\sim_I Q$.

□

It is worth noticing that since open bisimilarity is the finest equivalence that we have defined, Theorem 4.5 holds also if we change $\dot\sim_I$ with any of the other strong or weak standard (early) equivalences.

Replication From the point of view of a denotational semantics, a concrete description of a replication operator can be bypassed using initial algebras, i.e. fixed points, to model recursively defined and thus also replicated processes. The crucial point is establishing, by tedious verification, the following:

Theorem 4.7 *If T is a $\mathcal{N}$-LTS with initial name set I, x a name not in I, y and z names in I, then the functors:*

- $(T \oplus -) : \mathcal{N}\text{-}\mathbf{LTS}_I \to \mathcal{N}\text{-}\mathbf{LTS}_I$,
- $(T \mid -) : \mathcal{N}\text{-}\mathbf{LTS}_I \to \mathcal{N}\text{-}\mathbf{LTS}_I$,
- $\nu_{x\in(I,x)}(-) : \mathcal{N}\text{-}\mathbf{LTS}_{I,x} \to \mathcal{N}\text{-}\mathbf{LTS}_I$,

- $\overline{y}z(-) : \mathcal{N}\text{-}\mathbf{LTS}_I \to \mathcal{N}\text{-}\mathbf{LTS}_I$,
- $\tau(-) : \mathcal{N}\text{-}\mathbf{LTS}_I \to \mathcal{N}\text{-}\mathbf{LTS}_I$,
- $yx(-) : \mathcal{N}\text{-}\mathbf{LTS}_{I,x} \to \mathcal{N}\text{-}\mathbf{LTS}_I$,
- $[y = z](-) : \mathcal{N}\text{-}\mathbf{LTS}_I \to \mathcal{N}\text{-}\mathbf{LTS}_I$ *and*
- $[y \neq z](-) : \mathcal{N}\text{-}\mathbf{LTS}_I \to \mathcal{N}\text{-}\mathbf{LTS}_I$

preserve colimits of ω-chains.

The semantics can then be given as follows.

- $[\![!P]\!]_I = ![\![P]\!]_I$, where for every $\mathcal{N}$-LTS T with initial name set I, $!T$ is defined to be $\mu(T \mid -)$. This is the object part of an initial algebra

$$T \mid \mu(T \mid -) \xrightarrow{\simeq} \mu(T \mid -)$$

for the endofunctor $(T \mid -) : \mathcal{N}\text{-}\mathbf{LTS}_I \to \mathcal{N}\text{-}\mathbf{LTS}_I$. Notice that such an initial algebra exists since $\mathcal{N}\text{-}\mathbf{LTS}_I$ has colimits of ω-chains and every operation involved in the semantics preserves this kind of colimits (Theorem 4.7). Moreover it can be calculated in the usual way as the colimit of the "standard" chain

$$0_I \to T \mid 0_I \to T \mid (T \mid 0_I) \to T \mid (T \mid (T \mid 0_I)) \to \cdots .$$

To generalise Theorem 4.5 to a calculus with replication, exhibiting a correspondence between this construction and the standard operational semantics for replication

$$\textsc{Rep}\ \frac{A \vdash P \mid !P \xrightarrow{\ell} P'}{A \vdash !P \xrightarrow{\ell} P'}$$

would require a lemma showing that if $(\![P]\!)_I$ and $[\![P]\!]_I$ are bisimilar then so are their n-way parallel compositions, and then a verification of a bisimulation relation. We do not develop the details here.

5 $\mathcal{N}$-LATS

In this section we define a class of causal models by smoothly lifting the notion of labelled asynchronous[1] transition system [3, 55, 58] (LATS for short) to our indexed setting. LATS are a simple extension of standard LTS in which transitions have both standard labels and *events*, upon which an independence relation is defined. Roughly speaking, concurrency is modelled by requiring that transitions tagged with independent events might occur in any order. As discussed in the introduction, in π-calculi dependencies between transitions may arise from their name usage:

Definition 5.1 *If A is a set of names and ℓ_1 and ℓ_2 are two labels, we say that ℓ_2 is A-dependent on ℓ_1 if one of the following two cases applies:*

1. $\mathrm{val}(\ell_1) = \mathrm{chan}(\ell_2) \not\subseteq A$

[1]There is an unfortunate clash of terminology here: this usage of 'asynchronous' is unrelated to the usage describing process calculi without output prefixing.

2. $\mathrm{val}(\ell_1) = \mathrm{val}(\ell_2) \not\subseteq A$*, one of* ℓ_1, ℓ_2 *is an input action and the other is an output action.*

Definition 5.2 *Define an Indexed LATS ($\mathcal{N}$-LATS) to be a structure*

$$T = \langle S : \mathcal{N} \to \mathbf{Set},\ \longrightarrow,\ \mathsf{i},\ E,\ \mathcal{I} \rangle$$

where $\mathsf{i} = \langle I, i \rangle \in \mathsf{S}$,

$$\longrightarrow\ \subseteq \mathsf{S} \times (\mathrm{Lab} \times E) \times \mathsf{S}\ ,$$

E is a set of events, $\mathcal{I} \subseteq E \times E$ is an independence relation between events and the following conditions hold.

1. *For every event $e \in E$, the structure $\langle S : \mathcal{N} \to \mathbf{Set}, \underset{e}{\longrightarrow}, \mathsf{i} \rangle$ is a $\mathcal{N}$-LTS, where $\underset{e}{\longrightarrow}$ is the set $\{ \langle \mathsf{s}, \ell, \mathsf{t} \rangle \mid \langle \mathsf{s}, \ell, e, \mathsf{t} \rangle \in \longrightarrow \}$.*
2. *$\mathcal{I}$ is irreflexive and symmetric*
3. *If $A \vdash \mathsf{s} \xrightarrow[e_1]{\ell_1} \mathsf{t}$, and $\mathsf{t} \xrightarrow[e_2]{\ell_2} \mathsf{u}$, and $e_1 \mathcal{I} e_2$, and moreover ℓ_2 is not A-dependent on ℓ_1, then there exists a state t' such that $\mathsf{s} \xrightarrow[e_2]{\ell_2} \mathsf{t}'$ and $\mathsf{t}' \xrightarrow[e_1]{\ell_1} \mathsf{u}$.*

Often LATS are defined using more axioms (see [3, 58]). Here we have decided to keep the axiomatisation as light as possible, as none of the extra axioms is directly relevant for the definability of the semantic constructions that we consider. Moreover we allow the same event to carry different labels. This is particularly useful in coping with the proliferation of transitions induced by reindexing and by the input actions. It is not difficult to devise simple variations of our definition which adhere more closely to the traditional case.

Building on the independence relation, transitions occurring in a run of a process can be given a causal partial order describing which transitions are necessary conditions for the occurrence of others. Roughly speaking one transition causes the following one if the corresponding events are not independent of each other. As discussed in the introduction, one can choose whether or not to consider name dependencies – for $\mathcal{N}$-LATS there are two natural ways of defining partial orders out of runs, one taking account only of the independence relation and another which also takes name dependencies into account.

NOTATION: For every natural number n, write $[n]$ for the set $\{k \mid 1 \leq k \leq n\}$. Observe that, in particular, $[0] = \emptyset$.

Definition 5.3 *For every run ρ*

$$A_0 \vdash \mathsf{s}_0 \xrightarrow[e_1]{\ell_1} \mathsf{s}_1 \xrightarrow[e_2]{\ell_2} \mathsf{s}_2 \cdots \xrightarrow[e_n]{\ell_n} \mathsf{s}_n$$

of an $\mathcal{N}$-LATS we define two labelled partial orders:

1. *Define $\mathrm{po}(\rho)_{\mathcal{I}} = \langle [n], \trianglelefteq^{\rho}_{\mathcal{I}}, l^{\rho} \rangle$, where*
   (a) *n is the length of the run ρ.*
   (b) *$\trianglelefteq^{\rho}_{\mathcal{I}}$ is the transitive closure of $\preceq^{\rho}_{\mathcal{I}}$ which is defined as $i \preceq^{\rho}_{\mathcal{I}} j$ if $i \leq j$ and $\neg(e_i \mathcal{I} e_j)$*
   (c) *$l^{\rho}(k) = \ell_k$, for every $k \in [n]$*
2. *Define $\mathrm{po}(\rho)_{\mathcal{I}D} = \langle [n], \trianglelefteq^{\rho}_{\mathcal{I}D}, l^{\rho} \rangle$, where n and l^{ρ} are obtained as above, while $\trianglelefteq^{\rho}_{\mathcal{I}D}$ is the transitive closure of $\preceq^{\rho}_{\mathcal{I}D}$ which is defined as $i \preceq^{\rho}_{\mathcal{I}D} j$ if $i \leq j$ and either not $e_i \mathcal{I} e_j$ or ℓ_j is A_i-dependent on ℓ_i, where $\mathsf{s}_i = (A_i, s_i)$.*

History preserving bisimulation [46, 23, 17] is a bisimulation between runs of processes which accounts for causality by requiring related runs to originate isomorphic partial orders of transitions:

Definition 5.4 *If T is an $\mathcal{N}$-LATS, define* $\mathrm{Run}(T)$ *to be the set of runs of T. If ρ is a run* $I \vdash \mathsf{i} \xrightarrow[e_1]{\ell_1} \mathsf{s}_1 \xrightarrow[e_2]{\ell_2} \mathsf{s}_2 \longrightarrow \cdots \xrightarrow[e_n]{\ell_n} \mathsf{s}_n$ *of T then define*

1. $\mathrm{end}(\rho) \stackrel{\mathrm{def}}{=} \mathsf{s}_n$
2. $\mathrm{names}(\rho) \stackrel{\mathrm{def}}{=} A_n$, *where* $\mathsf{s}_n = \langle A_n, s \rangle$.

NOTATION: If ρ is a run of an $\mathcal{N}$-LATS and $\mathrm{names}(\rho) \vdash \mathrm{end}(\rho) \xrightarrow[e]{\ell} \mathsf{s}$ is a transition, we write $\rho \xrightarrow[e]{\ell} \mathsf{s}$ for the run which extends ρ with that transition. If ρ and ρ' are runs, we write $\rho \xrightarrow[e]{\ell} \rho'$ if ρ' extends ρ with the transition $\mathrm{names}(\rho) \vdash \mathrm{end}(\rho) \xrightarrow[e]{\ell} \mathrm{end}(\rho')$.

