

## PSI-CALCULI REVISITED: CONNECTIVITY AND COMPOSITIONALITY

JOHANNES ÅMAN POHJOLA

Data61/CSIRO, Sydney, Australia, University of New South Wales, Sydney, Australia  
*e-mail address:* johannes.amanpohjola@data61.csiro.au

**ABSTRACT.** Psi-calculi is a parametric framework for process calculi similar to popular pi-calculus extensions such as the explicit fusion calculus, the applied pi-calculus and the spi calculus. Mechanised proofs of standard algebraic and congruence properties of bisimilarity apply to all calculi within the framework.

A limitation of psi-calculi is that communication channels must be symmetric and transitive. In this paper, we give a new operational semantics to psi-calculi that allows us to lift these restrictions and simplify some of the proofs. The key technical innovation is to annotate transitions with a *provenance*—a description of the scope and channel they originate from.

We give mechanised proofs that our extension is conservative, and that the standard algebraic and congruence properties of strong and weak bisimilarity are maintained. We show correspondence with a reduction semantics and barbed bisimulation. We show how a pi-calculus with preorders that was previously beyond the scope of psi-calculi can be captured, and how to encode mixed choice under very strong quality criteria.

### 1. INTRODUCTION

This paper is mainly concerned with *channel connectivity*, by which we mean the relationship that describes which input channels are connected to which output channels in a setting with message-passing concurrency. In the pi-calculus [MPW92], channel connectivity is syntactic identity: in the process

$$\underline{a}(x).P \mid \bar{b}y.Q$$

where one parallel component is waiting to receive on channel  $a$  and the other is waiting to send on channel  $b$ , interaction is possible only if  $a = b$ .

Variants of the pi-calculus may have more interesting channel connectivity. The explicit fusion calculus pi-F [GW00] extends the pi-calculus with a primitive for *fusing* names; once fused, they are treated as being for all purposes one and the same. Channel connectivity is then given by the equivalence closure of the name fusions. For example, if we extend the above example with the fusion  $(a = b)$

$$\underline{a}(x).P \mid \bar{b}y.Q \mid (a = b)$$

*Key words and phrases:* Process algebra, Psi-calculi, Nominal logic, Interactive theorem proving, Bisimulation.

then communication is possible. Other examples may be found in e.g. calculi for wireless communication [NH06], where channel connectivity can be used to directly model the network's topology.

Psi-calculi [BJPV11] is a family of applied process calculi, where standard meta-theoretical results, such as the algebraic laws and congruence properties of bisimulation, have been established once and for all through mechanised proofs [BPW16] for all members of the family. Psi-calculi generalises e.g. the pi-calculus and the explicit fusion calculus in several ways. In place of atomic names it allows channels and messages to be taken from an (almost) freely chosen term language. In place of fusions, it admits the formulas of an (almost) freely chosen logic as first-class processes. Channel connectivity is determined by judgements of said logic, with one restriction: the connectivity thus induced must be symmetric and transitive.

The main contribution of the present paper is a new way to define the semantics of psi-calculi that lets us lift this restriction, without sacrificing any of the algebraic laws and compositionality properties. It is worth noting that this was previously believed to be impossible: Bengtson et al. [BJPV11, p. 14] even offer counterexamples to the effect that without symmetry and transitivity, scope extension is unsound. However, a close reading reveals that these counterexamples apply only to their particular choice of labelled semantics, and do not rule out the possibility that the counterexamples could be invalidated by a rephrasing of the labelled semantics such as ours.

The price we pay for this increased generality is more complicated transition labels: we decorate input and output labels with a *provenance* that keeps track of which prefix a transition originates from. The idea is that if I am an input label and you are an output label, we can communicate if my subject is your provenance, and vice versa. This is offset by other simplifications of the semantics and associated proofs that provenances enable.

**Contributions.** This paper makes the following specific technical contributions:

- We define a new semantics of psi-calculi that lifts the requirement that channel connectivity must be symmetric and transitive, using the novel technical device of provenances. (Section 2)
- We prove that strong and weak bisimulation is a congruence and satisfies the usual algebraic laws such as scope extension. Interestingly, provenances can be ignored for the purpose of bisimulation. These proofs are machine-checked in Nominal Isabelle [Urb08]. (Section 3.1)
- We prove, again using Nominal Isabelle, that this paper's developments constitute a conservative extension of the original psi-calculi. (Section 3.5)
- To further validate our semantics, we define a reduction semantics and strong barbed congruence, and show that they agree with their labelled counterparts. (Section 3.5)
- We capture a pi-calculus with preorders by Hirschhoff et al. [HMS13], that was previously beyond the scope of psi-calculi because of its non-transitive channel connectivity. The bisimilarity we obtain turns out to coincide with that of Hirschhoff et al. (Section 4.1)
- We exploit non-transitive connectivity to show that mixed choice is a derived operator of psi-calculi in a very strong sense: its encoding is fully abstract and satisfies strong operational correspondence. (Section 4.2)