Definition 5.5 *Let T_1 and T_2 be two $\mathcal{N}$-LATSs with initial name-set I. A relation $\mathcal{B} \subseteq$* $\mathrm{Run}(T_1) \times \mathrm{Run}(T_2)$ *is an history preserving bisimulation (hpb) if it satisfies the following conditions*

1. $(I \vdash \mathsf{i}_1, I \vdash \mathsf{i}_2) \in \mathcal{B}$
2. $(\rho_1, \rho_2) \in \mathcal{B}$ *implies*
   (a) $\mathrm{po}(\rho_1)_{\mathcal{I}} = \mathrm{po}(\rho_2)_{\mathcal{I}}$
   (b) *if* $\rho_1 \xrightarrow[e_1]{\ell}_1 \rho_1'$ *then there exists a run* ρ_2' *of* T_2 *and an event* $e_2 \in E_2$ *such that* $\rho_2 \xrightarrow[e_2]{\ell}_2 \rho_2'$ *and* $(\rho_1', \rho_2') \in \mathcal{B}$
   (c) *the symmetric condition to the above.*

The relation $\mathcal{B}$ is a name-dependency aware hpb (ndahpb) if the condition $2(a)$ *is changed into* $\mathrm{po}(\rho_1)_{\mathcal{I}D} = \mathrm{po}(\rho_2)_{\mathcal{I}D}$.

The constructions of Section 4 can be easily adapted to become constructions on $\mathcal{N}$-LATS. We shall now briefly indicate how they need to be extended to take account of the presence of events and of the independence relation. In all cases where a label is carried from the premise to the conclusion, the event is also carried (suitably injected).

Restriction The set of events and the independence relation does not change.

Prefixes A new event, not in the independence relation with any other is added and it decorates all of the new transitions.

Deadlock The set of events is empty and so is the independence relation.

Matching and Mismatching Events and the independency relation are left untouched.

Sum The set of events is taken to be the disjoint union of the originals but no new independence pairs are added.

Parallel composition If E_1 and E_2 are the two sets of events we define E_0 to be the disjoint union $E_1 \uplus (E_1 \times E_2) \uplus E_2$. Writing this as $(E_1 \times \{\star\}) \cup (E_1 \times E_2) \cup (\{\star\} \times E_2)$ for $\star \notin E_1 \cup E_2$, the independence relation is defined by $\langle e_1, e_2 \rangle \mathcal{I}_0 \langle e_1' e_2' \rangle$ if both $e_1 \hat{\mathcal{I}}_1 e_1'$ and $e_2 \hat{\mathcal{I}}_2 e_2'$, where $\hat{\mathcal{I}}_k$ is the union of $\mathcal{I}_k$ and $\langle \star, \star \rangle$. The new τ-transitions are decorated by the pairs of enabling events.

Process terms can then be given a denotational semantics and be related by (nda) history preserving bisimilarity. In the remainder of this section we will mostly concentrate on the relationship between our semantics and the causal bisimulation of [6]. In particular we present correspondence results relating our hbp semantics to causal bisimulation, and further discuss name-dependency.

In the paper [6], no notion of strong causal bisimulation is defined – the authors directly defined causal bisimulation in the weak form, abstracting away from τ actions. To match with our definitions we therefore need either to define weak history-preserving bisimulation or to modify their setting in order to make τ actions visible. We will in fact do both, ending up with two correspondence results, one for strong and one for weak bisimulation. We now briefly recall the definition of causal bisimulation of *loc. cit.* which also requires defining a variant of the π-calculus with explicit *causality information.* We refer to *loc. cit.* for a detailed discussions of the relevance of their approach. Also, we show here what modifications are needed in order to define, in their setting, strong causal bisimulation.

As in [6] we take a set $\mathcal{K}$ of *causes*, distinct from the names N, and extend the process syntax with a construct $K :: P$ where $K \subseteq_{\mathsf{fin}} \mathcal{K}$ is the set of causes that have affected P. For simplicity we also omit matching and mismatching (as in [6]), omit replication, and adopt the unstratified grammar below (in contrast to [6]).

$$P ::= 0 \;\big|\; P \mid P \;\big|\; \tau.P \;\big|\; \overline{x}v.P \;\big|\; xp.P \;\big|\; P + P \;\big|\; (\boldsymbol{\nu} x)P \;\big|\; K :: P$$

NOTATION: We write $\mathcal{K}(P)$ for the set of causes occurring anywhere in P, and $[k \rightsquigarrow K]P$ for P with each cause-set K' containing k replaced by $K \cup (K' \setminus \{k\})$. In the appendix we let P, Q range over terms of the grammar that do not contain subterms $K :: R$ and C, D range over arbitrary terms.

The causal operational semantics is given in Figure 3 which defines a transition relation

$$A \vdash P \xrightarrow[K;k]{\ell} Q$$

Here K is the set of prior causes of this transition and k is a new cause. The definition differs slightly from that of [6] – it is explicitly indexed and it maintains causes for τ-transitions.

Strong causal bisimulation can now be defined in the usual way, by requiring transitions to agree not only on the labels but on the causes too. In detail, take *strong causal bisimulation* $\dot{\sim}^c$ to be the largest family of relations indexed by finite sets of names such that each $\dot{\sim}^c{}_A$ is a symmetric relation over $\{ P \mid \mathrm{fn}(P) \subseteq A \}$ and for all $P \mathrel{\dot{\sim}^c{}_A} Q$,

- if $A \vdash P \xrightarrow[K;k]{\ell} P'$ then $\exists Q' \,.\, A \vdash Q \xrightarrow[K;k]{\ell} Q' \wedge P' \mathrel{\dot{\sim}^c{}_{A \cup \mathrm{fn}(\ell)}} Q'$.

This is the obvious adaptation of the definition in [6] to the explicitly-indexed strong case. We can now state our first non-interleaving correspondence result:

Theorem 5.6 *Let P and Q be two terms of the π-calculus with free names in I and let $[\![P]\!]^c_I$ and $[\![Q]\!]^c_I$ be their interpretations as $\mathcal{N}$-LATS's. Then $[\![P]\!]^c_I$ is history preserving bisimilar to $[\![Q]\!]^c_I$ if and only if P is strongly causal bisimilar to Q.*

Proof:[Hint] The proof requires some constructions and lemmas. First of all one relates the (causal) operational semantics of π-terms with transitions in asynchronous transition systems, in terms of so-called *run-bisimulations.* These are relations between runs of processes and runs of $\mathcal{N}$-LATS's satisfying the usual coinductive properties as well as a condition relating corresponding partial order of causes and of events. The crucial step is then that of establishing

$$\text{OUT}\ \frac{}{A \vdash \overline{x}v.P \xrightarrow[\emptyset;k]{\overline{x}v} k :: P} \qquad \text{IN}\ \frac{}{A \vdash xp.P \xrightarrow[\emptyset;k]{xv} k :: \{^v/_p\}P}$$

$$\text{TAU}\ \frac{}{A \vdash \tau.P \xrightarrow[\emptyset;k]{\tau} k :: P} \qquad \text{CAU}\ \frac{A \vdash P \xrightarrow[K;k]{\ell} P'}{A \vdash K' :: P \xrightarrow[K \cup K';k]{\ell} K' :: P'}$$

$$\text{RES}\ \frac{A, x \vdash P \xrightarrow[K;k]{\ell} P' \quad x \notin \text{fn}(\ell)}{A \vdash (\boldsymbol{\nu} x)P \xrightarrow[K;k]{\ell} (\boldsymbol{\nu} x)P'} \qquad \text{OPEN}\ \frac{A, x \vdash P \xrightarrow[K;k]{\overline{y}x} P' \quad y \neq x}{A \vdash (\boldsymbol{\nu} x)P \xrightarrow[K;k]{\overline{y}x} P'}$$

$$\text{SUM}\ \frac{A \vdash P \xrightarrow[K;k]{\ell} P'}{A \vdash P + Q \xrightarrow[K;k]{\ell} P'} \qquad \text{PAR}\ \frac{A \vdash P \xrightarrow[K;k]{\ell} P'}{A \vdash P \mid Q \xrightarrow[K;k]{\ell} P' \mid Q}$$

$$\text{COM}\ \frac{A \vdash P_1 \xrightarrow[K_1;k]{\overline{x}y} P_1' \qquad A \vdash P_2 \xrightarrow[K_2;k]{xy} P_2'}{A \vdash P_1 \mid P_2 \xrightarrow[K_1 \cup K_2:k]{\tau} (\boldsymbol{\nu}\{y\} - A)([k \rightsquigarrow (K_2 \cup k)]P_1' \mid [k \rightsquigarrow (K_1 \cup k)]P_2')}$$

In all rules with conclusion of the form $A \vdash P \xrightarrow[K;k]{\ell} Q$ there are implicit sideconditions $\text{fn}(P) \subseteq A$ and $k \notin \mathcal{K}(P)$. Symmetric versions of PAR, COM, and SUM are elided.

Figure 3: π causal operational semantics

that every π-term P, with free names in I, is run-bisimilar to its corresponding $\mathcal{N}$-LATS, $[\![P]\!]^c_I$. This latter result paves the way for the proof of the Theorem. Details can be found in Appendix A.

$\square$

A weak version of history preserving bisimulation can be given in the spirit of [57].

Definition 5.7 *Let ρ be a run in an asynchronous transition system, let n be the length of ρ, and let n^τ be the number of transitions in ρ which are not labelled τ. For every $i \leq n^\tau$, define $n_i \leq n$ inductively as follows: n_1 is the smallest number h such that the h-th transition of ρ has label $\ell_h \neq \tau$; n_{j+1} is the smallest number h such that the h-th transition of ρ has label $\ell_h \neq \tau$ and that moreover is strictly bigger than n_j.*

Starting with a run ρ of an $\mathcal{N}$-LATS, by means of the above definition, we can define partial orders of observable events in runs as follows:

Definition 5.8 *Let ρ be a run of an $\mathcal{N}$-LATS and let $\text{po}(\rho)_\mathcal{I}$ and $\text{po}(\rho)_{\mathcal{I}D}$ be the corresponding partial orders as in Definition 5.3. Define $\text{po}(\rho)_{w\mathcal{I}}$ and $\text{po}(\rho)_{w\mathcal{I}D}$ to be the partial orders $\langle [n^\tau], \trianglelefteq_{w\mathcal{I}}, l^\rho_w \rangle$ and $\langle [n^\tau], \trianglelefteq_{w\mathcal{I}D}, l^\rho_w \rangle$, respectively, where $l^\rho_w(i) = l^\rho(n_i)$, $i \trianglelefteq_{w\mathcal{I}} j$ if $n_i \trianglelefteq_\mathcal{I} n_j$ and $i \trianglelefteq_{w\mathcal{I}D} j$ if $n_i \trianglelefteq_{\mathcal{I}D} n_j$.*

Weak history preserving bisimulations are now defined as relations between runs as in Definition 5.5 but where, as usual, "strong" transitions $\rho_1 \xrightarrow[e_1]{\ell}_1 \rho_1'$ are simulated by "weak" ones $\rho_2 \overset{\hat{\ell}}{\underset{e_2}{\Longrightarrow}}_2 \rho_2'$ (and symmetrically) and with condition $2(a)$ replaced by

$$\text{po}(\rho_1)_{w\mathcal{I}} = \text{po}(\rho_2)_{w\mathcal{I}}$$

or by $\mathrm{po}(\rho_1)_{w\mathcal{I}D} = \mathrm{po}(\rho_2)_{w\mathcal{I}D}$, for the name-dependency aware case. One can use the main lemma developed for the proof of Theorem 5.6, to prove the following result:

Theorem 5.9 *Let P and Q be two terms of the π-calculus with free names in I and let $[\![P]\!]^c_I$ and $[\![Q]\!]^c_I$ be their interpretations as $\mathcal{N}$-LATSs. Then $[\![P]\!]^c_I$ is weak history preserving bisimilar to $[\![Q]\!]^c_I$ if and only if P is causal bisimilar to Q in the sense of [6].*

Proof:[Sketch] Weak causal bisimulations between π-terms, generate relations between runs of the corresponding processes. These can be *composed* with the run-bisimulations of Lemma A.9 to give weak history preserving bisimulations. Vice versa given a weak history preserving bisimulation between the denotations of two processes, one obtains a relation between runs of the two processes by composing with the run-bisimulations of Lemma A.9. The relation on causal π agents, which relates end points of related runs is a weak causal bisimulation.

$\square$

In [6] it is argued that, because of the dependencies due to the binding of names, processes like $(\boldsymbol{\nu} y)(\overline{x}y.\overline{y}z)$ and $(\boldsymbol{\nu} y)(\overline{x}y|\overline{y}z)$ should be indistinguishable by an external observer. Nonetheless causal bisimulation distinguishes them, as it only tracks the dependencies due to the structure of processes – in the example, one output is prefixing the other in the first process but not in the second. The paper leaves open the possibility of a further refinement of the treatment of causes in the operational semantics to identify the above two processes.

Their remark has been tackled in [30], where a domain model of π-terms based on Kahn networks is presented. There the induced equivalence equates the two processes, but it seems to us that the equivalence is anyway a traced-based rather than a bisimulation based one. In [18], the authors use the combination of different partial orders to achieve the effect of equating the two processes above. In this paper we instead refined the way the causal order of events in a run is determined. This has led to the notion of name-dependency aware history preserving bisimulation defined above. It is easy to verify that name-dependency aware history preserving bisimilarity is a coarser relation than history preserving bisimilarity and that the former equates the two example processes:

Proposition 5.10 *If two asynchronous transition systems are history preserving bisimilar than they are name-dependency aware history preserving bisimilar.*

Proposition 5.11 *The denotations of the process terms $(\boldsymbol{\nu} y)(\overline{x}y.\overline{y}z)$ and $(\boldsymbol{\nu} y)(\overline{x}y|\overline{y}z)$ are name-dependency aware history preserving bisimilar but not history preserving bisimilar.*

6 Relating $\mathcal{N}$-LTSs and $\mathcal{N}$-LATSs

We continue now the abstract study, initiated in the second part of Section 3, of the structures defined in this paper. We begin by defining, for name-sets I, categories $\mathcal{N}$-$\mathbf{LATS}_I$ of $\mathcal{N}$-LATSs. We then establish some categorical properties of $\mathcal{N}$-$\mathbf{LATS}_I$ and, analogously to the case of $\mathcal{N}$-LTSs, define a category $\mathcal{N}$-$\mathbf{LATS}$ which cofibres over $\mathcal{N}$ – the fibres being the categories $\mathcal{N}$-$\mathbf{LATS}_I$. Finally we conclude by showing the existence of adjunctions relating $\mathcal{N}$-$\mathbf{LTS}_I$ and (a full subcategory of) $\mathcal{N}$-$\mathbf{LATS}_I$, and how these can be glued together to provide a (fibred) adjunction between $\mathcal{N}$-$\mathbf{LTS}$ and (a full subcategory of) $\mathcal{N}$-$\mathbf{LATS}$.

Definition 6.1 *Define a morphism $T_1 \to T_2$ between $\mathcal{N}$-LATSs with initial name-set I to consist of a pair $\langle \alpha, \eta \rangle$ where*

- *$\alpha : S_1 \Longrightarrow S_2$ is a natural transformation*

- $\eta : E_1 \to E_2$ *is a function*

such that

1. $\alpha \mathsf{i}_1 = \mathsf{i}_2$
2. $\mathsf{s} \xrightarrow[e]{\ell} \mathsf{s}'$ *implies* $\alpha\mathsf{s} \xrightarrow[\eta e]{\ell} \alpha\mathsf{s}'$
3. $e\mathcal{I}_1 e'$ *implies* $\eta e \mathcal{I}_2 \eta e'$.

Define $\mathcal{N}$-$\mathbf{LATS}_I$ *to be the category of* $\mathcal{N}$-*LATSs with initial name-set* I *and these morphisms.*

Because of restrictions imposed by the axioms which governs the independence relation between events of a $\mathcal{N}$-LATS, the category $\mathcal{N}$-$\mathbf{LATS}_I$ does not enjoy all the completeness and cocompleteness properties of $\mathcal{N}$-$\mathbf{LTS}_I$, still it has enough colimits to allow for a denotational semantics of recursively defined processes.

Theorem 6.2 *For every name set* I, *the category* $\mathcal{N}$-$\mathbf{LATS}_I$ *has small coproducts and colimits of filtered diagrams.*[2] *It also has limits of every non-empty diagram.*

Proof: The constructions are extensions of those seen in the proof of Theorem 3.13 for $\mathcal{N}$-LTSs. We outline some.

- The initial object is $\langle \mathcal{N}(I,-), \emptyset, \langle I, 1_I\rangle, \emptyset, \emptyset\rangle$.
- The coproduct of a family $(T_k = \langle S_k, \longrightarrow_k, \langle I, i_k\rangle, E_k, \mathcal{I}_k\rangle)_{k\in K}$ of $\mathcal{N}$-LATSs is given by $\langle S, \longrightarrow, \langle I, i\rangle, E, \mathcal{I}\rangle$, where S and the initial state $\langle I, i\rangle$ are obtained as in the analogous construction in the proof of Theorem 3.13; $E = \coprod_{k\in K} E_k$, while $\mathcal{I} = \coprod_{k\in K} \mathcal{I}_k$. If $(in_k^S : S_k \to S)_{k\in K}$ and $(in_k^E : E_k \to E)_{k\in K}$ are the cones of injections, then define $\mathsf{s} \xrightarrow[e]{\ell} \mathsf{t}$ if there exist $k \in K$, $\bar{\mathsf{s}}, \bar{\mathsf{t}} \in \mathsf{S}_k$ and $\bar{e} \in E_k$, such that $\bar{\mathsf{s}} \xrightarrow[\bar{e}]{\ell}_k \bar{\mathsf{t}}$ and such that $in_k^S \bar{\mathsf{s}} = \mathsf{s}$, $in_k^S \bar{\mathsf{t}} = \mathsf{t}$ and $\in_k^E (\bar{e}) = e$.
- If $\mathbb{D}$ is a filtered category and $\Delta : \mathbb{D} \to \mathcal{N}$-$\mathbf{LATS}_I$ is a functor, then a colimiting cone

$$(\langle \alpha_D, \eta_D\rangle : \Delta(D) \to T)_{D\in|\mathbb{D}|}$$

  can be built as follows:

  - $(\alpha_D : S_D \to S)_{D\in|\mathbb{D}|}$ is the colimit of $\mathbb{D} \xrightarrow{\Delta} \mathcal{N}$-$\mathbf{LATS}_I \xrightarrow{p^S} \mathbf{Set}^{\mathcal{N}}$, where p^S is the obvious projection functor.
  - Similarly $(\eta_D : E_D \to E)_{D\in|\mathbb{D}|}$ is the colimit in $\mathbf{Set}$ of $\mathbb{D} \xrightarrow{\Delta} \mathcal{N}$-$\mathbf{LATS}_I \xrightarrow{p^E} \mathbf{Set}$.
  - $e\mathcal{I}e'$ if there exist $d \in |\mathbb{D}|$ and $\bar{e}, \bar{e'} \in E_D$ such that $\bar{e}\mathcal{I}_D\bar{e'}$ and $\eta_D(\bar{e}) = e$ and $\eta_D(\bar{e'}) = e'$.
  - $\mathsf{s} \xrightarrow[e]{\ell} \mathsf{t}$ if there exist $d \in |\mathbb{D}|$, $\bar{e} \in E_D$ and $\bar{\mathsf{s}}, \bar{\mathsf{t}} \in \mathsf{S}_D$ such that $\bar{\mathsf{s}} \xrightarrow[e]{\ell}_D \bar{\mathsf{t}}$ and $\alpha_D(\bar{\mathsf{s}}) = \mathsf{s}$ and $\alpha_D(\bar{\mathsf{t}}) = \mathsf{t}$.
  - $\langle I, i\rangle = \langle I, (\alpha_D)_I(i_D)\rangle$, for some D (by filteredness, all the initial states are mapped to the same state of $S(I)$)

[2] A filtered diagram is a diagram obtained from a filtered category (see [36] for the definition of this notion).

It is straightforward to verify that T is a $\mathcal{N}$-LATS and that the cone is colimiting.

Turning to limits it suffices to show that $\mathcal{N}$-$\mathbf{LATS}_I$ has equalisers and small (but non-empty) products. Again these are defined by simple extensions of the analogous constructions in $\mathcal{N}$-$\mathbf{LTS}_I$. We leave the reader to work out such details.

□

In particular we can see that $\mathcal{N}$-$\mathbf{LATS}_I$ does not have terminal object (the limit of the empty diagram) and in general it does not have coequalisers – to exist, these would require the possibility of "autoconcurrent" events. It is worth remarking that the situation here is quite similar to that of standard LTSs and Asynchronous Transition Systems or Event Structures [58]. Again it is easy to see how every reindexing function $f : A \to B$ induces a functor $R(f) : \mathcal{N}\text{-}\mathbf{LATS}_A \to \mathcal{N}\text{-}\mathbf{LATS}_B$ and thus that all the categories $\mathcal{N}$-$\mathbf{LATS}_I$ can be "glued" together with the Grothendieck construction to provide a cofibration $\mathcal{N}\text{-}\mathbf{LATS} \to \mathcal{N}$.

An adjunction The category $\mathcal{N}$-$\mathbf{LTS}_I$ is a full subcategory of $\mathcal{N}$-$\mathbf{LATS}_I$. Moreover the embedding functor, L_I, has a right inverse R_I. The functor L_I can be defined as follows:

On objects $L_I\langle S, \longrightarrow, \mathsf{i}\rangle = \langle S, \longrightarrow', \mathsf{i}, E, \mathcal{I}\rangle$. The set of events E is the set of equivalence classes of transitions of the equivalence relation $\frown$ generated by the following reflexive and transitive relation: $(A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t} \dashv B) \frown^{\circ} (A' \vdash \mathsf{s}' \xrightarrow{\ell'} \mathsf{t}' \dashv B')$ if there exist $f : A \to_{\mathrm{inj}} A'$ and $g : B \setminus A \to B'$ such that $\mathsf{s}' = f\mathsf{s}$, $\mathsf{t}' = [\iota f, g]\mathsf{t}$ and $\ell' = [\iota f, g]\ell$, where $\iota : A' \hookrightarrow B'$ is the inclusion function. The independence relation $\mathcal{I}$ is the empty relation. The transition relation is defined as follows: $A \vdash \mathsf{s} \underset{e}{\overset{\ell}{\longrightarrow}}{}' \mathsf{t} \dashv B$ if $e = [A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t} \dashv B]_{\frown}$.

On arrows $L_I(\alpha) = \langle \alpha, \eta\rangle$, where $\eta([A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t} \dashv B]_{\frown}) = [A \vdash \alpha_A \mathsf{s} \xrightarrow{\ell} \alpha_B \mathsf{t} \dashv B]_{\frown}$.

The functor R_I is defined by simply forgetting the extra structure, i.e. the set of events and the independence relation, which distinguishes $\mathcal{N}$-LATSs from $\mathcal{N}$-LTSs. From a categorical perspective it would be nice if the pair of functors $\langle L_I, R_I\rangle$ was an adjoint pair. This might help in providing a formal understanding of the relationship between the constructions used for the semantics of π-terms in either categories. Unfortunately, as also noted in [58] for the case of "standard" LATS and LTS, this is not the case: in an (Indexed) LATS, one can have several transitions carrying the same label between two states. This is something clearly not possible for an (Indexed) LTS. This mismatching leads to the impossibility of defining a natural transformation $LR \Longrightarrow Id$ which can act as the counit of the adjunction. A solution out of this problem is that of imposing an extra *extensionality* axiom on (Indexed) LATS. It is worth noticing that this axiom also appeared in [25] where a similar formal relationship was sought between LATS and Winskel and Nielsen's Transition System with Independence (a transition system analogue of Petri nets).

Theorem 6.3 *The pair of functors $\langle L_I, R_I\rangle$ forms an adjoint pair, with L_I left adjoint and R_I right adjoint, if we restrict to consider $\mathcal{N}$-LATS such that for every events $e, e' \in E$,*

$$\mathsf{s} \underset{e}{\overset{\ell}{\longrightarrow}} \mathsf{t} \text{ and } \mathsf{s} \underset{e'}{\overset{\ell'}{\longrightarrow}} \mathsf{t} \text{ implies } \ell \neq \ell' \ .$$

Proof: From the definition of adjunction [36], to show that $L_I \dashv R_I$, one has to provide natural transformations $\eta : Id \Rightarrow R_I L_I$ and $\varepsilon : L_I R_I \Rightarrow Id$ such that

$$\varepsilon_{L_I}(L_I \eta) = Id_{L_I} \qquad and \qquad (R_I \varepsilon)\eta_{R_I} = Id_{R_I} \ .$$

As we said R_I is right inverse to L_I, thus η is naturally defined to be the identity natural transformation. Now, if $T = \langle S, \longrightarrow, \mathsf{i}, E, \mathcal{I}\rangle$ is a $\mathcal{N}$-LATS, define $\varepsilon_T : L_I R_I T \to T$ to be $\varepsilon_T = \langle 1_S, \sigma\rangle$, where for every event $[A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t} \dashv B]_{\frown}$ of $L_I R_I T$, $\sigma([A \vdash \mathsf{s} \xrightarrow{\ell} \mathsf{t} \dashv B]_{\frown}) = e$ with e the unique event of E, such that $A \vdash \mathsf{s} \xrightarrow[e]{\ell} \mathsf{t} \dashv B$ in T.

It should be clear the importance of the extensionality axiom in making ε_T well-defined.

It is now immediately clear that, if T is an $\mathcal{N}$-LTS, then $\varepsilon_{L_I T}$ is the identity arrow and thus $\varepsilon_{L_I T}(L_I \eta_T) = 1_{L_I T}$.

On the other hand, for every $\mathcal{N}$-LATS T, $R_I \varepsilon_T$ is clearly the identity on $R_I T$, thus it is also the case that $(R_I \varepsilon_T)\eta_{R_I T} = 1_{R_I T}$.

$\square$

Notice that for any π-term P with free names in I, the corresponding Indexed LATS $[\![P]\!]^c_I$ satisfies the condition of the Theorem above. Moreover, under the hypothesis of Theorem 6.3, the adjoint functors $L_I \dashv R_I$ induce a (fibred) adjunction $L \dashv R : \mathcal{N}\text{-}\mathbf{LATS} \to \mathcal{N}\text{-}\mathbf{LTS}$.

7 Alternative indexing structure

In this paper we considered transition systems with an indexed sets of states. The indexing structure (the category $\mathcal{N}$) that we have chosen is not the only possible one. Other possibilities might reasonably be conceived. In this last section we examine those which have occurred to us and we discuss trade-offs briefly.

Sets and injections Instead of indexing by the category $\mathcal{N}$ one can index by $\mathcal{N}_{\mathrm{inj}}$, the subcategory of $\mathcal{N}$ with all objects but only injective functions as arrows. This gives a simpler structure, in which the transitions of a reindexed state $f\mathsf{s}$ are always determined by those of s. Specifically, define an $\mathcal{N}_{inj}$*-LTS* to be a structure $T = \langle S : \mathcal{N}_{\mathrm{inj}} \to \mathbf{Set}, \longrightarrow, \mathsf{i}\rangle$ where $\mathsf{i} \in \mathsf{S}$ and $\longrightarrow \subseteq \mathsf{S} \times \mathrm{Lab} \times \mathsf{S}$, satisfying axioms 1, 3a, 3b, 4 of Section 3. To make input prefix definable, however, the denotation of a process with n free names must be a function from n-tuples of names to $\mathcal{N}_{\mathrm{inj}}$-LTSs, not simply an $\mathcal{N}$-LTS – to define $[\![xy.P]\!]$ one would need (at least) $[\![\{^z/_y\}P]\!]$ for all $z \in \mathrm{fn}(P), w$. Moreover, we doubt whether an analogue of the input axioms 2a,2b could be stated. Alternatively, one might consider the half-way house of an $\mathcal{N}_{\mathrm{inj}+}$-LTS – an $\mathcal{N}_{\mathrm{inj}}$-LTS with additional data (and appropriate axioms) specifying how the initial state (but no other state) is affected by non-injective renamings. This is less mathematically natural, but has enough structure to support definitions of input prefixing, and of the bisimulation congruence $\sim$ obtained by closing $\dot\sim$ under arbitrary renamings.

Building restriction into the indexing It is arguable that, as restriction is a fundamental π-calculus concept, one should take models with more data than our $\mathcal{N}$-LTSs, specifying how the transitions of states change when names are restricted. This leads to more complex axioms, though clearly also to a simpler definition of the restriction operator. In more detail, define $\mathcal{N}_\nu$ to be the category with objects finite subsets of $\mathcal{N}$ and arrows pairs $\langle f, R_f\rangle : A \to B$ where $f : A \rightharpoonup B$ is a partial function and $R_f \subseteq (A \setminus \mathrm{dom}(f)) \times (A \setminus \mathrm{dom}(f))$ is an equivalence relation. If $A \vdash \mathsf{s}$ then the re-indexing of s along $\langle f, R_f\rangle$ should be thought of as the state in which names in $A \setminus \mathrm{dom}(f)$ have been restricted, after being quotiented by R_f, and other names have been substituted as specified by f. Define composition of arrows by $\langle g, R_g\rangle \circ \langle f, R_f\rangle = \langle g \circ f, R_{g\circ f}\rangle$ where $R_{g\circ f} = R_f \cup \{\, (a, a') \mid f(a)\ R_g\ f(a')\,\}$. Preliminary investigation suggests that with this structure it may be possible to relate parallel composition to the categorical product, as in [58]. Moreover, from the categorical point of view, the explicit restriction reindexings, makes the resulting cofibrations $\mathcal{N}$-**LTS** $\rightarrow \mathcal{N}_\nu$ and $\mathcal{N}$-**LATS** $\rightarrow \mathcal{N}_\nu$ bifibrations, i.e. fibrations as well as cofibrations.

Choosing new names In our definition, for a state s above A, all names $w \notin A$ are treated symmetrically – corresponding to the operational fact that (if $x \in A$) there is a transition $A \vdash (\boldsymbol{\nu} z)\overline{x}z \xrightarrow{\overline{x}w} 0$ for *any* $w \notin A$. One can instead take a chosen new – a function $\boldsymbol{\nu} : \mathcal{P}_{\mathrm{fin}}(\mathcal{N}) \rightarrow \mathcal{N}$ such that $\forall A \,.\, \boldsymbol{\nu} A \notin A$. This leads to an endofunctor $\delta : \mathcal{N} \rightarrow \mathcal{N}$ defined by $\delta\, A = A \cup \{\boldsymbol{\nu} A\}$ and $\delta(f) = f \cup \{\boldsymbol{\nu} A \mapsto \boldsymbol{\nu} B\}$; the axioms can be restated in terms of δ. In this paper we have not taken a chosen new in order to keep the tight correspondence with the operational semantics, and for notational simplicity. The obvious advantage of taking such an approach would be that of drastically restricting the degree of branching of the transition systems. The drawback is in the extra complications occurring in the axiomatisation having to do with the renaming of newly generated and communicated channels.

The chosen-new version of $\mathcal{N}_{\mathrm{inj}}$ is essentially the indexing structure used in [56, 20, 24, 12].

8 Future work and applications

We conclude by hinting at several possible applications and extensions.

Syntax-free Metatheory

Firstly, $\mathcal{N}$-LTSs provide a setting in which some of the π-calculus early metatheory can be developed in a syntax-free style, independently of the exact choice of calculus. In particular, one could show congruence results for operational equivalences with respect to the $\mathcal{N}$-LTS operations defined in Section 4, perhaps developing the open-map approach of [31, 13, 14], and could prove characterisation results relating operational equivalences with classes of formulae of suitable modal logics.

Model Checking

We believe our structures may form a useful basis for π-calculus interleaving and partial-order model checking, via notions of *finitely presentable* $\mathcal{N}$-LTS and $\mathcal{N}$-LATS. The π labelled transition relation is (for non-trivial processes) infinite-branching, but when checking (eg) bisimulation of processes it is often intuitively clear that only a finite number of transitions are 'important'. Several authors have worked on finitary characterisations using a refined symbolic operational semantics [35, 45, 5]. It may be fruitful to consider the alternative approach of model-checking algorithms that work directly over finite presentations, thereby again decoupling the algorithm design from the exact choice of calculus. Depending on what equivalence (or modal properties) one wishes to check, it may be appropriate to work not with $\mathcal{N}$-LTSs but with the $\mathcal{N}_{\mathrm{inj}}$-LTSs or $\mathcal{N}_{\mathrm{inj}+}$-LTSs of Section 7. For illustration we sketch a notion of finite presentation of an $\mathcal{N}_{\mathrm{inj}}$-LTS. We expect there to be interesting relationships with the HD-automata of [43], which provide a notion of minimal realization.

Definition 8.1 *A finite presentation T_0 of an $\mathcal{N}_{inj}$-LTS consists of data $\langle S_0, \longrightarrow_0, \mathsf{i}\rangle$, where*

1. *S_0 is function from $|\mathcal{N}_{inj}|$ to finite sets which is empty almost everywhere. We then define a functor $S : \mathcal{N}_{inj} \rightarrow$ **Set** by*

$$\begin{array}{rcl} S(B) & \stackrel{def}{=} & \{\, \langle s, f\rangle \mid \exists A \,.\, s \in S_0(A) \wedge f : A \rightarrow_{inj} B \,\} \\ S(g)(\langle s, f\rangle) & \stackrel{def}{=} & \langle s, g \circ f\rangle \qquad \textit{for } g : B \rightarrow_{inj} C \textit{ and } \langle s, f\rangle \in S(B) \end{array}$$

As before we write S *for the set* $\coprod_{A\in|\mathcal{N}|} S(A)$*, and also write* S_0 *for* $\coprod_{A\in|\mathcal{N}|} S_0(A)$.

2. $\longrightarrow_0 \subseteq \mathsf{S}_0 \times \mathrm{Lab} \times \mathsf{S}$ *is a finite relation.*

3. $\mathsf{i} \in \mathsf{S}$.

satisfying the axiom

1. *(Naming)* $\langle A, s\rangle \xrightarrow{\ell}_0 \langle B, \langle t, f\rangle\rangle \implies \mathrm{chan}(\ell) \subseteq A \wedge B = A \cup \mathrm{fn}(\ell)$ *(here* $s \in S_0(A)$, $t \in S_0(C)$*, and* $f : C \to_{inj} B$*).*

A more sophisticated notion would include also a finite number of equalities between elements of S. One could now relate the finitely-presentable $\mathcal{N}_{\mathrm{inj}}$-LTSs to those denotable by finite-control process terms, and consider model-checking algorithms over the finite presentations.

Adding Values

In this paper we addressed only *monadic* π-calculus, for notational simplicity. Extension to calculi with polyadic or tuple communication is straightforward. Some applications – notably those involving cryptography – require a further extension to allow communication of values of datatypes that are specified with equations or rewrite rules. This should also be straightforward (though one may wish to exclude equations that discard variables, to obtain an unambiguous notion of the new names in a value). An operational development with equational datatypes has been given by Abadi and Fournet [1].

Synchronisation Algebras

More speculatively, one can ask whether *synchronisation algebras* [58] can be generalised to cover a useful variety of name-passing calculi, thereby supporting uniform definitions of $\mathcal{N}$-LTS and $\mathcal{N}$-LATS. A good test-case here is the synchronisations of the box-π calculus [53, 54], which has an early LTS that is interestingly different from that of the π-calculus. One might also begin a study of early rule formats.

Security Protocols

An analogue of $\mathcal{N}$-LTS, extended to allow communication of tuples and encrypted values, may provide a useful basis for proofs and model-checking of cryptographic protocols. There are two points here. Firstly, many security properties are stated using a quantification over all possible 'attacker' processes. Quantifying over all elements of the model, not merely over syntactically-denotable elements, therefore gives stronger security properties that are less dependent on the precise expressiveness of the calculus used.

Secondly, the model allows alternative styles of definition of system behaviours. For example, in the work of Paulson (see e.g. [44]) systems are described by disjunctions of predicates specifying when a given trace can be extended by a particular label. One can characterise the *well-formed* such predicates (loosely, those that are preserved by new-name substitutions), that define $\mathcal{N}_{\mathrm{inj}}$-LTSs (with state-sets simply the set of all traces). These can then be composed by parallel and restriction operators. One may thereby obtain a tight connection between this work and process-calculus modeling of protocols, e.g. [2].

$$\text{OUT}\ \frac{}{A \vdash \mathsf{C}\!:\!\overline{x}v \xrightarrow{\overline{x}v}_{\mathsf{C}} 0} \qquad \text{IN}\ \frac{}{A \vdash xp.P \xrightarrow{xv}_{\mathsf{C}} \mathsf{C}\bullet\{{}^{v}\!/_{p}\}P}$$

$$\text{PAR}\ \frac{A \vdash P \xrightarrow{\ell}_{\mathsf{C}} P'}{A \vdash P \mid Q \xrightarrow{\ell}_{\mathsf{C}} P' \mid Q} \qquad \text{COMM}\ \frac{A \vdash P \xrightarrow{\overline{x}v}_{\mathsf{C}} P' \quad A \vdash Q \xrightarrow{xv}_{\mathsf{C}} Q'}{A \vdash P \mid Q \xrightarrow{\tau}_{\emptyset} \boldsymbol{\nu}(\{v\} \setminus A)(P' \mid Q')}$$

$$\text{RES}\ \frac{A, x \vdash P \xrightarrow{\ell}_{\mathsf{C}} P' \quad x \notin \mathrm{fn}(\ell)}{A \vdash \boldsymbol{\nu}xP \xrightarrow{\ell}_{\mathsf{C}} \boldsymbol{\nu}xP'} \qquad \text{OPEN}\ \frac{A, x \vdash P \xrightarrow{\overline{y}x}_{\mathsf{C}} P'}{A \vdash \boldsymbol{\nu}xP \xrightarrow{\overline{y}x}_{\mathsf{C}} P'}$$

In all rules with conclusion of the form $A \vdash P \xrightarrow{\ell}_{\mathsf{C}} Q$ there is an implicit side condition $\mathrm{fn}(P) \subseteq A$. Symmetric versions of (PAR) and (COMM) are elided.

Figure 4: Coloured π operational semantics

Relating to Coloured Semantics

Lastly, we observe that the model-theoretic view would enhance work on secure encapsulation [53, 54]. As above, quantifying over elements of the model, rather than over syntactic processes, would allow stronger security properties to be stated. Further, that work introduced an ad-hoc *coloured operational semantics*, to provide a tractable approximate notion of causality for the box-π calculus used there. We state conjectures relating the (π fragment of the) coloured semantics to $\mathcal{N}$-LATS, as a step towards understanding exactly what approximation is involved.

The coloured semantics takes a fixed set col of *colours* (or *initial causes*, or *principals* – we use the terms interchangeably) disjoint from $\mathcal{N}$. Let $\mathsf{C}, \mathsf{D}, \mathsf{K}$ range over subsets of col. We define a coloured box-π calculus by annotating all outputs with sets of colours:

$$P, Q \quad ::= \quad 0 \;\big|\; P \mid Q \;\big|\; \mathsf{C}\!:\!\overline{x}v \;\big|\; xp.P \;\big|\; \boldsymbol{\nu}xP$$

If P is a coloured term we write $|P|$ for the term of the original syntax obtained by erasing all annotations. Conversely, for a term P of the original syntax $\mathsf{C} \circ P$ denotes the term with every particle coloured by C. In the coloured output $\mathsf{C}\!:\!\overline{x}v$ think of C as recording the causal history of the output particle – C is the set (possibly empty) of principals $\mathsf{p} \in \mathsf{C}$ that have affected the particle in the past. In an initial state all outputs might typically be coloured by singleton sets giving their actual principals. The coloured labelled transition relation has the form

$$A \vdash P \xrightarrow{\ell}_{\mathsf{C}} Q$$

where A is a finite set of names, $\mathrm{fn}(P) \subseteq A$, and ℓ is a label as before; it should be read as 'in a state where the names A may be known to P and its environment, process P can do ℓ, *coloured* C, to become Q'. Here C records causal history, giving all the principals which have directly or indirectly contributed to this action. The relation is defined as the smallest relation satisfying the rules in Figure 4.

Consider a coloured trace t and a run r

$$A_0 \vdash \emptyset \circ P \xrightarrow{\ell_1}_{\mathsf{C}_1} R_1 \xrightarrow{\ell_2}_{\mathsf{C}_2} R_2 \ldots \xrightarrow{\ell_n}_{\mathsf{C}_n} R_n, \qquad A_0 \vdash \mathsf{s}_0 \xrightarrow[e_1]{\ell_1} \mathsf{s}_1 \xrightarrow[e_2]{\ell_2} \mathsf{s}_2 \cdots \xrightarrow[e_n]{\ell_n} \mathsf{s}_n$$

Given a run r and an arbitrary equivalence relation $\simeq$ on the set $\{\, e_i \mid \exists x, v \,.\, \ell_i = xv \,\}$ and col the equivalence classes, define a coloured trace $r/\simeq$ (without the process parts) with labels

$\ell_1 \ldots \ell_n$, with inputs ℓ_j coloured $[e_j]_\simeq$, outputs ℓ_j coloured by $\bigcup\{\, [e_i]_\simeq \mid i$ an input and $i \trianglelefteq^r_{\mathcal{I}} j \,\}$ and taus coloured $\emptyset$. We conjecture that given r and an arbitrary equivalence relation $\simeq$ there is a coloured trace $r/\simeq$, and that given t there exists a run r and an equivalence relation $\simeq$ such that $t = r/\simeq$.

A Proof of Theorem 5.6

In this appendix we provide all the necessary auxiliary results for and the proof of Theorem 5.6 whose text, we recall, is the following:

> Let P and Q be two terms of the π-calculus with free names in I and let $[\![P]\!]^c_I$ and $[\![Q]\!]^c_I$ be their interpretations as $\mathcal{N}$-LATS's. Then $[\![P]\!]^c_I$ is history preserving bisimilar to $[\![Q]\!]^c_I$ if and only if P is strongly causal bisimilar to Q.

NOTATION: Throughout this appendix we write P, Q for generic π-terms, regarded as causal π-terms and C (possibly indexed) for generic causal π-terms.

We then begin with some auxiliary definitions and lemmas.

Definition A.1 *If P is a π-term with free names in I, define $\mathrm{Run}(P)_I$ to be the set of runs of P with initial name set I. Given a causal run $r = I \vdash P \xrightarrow[K_1;k_1]{\ell_1} C_1 \xrightarrow[K_2;k_2]{\ell_2} C_2 \longrightarrow \cdots \xrightarrow[K_n;k_n]{\ell_n} C_n$ define*

1. $\mathrm{end}(r) \stackrel{\mathrm{def}}{=} C_n$
2. $\mathrm{label}(r) \stackrel{\mathrm{def}}{=} \{\ell_j \mid 1 \leq j \leq n\}$
3. $\mathrm{names}(r) \stackrel{\mathrm{def}}{=} I \cup \bigcup_{\ell \in \mathrm{label}(r)} \mathrm{fn}(\ell)$
4. $\mathrm{po}(r)_{\mathcal{I}}$ *to be the following labelled partial order: $([n], \leq_c, l)$, where $l(i) = \ell_i$ and $i \leq_c j$ if $k_i \in K_j \cup \{k_j\}$.*
5. *for any set of causes $K \subseteq \mathcal{K}(C_n)$, the* cause-closure *of K with respect to r, $\overline{K} \stackrel{\mathrm{def}}{=} K \cup \bigcup_{k_j \in K} K_j$*

To prove that the relation $\leq_c$ is a partial order it suffices to prove the following theorem:

Theorem A.2 *Given a causal-π run, $I \vdash P \xrightarrow[K_1;k_1]{\ell_1} C_1 \xrightarrow[K_2;k_2]{\ell_2} C_2 \longrightarrow \cdots \xrightarrow[K_n;k_n]{\ell_n} C_n$, for every $i < n$, if $k_i \in K_n$, then $K_i \subseteq K_n$.*

The proof requires a new definition and a lemma.

Definition A.3 *For every causal-π term, C and cause k, define $k \downarrow C$ to be the set of sets of causes inductively defined as follows:*

$$
\begin{array}{cc}
k \downarrow 0 = \emptyset & k \downarrow \mu.C = k \downarrow C \\
k \downarrow (\boldsymbol{\nu} x)C = k \downarrow C & k \downarrow (C_1 + C_2) = (k \downarrow C_1) \cup (k \downarrow C_2) \\
k \downarrow (C_1 \mid C_2) = (k \downarrow C_1) \cup (k \downarrow C_2) & k \downarrow K :: C = \begin{cases} \{K \cup K' \mid K' \in k \downarrow C\} & \text{if } k \notin K \\ \{K \cup K' \mid K' \in k \downarrow C\} \cup \{K\} & \text{otherwise,} \end{cases}
\end{array}
$$

where μ is any output, input or τ prefix.

The set $k \downarrow C$ aims at syntactically identify the sets of causes which precede (occurrences of) k in the partial order. A few useful properties can be proved to hold:

Lemma A.4 *If, for every set X and Y, we write $[x \rightsquigarrow Y]X$, for the set $(X \setminus \{x\}) \cup Y$ if $x \in X$ and X if $x \notin X$, then $k \downarrow [k \rightsquigarrow K]C = \{[k \rightsquigarrow K]X \mid X \in k \downarrow C\}$, for any causal π-term C, cause k and cause set K with $k \in K$.*

Proof: The proof is a straightforward structural induction.

□

Lemma A.5 *For any causal π-transition $A \vdash C \xrightarrow[K;k]{\ell} C'$ the following facts hold:*

1. $k \downarrow C' = \{K \cup \{k\}\}$
2. *for all $\hat{k} \in K$, there exists $\widehat{K} \in \hat{k} \downarrow C$ such that $\widehat{K} \subseteq K$*
3. *for all $\hat{k} \neq k$ and for all all $K' \in \hat{k} \downarrow C'$, there exists $\widehat{K} \in \hat{k} \downarrow C$ such that $\widehat{K} \subseteq K'$*

Proof: The proof is an easy rule induction. Point (1) requires point (A.4) to resolve the case of the rule COM.

□

Equipped with the Lemma A.5 it is now easy to prove the Theorem A.2.

Proof:[of Theorem A.2] The proof is by induction on the length of the run. The base case, i.e. the length is 0, is obviously trivial. For the inductive step, suppose $k_i \in K_n$ (with $n > 0$), then by Lemma A.5(2) there exists $\widehat{K} \in k_i \downarrow C_{n-1}$ such that $\widehat{K} \subseteq K_n$. By induction on $(n-1-i)$ it is now easy to show that $K_i \subseteq \widehat{K}$ and thus that $K_i \subseteq K_n$. In fact, if $n-1-i = 0$ then $\widehat{K} = K_i \cup \{k_i\}$, by Lemma A.5(1), otherwise, by Lemma A.5(3), there exists $\widehat{\widehat{K}} \in k_i \downarrow C_{n-2}$ such that $\widehat{\widehat{K}} \subseteq \widehat{K}$ and by inductive hypothesis $K_i \subseteq \widehat{\widehat{K}}$.

□

We define now the notion of *run-bisimulation*, relating the behaviour of causal π-processes and $\mathcal{N}$-LATSs.

NOTATION: If r is a causal run and $\text{names}(r) \vdash \text{end}(r) \xrightarrow[K;k]{\ell} C'$ is a transition, we write $r \xrightarrow[K;k]{\ell} C'$ for the run which extends r with that transition. If r and r' are runs, we write $r \xrightarrow[K;k]{\ell} r'$ if r' extends r with the transition $\text{names}(r) \vdash \text{end}(r) \xrightarrow[K;k]{\ell} \text{end}(r')$.

Definition A.6 *A relation $\mathcal{R} \subseteq \text{Run}(P) \times \text{Run}(T)$ between causal runs of a π-term P with free names in I and runs of an $\mathcal{N}$-LATS T with initial name-set I is a* run-bisimulation *if*

1. $(I \vdash P, I \vdash \mathsf{i}) \in \mathcal{R}$
2. $(r, \rho) \in \mathcal{R}$ *implies* $\text{po}(r)_{\mathcal{I}} = \text{po}(\rho)_{\mathcal{I}}$
3. $(r, \rho) \in \mathcal{R}$ *and* $r \xrightarrow[K;k]{\ell} r'$ *implies that there exists an $e \in E$ such that* $\rho \xrightarrow[e]{\ell} \rho'$ *and* $(r', \rho') \in \mathcal{R}$
4. $(r, \rho) \in \mathcal{R}$ *and* $\rho \xrightarrow[e]{\ell} \rho'$, *implies that there exist K and k such that* $r \xrightarrow[K;k]{\ell} r'$ *and* $(r', \rho') \in \mathcal{R}$.

Causal bisimulations in π, gives rise to relations between runs of processes:

Definition A.7 *If $\mathcal{B}$ is a causal bisimulation, such that $P\mathcal{B}Q$, define*

$$\mathrm{Run}(\mathcal{B}) \subseteq \mathrm{Run}(P) \times \mathrm{Run}(Q)$$

to be the following relation between causal runs of P and Q respectively:
if $r = I \vdash P \xrightarrow[K_1;k_1]{\ell_1} C_1 \xrightarrow[K_2;k_2]{\ell_2} C_2 \longrightarrow \cdots \xrightarrow[K_n;k_n]{\ell_n} C_n$ and $s = I \vdash Q \xrightarrow[K_1;k_1]{\ell_1} D_1 \xrightarrow[K_2;k_2]{\ell_2} D_2 \longrightarrow \cdots \xrightarrow[K_n;k_n]{\ell_n} D_n$ are causal runs (notice the coincidence of initial name-set, labels and causes), then $r\,\mathrm{Run}(\mathcal{B})\,s$ if for all $i \leq n$ it is the case that $C_i \mathcal{B} D_i$.

Run-bisimulations and causal bisimulations originate history-preserving bisimulations between $\mathcal{N}$-LATSs, while run-bisimulations and history preserving bisimulations originate causal bisimulations.

Lemma A.8 *Let $\mathcal{R} \subseteq \mathrm{Run}(P) \times \mathrm{Run}(T)$ and $\mathcal{S} \subseteq \mathrm{Run}(Q) \times \mathrm{Run}(U)$ be two run-bisimulations. Then the following two facts hold:*

1. *If $\mathcal{B}$ is a causal bisimulation relating P and Q, then $\mathcal{S} \circ \mathrm{Run}(\mathcal{B}) \circ \mathcal{R}^\circ$, where $\mathcal{R}^\circ$ is the relation opposite of $\mathcal{R}$, is an history-preserving bisimulation.*
2. *If $\mathcal{B} \subseteq \mathrm{Run}(T) \times \mathrm{Run}(U)$ is an history preserving bisimulation, then the symmetric closure of the relation*

$$\{(C, D) \mid \exists r, r'.\, r\,(\mathcal{S}^\circ \circ \mathcal{B} \circ \mathcal{R})\,r' \wedge \mathrm{end}(r) = C \wedge \mathrm{end}(r') = D\}$$

is a causal bisimulation, which obviously relates P and Q.

Proof: Straightforward.

□

In order to prove Theorem 5.6, we then show that for every process term P, with free names in I, there is a run-bisimulation between $\mathrm{Run}(P)_I$ and $\mathrm{Run}([\![P]\!]^c_I)$. Combining this with Lemma A.8 will then give us a proof of the Theorem.

Lemma A.9 *Let P be a π-term with free names in I, then there exists a run-bisimulation $\mathcal{R}^P \subseteq \mathrm{Run}(P)_I \times \mathrm{Run}([\![P]\!]^c_I)$.*

Proof: The proof is by induction on the size of P, where the size of the process is defined as its number of process constructors. The base case is trivially provided by the unique process of size one, namely the nil process. One then shows how to build run bisimulations for compound process terms out of run bisimulation of processes of smaller size.

Most cases are straightforward, we consider here only the two delicate ones, i.e. when a term of size greater than 1 is either a restriction or a parallel composition of terms.

Restriction Suppose $P = (\boldsymbol{\nu} x)Q$ and assume, by inductive hypothesis, the existence for each $\hat{x} \in \mathcal{N} \setminus I$ of a run-bisimulation $\mathcal{R}^{\{\hat{x}/x\}Q} \subseteq \mathrm{Run}(\{\hat{x}/x\}Q)_{I,\hat{x}} \times \mathrm{Run}([\![\{\hat{x}/x\}Q]\!]^c_{I,\hat{x}})$. Let also $[\![\{\hat{x}/x\}Q]\!]^c_{I,\hat{x}}$ be the tuple $\langle S_{\hat{x}}, \longrightarrow_{\hat{x}}, \mathsf{i}_{\hat{x}}, E_{\hat{x}}, \mathcal{I}_{\hat{x}}\rangle$. Define $\mathcal{R}$ to be the smallest set of 4-tuples

$$\langle r, \rho, \hat{r}, \hat{\rho}\rangle \in \bigcup_{\hat{x} \in \mathcal{N} \setminus I} (\mathrm{Run}(\{\hat{x}/x\}Q)_{I,\hat{x}} \times \mathrm{Run}([\![\{\hat{x}/x\}Q]\!]^c_{I,\hat{x}})) \times \mathrm{Run}(P)_I \times \mathrm{Run}([\![P]\!]^c_I)$$

such that:

- $\langle (I, \hat{x}) \vdash \{\hat{x}/x\}Q,\ (I, \hat{x}) \vdash \mathsf{i}_{\hat{x}},\ I \vdash P,\ I \vdash \mathsf{r}[\mathsf{i}_{\hat{x}}]_{\leftrightarrow I}\rangle \in \mathcal{R}$, for each $\hat{x} \in \mathcal{N} \setminus I$

- if $\langle r, \rho, \hat{r}, \hat{\rho} \rangle \in \mathcal{R}$ then the following two conditions are satisfied for every label ℓ, cause set K, cause k, event e, causal-π agent C and state s:

  1. if $\langle r \xrightarrow[K;k]{\ell} C, \rho \xrightarrow[e]{\ell}_{\hat{x}} \mathsf{s} \rangle \in \mathcal{R}^{\{\hat{x}/x\}Q}$, and if for every $\ell' \in \mathrm{label}(r) \cup \{\ell\}$, $\hat{x} \notin \mathrm{fn}(\ell')$ then

     $$\langle r \xrightarrow[K;k]{\ell} C,\ \rho \xrightarrow[e]{\ell}_{\hat{x}} \mathsf{s}\,, \hat{r} \xrightarrow[K;k]{\ell} (\boldsymbol{\nu} \hat{x})C,\ \hat{\rho} \xrightarrow[e]{\ell} \mathsf{r}[\mathsf{s}]_{\leftrightarrow A} \rangle \in \mathcal{R}\ ,$$

     where $A = \mathrm{names}(r) \cup \mathrm{fn}(\ell)$.

  2. if $\langle r \xrightarrow[K;k]{\ell} C, \rho \xrightarrow[e]{\ell}_{\hat{x}} \mathsf{s} \rangle \in \mathcal{R}^{\{\hat{x}/x\}Q}$, and if there exists $\ell' \in \mathrm{label}(r) \cup \{\ell\}$ such that $\ell' = \overline{y}\hat{x}$, for some $y \in N$ then

     $$\langle r \xrightarrow[K;k]{\ell} C,\ \rho \xrightarrow[e]{\ell}_{\hat{x}} \mathsf{s},\ \hat{r} \xrightarrow[K;k]{\ell} C,\ \hat{\rho} \xrightarrow[e]{\ell} \mathsf{l}\mathsf{s} \rangle \in \mathcal{R}$$

Define now $\mathcal{R}^P \stackrel{\mathrm{def}}{=} \{\langle \hat{r}, \hat{\rho} \rangle \mid \exists\, r, \rho.\ \langle r, \rho, \hat{r}, \hat{\rho} \rangle \in \mathcal{R}\}$. It is not difficult to verify that $\mathcal{R}^P$ is a run-bisimulation.

Parallel composition Suppose $P = P_1 \mid P_2$ and that $\mathcal{R}^{P_1}$ and $\mathcal{R}^{P_2}$ are run-bisimulations. Define $\mathcal{R}$ to be the smallest set of 7-tuples $\langle Z, r_1, \rho_1, r_2, \rho_2, r, \rho \rangle$ in

$$\mathcal{P}_{fin}(\mathcal{N}) \times \mathrm{Run}(P_1)_I \times \mathrm{Run}(\llbracket P_1 \rrbracket_I^c) \times \mathrm{Run}(P_2)_I \times \mathrm{Run}(\llbracket P_2 \rrbracket_I^c) \times \mathrm{Run}(P)_I \times \mathrm{Run}(\llbracket P \rrbracket_I^c)$$

such that

- $\langle \emptyset, I \vdash P_1, I \vdash \mathsf{i}_1, I \vdash P_2, I \vdash \mathsf{i}_2, I \vdash P, I \vdash \mathsf{i}_1 \mid_I \mathsf{i}_2 \rangle \in \mathcal{R}$
- if $\langle Z, r_1, \rho_1, r_2, \rho_2, r, \rho \rangle \in \mathcal{R}$ then the following five conditions and their obvious symmetric counterparts hold for any (possibly subscripted) label ℓ, cause set K, cause k, event e, causal-π agent C, state s and names $x, y \in \mathcal{N}$:

  1. if $\langle r_1 \xrightarrow[K;k]{\ell} C, \rho_1 \xrightarrow[e]{\ell} \mathsf{s} \rangle \in \mathcal{R}^{P_1}$ and $k \notin \mathcal{K}(\mathrm{end}(r))$ and $\mathrm{chan}(\ell) \cap Z = \emptyset$ and $\mathrm{val}(\ell) \cap (Z \cup \mathrm{names}(r_2) \setminus \mathrm{names}(r_1)) = \emptyset$ then

     $$\langle Z, r_1 \xrightarrow[K;k]{\ell} C, \rho \xrightarrow[e]{\ell} \mathsf{s}, r_2, \rho_2, r \xrightarrow[\overline{K};k]{\ell} (\boldsymbol{\nu} Z)(C \mid \mathrm{end}(r_2)), \rho \xrightarrow[\langle e,* \rangle]{\ell} (\jmath_1 \mathsf{s} \mid_A \jmath_2 \mathrm{end}(\rho_2)) \rangle \in \mathcal{R}\ ,$$

     where $\mathrm{names}(r_1) \cup \mathrm{val}(\ell) \stackrel{\jmath_1}{\hookrightarrow} \mathrm{names}(r_1) \cup \mathrm{val}(\ell) \cup \mathrm{names}(r_2) \stackrel{\jmath_2}{\hookleftarrow} \mathrm{names}(r_2)$ and $A = \mathrm{names}(r) \cup \mathrm{val}(\ell)$.

  2. if $\langle r_1 \xrightarrow[K;k]{xy} C, \rho \xrightarrow[e]{xy} \mathsf{s} \rangle \in \mathcal{R}^{P_1}$ and $k \notin \mathcal{K}(\mathrm{end}(r))$ and $x \notin Z$ and $y \in (\mathrm{names}(r_2) \setminus \mathrm{names}(r_1))$ then

     $$\langle Z, r_1 \xrightarrow[K;k]{xy} C, \rho \xrightarrow[e]{xy} \mathsf{s}, r_2, \rho_2, r \xrightarrow[\overline{K};k]{xy} (\boldsymbol{\nu} Z)(C \mid \mathrm{end}(r_2)), \rho \xrightarrow[\langle e,* \rangle]{xy} (\jmath_1 \mathsf{s} \mid_A \jmath_2 \mathrm{end}(\rho_2)) \rangle \in \mathcal{R}\ ,$$

     where $\mathrm{names}(r_1) \cup \{y\} \stackrel{\jmath_1}{\hookrightarrow} \mathrm{names}(r_1) \cup \mathrm{names}(r_2) \stackrel{\jmath_2}{\hookleftarrow} \mathrm{names}(r_2)$ and $A = \mathrm{names}(r)$.

3. if $\langle r_1 \xrightarrow[K;k]{\overline{x}y} C, \rho_1 \xrightarrow[e]{\overline{x}y} \mathsf{s} \rangle \in \mathcal{R}^{P_1}$ and $k \notin \mathcal{K}(\text{end}(r))$ and $x \notin Z$ and $y \in Z$ then

   $$\langle Z \setminus \{y\}, r_1 \xrightarrow[K;k]{\overline{x}y} C, \rho \xrightarrow[e]{\overline{x}y} \mathsf{s}, r_2, \rho_2, r \xrightarrow[\overline{K};k]{\overline{x}y} (\boldsymbol{\nu}(Z \setminus \{y\}))(C \mid \text{end}(r_2)), \rho \xrightarrow[\langle e,* \rangle]{xy} (\jmath_1 \mathsf{s} \mid_A \jmath_2 \text{end}(\rho_2)) \rangle \in \mathcal{R} ,$$

   where $\text{names}(r_1) \overset{\jmath_1}{\hookrightarrow} \text{names}(r_1) \cup \text{names}(r_2) \overset{\jmath_2}{\hookleftarrow} \text{names}(r_2)$ and $A = (\text{names}(r), y)$.

4. if $\langle r_1 \xrightarrow[K_1;k]{\overline{x}y} C_1, \rho_1 \xrightarrow[e_1]{\overline{x}y} \mathsf{s}_1 \rangle \in \mathcal{R}^{P_1}$ and $\langle r_2 \xrightarrow[K_2;k]{xy} C_2, \rho_2 \xrightarrow[e_2]{xy} \mathsf{s}_2 \rangle \in \mathcal{R}^{P_2}$ and $y \notin \text{names}(r_1) \cup \text{names}(r_2)$ then

   $$\langle (Z, y), r_1 \xrightarrow[K_1;k]{\overline{x}y} C_1, \rho_1 \xrightarrow[e_1]{\overline{x}y} \mathsf{s}_1, r_2 \xrightarrow[K_2;k]{xy} C_2, \rho_2 \xrightarrow[e_2]{xy} \mathsf{s}_2, r \xrightarrow[K_1 \cup K_2;k]{\tau} (\boldsymbol{\nu}(Z, y))(C_1 \mid C_2), \rho \xrightarrow[\langle e_1,e_2 \rangle]{\tau} (\jmath_1 \mathsf{s}_1 \mid_A \jmath_2 \mathsf{s}_2) \rangle \in \mathcal{R} ,$$

   where $A = \text{names}(r)$ and

   $$\text{names}(\mathsf{s}_1) = \text{names}(r_1) \cup \{y\} \overset{\jmath_1}{\hookrightarrow} \text{names}(r_1) \cup \{y\} \cup \text{names}(r_2) \overset{\jmath_2}{\hookleftarrow} \text{names}(r_2) \cup \{y\} = \text{names}(\mathsf{s}_2) .$$

5. if $\langle r_1 \xrightarrow[K_1;k]{\overline{x}y} C_1, \rho_1 \xrightarrow[e_1]{\overline{x}y} \mathsf{s}_1 \rangle \in \mathcal{R}^{P_1}$ and $\langle r_2 \xrightarrow[K_2;k]{xy} C_2, \rho_2 \xrightarrow[e_2]{xy} \mathsf{s}_2 \rangle \in \mathcal{R}^{P_2}$ and $y \in (\text{names}(r_1) \setminus \text{names}(r_2)) \cup Z$ then

   $$\langle Z, r_1 \xrightarrow[K_1;k]{\overline{x}y} C_1, \rho_1 \xrightarrow[e_1]{\overline{x}y} \mathsf{s}_1, r_2 \xrightarrow[K_2;k]{xy} C_2, \rho_2 \xrightarrow[e_2]{xy} \mathsf{s}_2, r \xrightarrow[K_1 \cup K_2;k]{\tau} (\boldsymbol{\nu} Z)(C_1 \mid C_2), \rho \xrightarrow[\langle e_1,e_2 \rangle]{\tau} (\jmath_1 \mathsf{s}_1 \mid_A \jmath_2 \mathsf{s}_2) \rangle \in \mathcal{R} ,$$

   where $A = \text{names}(r)$ and

   $$\text{names}(\mathsf{s}_1) = \text{names}(r_1) \overset{\jmath_1}{\hookrightarrow} \text{names}(r_1) \cup \text{names}(r_2) \overset{\jmath_2}{\hookleftarrow} \text{names}(r_2) = \text{names}(\mathsf{s}_2) .$$

Define now

$$\mathcal{R}^P = \{ \langle r, \rho \rangle \mid \exists \langle Z, r_1, \rho_1, r_2, \rho_2, r, \rho \rangle \in \mathcal{R} \} .$$

The five conditions (and their symmetric counterparts) cover all possible cases of how transitions in the components of a (restricted) parallel composition induce transition in the (restricted) parallel composition process itself. Condition 1 considers all possible kind of transitions in one of the components. The restriction on $\text{val}(\ell)$ is to ensure that no names from the set Z are communicated as outputs and that the outputs or inputs of new names, with respect to one component only are also outputs or inputs of a new name in the compound process. Condition 2 and 3 supplement Condition 1: Condition 2 by considering what happens in the cases of the input of a name which is new with respect to one component but not the other; Condition 3 by handling the case of the output of a name in Z. Condition 4 and 5 analyse the possibility of an internal communication between the two components which might lead to the extrusion of the scope of some restricted name.

It is slightly more laborious, compared to the case of restriction, to verify that $\mathcal{R}^P$ is a run bisimulation as the equality between the partial orders induced by related runs is not transparently clear but need to be proved by induction on the length of the (related) runs. We leave to the reader to work out the necessary details.

□

We are finally ready to put all this together and prove Theorem 5.6:

Proof:[of Theorem 5.6] Let P and Q be two π-terms with free names in I. By Lemma A.9, there exist run-bisimulations $\mathcal{R}^P \subseteq \text{Run}(P)_I \times \text{Run}([\![P]\!]^c_I)$ and $\mathcal{R}^Q \subseteq \text{Run}(Q)_I \times \text{Run}([\![Q]\!]^c_I)$. Thus is P is causal bisimilar to Q, by Lemma A.8 (1), $\text{Run}([\![P]\!]^c_I)$ is history preserving bisimilar to $\text{Run}([\![Q]\!]^c_I)$. Vice versa if $\text{Run}([\![P]\!]^c_I)$ is history preserving bisimilar to $\text{Run}([\![Q]\!]^c_I)$, by Lemma A.8 (2), P is causal bisimilar to Q.

□

References

[1] M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *Proceedings of POPL 2001*, Jan. 2001. To appear.

[2] M. Abadi and A. D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Proceedings of the Fourth ACM Conference on Computer and Communications Security, Zürich*, pages 36–47. ACM Press, Apr. 1997.

[3] M. Bednarczyk. *Categories of Asynchronous Systems.* PhD thesis, University of Sussex, 1988.

[4] M. Boreale and R. De Nicola. Testing equivalences for mobile processes. *Information and Computation*, 120:279–303, 1995.

[5] M. Boreale and R. De Nicola. A symbolic semantics for the π-calculus. *Information and Computation*, 126(1):34–52, 1996.

[6] M. Boreale and D. Sangiorgi. A fully abstract semantics for causality in the π-calculus. *Acta Informatica*, 35:353–400, 1998.

[7] G. Boudol. Asynchrony and the π-calculus. Technical Report 1702, INRIA, Sophia Antipolis, 1992.

[8] G. Boudol and I. Castellani. Permutation of transitions: an event structure semantics for CCS and SCCS. In *Proc. of REX School/Workshop*, volume 354 of *Lecture Notes in Computer Science*, pages 411–427, 1988.

[9] N. Busi and R. Gorrieri. A petri net semantics for π-calculus. In I. Lee and S. A. Smolk, editors, *Proceedings of the 6th International Conference on Concurrency Theory, CONCUR '95*, volume 962 of *Lecture Notes in Computer Science*, pages 145–159. Springer-Verlag, 1995.

[10] G. L. Cattani. *Presheaf Models for Concurrency.* PhD thesis, University of Aarhus, 1999.

[11] G. L. Cattani and P. Sewell. Models for name-passing processes: Interleaving and causal (extended abstract). In *LICS 2000, Proceedings of the Fifteenth Annual IEEE Symposium on Logic in Computer Science*, pages 322–333. IEEE Computer Society Press, 2000.

[12] G. L. Cattani, I. Stark, and G. Winskel. Presheaf models for the π-calculus. In *Proceedings of the 7th International Conference on Category Theory and Computer Science, CTCS '97*, number 1290 in Lecture Notes in Computer Science, pages 106–126. Springer-Verlag, 1997.

[13] G. L. Cattani and G. Winskel. Presheaf models for concurrency. In D. van Dalen and M. Bezem, editors, *Computer Science Logic. 10th International Workshop, CSL '96, Annual Conference of the European Association for Computer Science Logic. Selected Papers*, volume 1258 of *Lecture Notes in Computer Science*, pages 58–75. Springer-Verlag, 1997.

[14] G. L. Cattani and G. Winskel. Presheaf models for CCS-like languages. Technical Report 477, Cambridge University Computer Laboratory, 1999. Submitted for publication.

[15] P. Darondeau and P. Degano. Causal trees. In G. Ausiello, M. Dezani-Ciancaglini, and S. R. D. Rocca, editors, *ICALP '89, Sixteenth Colloquium on Automata, Languages and Programming*, volume 372 of *Lecture Notes in Computer Science*, pages 234–248. Springer-Verlag, 1989.

[16] P. Degano, R. De Nicola, and U. Montanari. On the consistency of "truly concurrent" operational and denotational semantics (extended abstract). In *LICS '88, Proceedings of the Third Annual IEEE Symposium on Logic in Computer Science*, pages 133–141. IEEE Computer Society Press, 1988.

[17] P. Degano, R. De Nicola, and U. Montanari. Partial orderings descriptions and observations of nondeterministic concurrent processes. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, number 354 in Lecture Notes in Computer Science, pages 438–496, 1988.

[18] P. Degano and C. Priami. Non-interleaving semantics for mobile processes. *Theoretical Computer Science*, 216(1-2):237–270, 1999.

[19] M. P. Fiore, G. L. Cattani, and G. Winskel. Weak bisimulation and open maps (extended abstract). In LICS '99 [34], pages 67–76.

[20] M. P. Fiore, E. Moggi, and D. Sangiorgi. A fully-abstract model for the π-calculus (extended abstract). In LICS '96 [33], pages 43–54.

[21] M. P. Fiore, G. D. Plotkin, and D. Turi. Abstract syntax and variable binding. In LICS '99 [34], pages 193–202.

[22] M. Gabbay and A. Pitts. A new approach to abstract syntax involving binders. In LICS '99 [34], pages 214–224.

[23] R. v. Glabbeek and U. Goltz. Equivalence notions for concurrent systems and refinement of actions. In *Mathematical Foundations of Computer Science 1989*, number 379 in Lecture Notes in Computer Science, pages 237–248. Springer-Verlag, 1989.

[24] M. Hennessy. A fully abstract denotational semantics for the π-calculus. Technical Report 96:04, School of Cognitive and Computing Sciences, University of Sussex, 1996. To appear in *Theoretical Computer Science*.

[25] T. T. Hildebrandt and V. Sassone. Comparing transition systems with independence and asynchronous transition systems. In U. Montanari and V. Sassone, editors, *CONCUR'96, Proceedings of the 7th International Conference on Concurrency Theory*, volume 1119 of *Lecture Notes in Computer Science*, pages 84–97. Springer-Verlag, 1996.

[26] M. Hofmann. Semantical analysis of higher-order abstract syntax. In LICS '99 [34], pages 204–213.

[27] K. Honda. Behavioural subtyping in name passing synchronisation trees. Available at `http://www.dcs.qmw.ac.uk/~kohei/`, 1999.

[28] K. Honda and M. Tokoro. An object calculus for asynchronous communication. In *Proceedings of ECOOP'91 European Conference on Object-Oriented Programming*, volume 512 of *Lecture Notes in Computer Science*, pages 133–147, 1991.

[29] B. Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.

[30] L. J. Jagadeesan and R. Jagadeesan. Causality and true concurrency: A data-flow analysis of the pi-calculus (extended abstract). In *Proceedings of AMAST '95*, Lecture Notes in Computer Science, pages 277–291. Springer-Verlag, 1995.

[31] A. Joyal, M. Nielsen, and G. Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, 1996.

[32] A. Kiehn. Comparing locality and causality based equivalences. *Acta Informatica*, 31:697–718, 1994.

[33] *LICS '96, Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1996.

[34] *LICS '99, Proceedings of the Fourteenth Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1999.

[35] H. Lin. Symbolic bisimulations and proof systems for the pi-calculus. Technical Report 1994:07, COGS, University of Sussex, 1994.

[36] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, 1971.

[37] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer-Verlag, 1992.

[38] A. W. Mazurkievicz and J. Winkowski, editors. *Proceedings of the 8th International Conference on Concurrency Theory, CONCUR '97*, volume 1243 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[39] R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.

[40] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes. II. *Information and Computation*, 100(1):41–77, 1992.

[41] R. Milner, J. Parrow, and D. Walker. Modal logics for mobile processes. *Theoretical Computer Science*, 114(1):149–171, 1993.

[42] U. Montanari and M. Pistore. Concurrent semantics for the π-calculus. In *MFPS XI, Mathematical Foundations of Programming Semantics, Eleventh Annual Conference*, volume 1 of *ENTCS*, pages 337–356. Elsevier, 1995.

[43] U. Montanari and M. Pistore. π-calculus, structured coalgebras and minimal HD-automata. In M. Nielsen and B. Rovan, editors, *Mathematical Foundations of Computer Science 2000*, volume 1893 of *Lecture Notes in Computer Science*. Springer, 2000.

[44] L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6:85–128, 1998.

[45] P. Quaglia. On the finitary characterization of π-congruences. Technical Report RS-97-52, BRICS, Aarhus University, 1997.

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

[47] D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms.* PhD thesis, University of Edinburgh, 1992.

[48] D. Sangiorgi. π-calculus, internal mobility, and agent-passing calculi. *Theoretical Computer Science*, 167(2):235–274, 1996.

[49] V. Sassone, M. Nielsen, and G. Winskel. Models for concurrency: towards a classification. *Theoretical Computer Science*, 170(1-2):297–348, 1996.

[50] P. Selinger. First order axioms for asynchrony. In Mazurkievicz and Winkowski [38], pages 376–390.

[51] P. Sewell. On implementations and semantics of a concurrent programming language. In Mazurkievicz and Winkowski [38], pages 391–405.

[52] P. Sewell. Pi calculi. In H. Bowman and J. Derrick, editors, *Formal Methods for Distributed Processing, An Object Oriented Approach.* CUP, 2000. To appear. Extended version as University of Cambridge TR 498, 2000.

[53] P. Sewell and J. Vitek. Secure composition of insecure components. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop. Mordano, Italy*, pages 136–150. IEEE Computer Society, June 1999. Extended version as University of Cambridge TR 463, 1999.

[54] P. Sewell and J. Vitek. Secure composition of untrusted code: Wrappers and causality types. In *Proceedings of CSFW 00: The 13th IEEE Computer Security Foundations Workshop.*, pages 269–284. IEEE Computer Society, July 2000. Extended version as University of Cambridge TR 478, 1999.

[55] M. W. Shields. Concurrent machines. *Theoretical Computer Science*, 28:449–465, 1985.

[56] I. Stark. A fully abstract domain model for the π-calculus. In LICS '96 [33], pages 36–42.

[57] W. Vogler. Generalized OM-bisimulation. *Information and Computation*, 118(1):38–47, 1995.

[58] G. Winskel and M. Nielsen. Models for concurrency. In *Handbook of logic in computer science, Vol. 4*, Oxford Sci. Publ., pages 1–148. Oxford Univ. Press, 1995.