This paper is an extended version of [ÅP19a]. In this version, we have extended many of the meta-theoretical results and the associated Isabelle formalisation from strong to weak bisimulation (Section 3.2). We have also added a discussion of the aforementioned

counterexamples by Bengtson et al. [BJPV11] (Section 3.4), and a more thorough motivation of the design decisions we have made when introducing provenances (Section 3.3). Moreover, the other sections have been edited for detail and clarity. We have opted against including the full proofs; the interested reader is referred to the associated technical report [ÅP19b] and Isabelle formalisation. Isabelle proofs are available online.<sup>1</sup>

## 2. DEFINITIONS

This section introduces core definitions such as syntax and semantics. Many definitions are shared with the original presentation of psi-calculi, so this section also functions as a recapitulation of [BJPV11]. We will highlight the places where the two differ. For readers who desire a gentler introduction to psi-calculi than the present paper offers, we recommend [Joh10].

Psi-calculi is built on the theory of nominal sets [GP02], which allows us to reason formally up to alpha-equivalence without committing to any particular syntax of the term language. We assume a countable set of *names*  $\mathcal{N}$  ranged over by  $a, b, c, \dots, x, y, z$ . A *nominal set* is a set equipped with a permutation action  $\cdot$ ; intuitively, if  $X \in \mathbf{X}$  and  $\mathbf{X}$  is a nominal set, then  $(x y) \cdot X$ , which denotes  $X$  with all occurrences of the name  $x$  swapped for  $y$  and vice versa, is also an element of  $\mathbf{X}$ .  $\mathfrak{n}(X)$  (the *support* of  $X$ ) is, intuitively, the set of names such that swapping them changes  $X$ . We write  $a \# X$  (“ $a$  is fresh in  $X$ ”) for  $a \notin \mathfrak{n}(X)$ . We overload  $\#$  to sequences of names:  $\tilde{a} \# X$  means that for each  $a_i$  in  $\tilde{a}$ ,  $a_i \# X$ . Similarly,  $a \# X, Y$  abbreviates  $a \# X \wedge a \# Y$ . A nominal set  $\mathbf{X}$  has *finite support* if for every  $X \in \mathbf{X}$ ,  $\mathfrak{n}(X)$  is finite. A function symbol  $f$  is *equivariant* if  $p \cdot f(x) = f(p \cdot x)$ , where  $p$  is a sequence of permutations  $(x y)$ ; this generalises to  $n$ -ary function symbols in the obvious way. Whenever we define inductive syntax with names, it is implicitly quotiented by permutation of bound names, so e.g.  $(\nu x)\bar{a}\langle x \rangle = (\nu y)\bar{a}\langle y \rangle$  if  $x, y \# a$ .

Psi-calculi is parameterised on an arbitrary term language and a logic of environmental assertions:

**Definition 2.1** (Parameters). A *psi-calculus* is a 7-tuple  $(\mathbf{T}, \mathbf{A}, \mathbf{C}, \vdash, \otimes, \mathbf{1}, \dot{\rightarrow})$  with three finitely supported nominal sets:

- (1)  $\mathbf{T}$ , the *terms*, ranged over by  $M, N, K, L, T$ ;
- (2)  $\mathbf{A}$ , the *assertions*, ranged over by  $\Psi$ ; and
- (3)  $\mathbf{C}$ , the *conditions*, ranged over by  $\varphi$ .

We assume each of the above is equipped with a substitution function  $[- := -]$  that substitutes (sequences of) terms for names. The remaining three parameters are equivariant function symbols written in infix:

$$\begin{aligned} \vdash & : \mathbf{A} \times \mathbf{C} \Rightarrow \mathbf{bool} && \text{(entailment)} \\ \otimes & : \mathbf{A} \times \mathbf{A} \Rightarrow \mathbf{A} && \text{(composition)} \\ \mathbf{1} & : \mathbf{A} && \text{(unit)} \\ \dot{\rightarrow} & : \mathbf{T} \times \mathbf{T} \Rightarrow \mathbf{C} && \text{(channel connectivity)} \end{aligned}$$

Intuitively,  $M \dot{\rightarrow} K$  means the prefix  $M$  can send a message to the prefix  $K$ . The substitution functions must satisfy certain natural criteria wrt. their treatment of names; see [BJPV11] for the details. We use  $\sigma$  to range over substitutions  $[\tilde{x} := \tilde{T}]$ . A substitution  $[\tilde{x} := \tilde{T}]$  is

<sup>1</sup><https://github.com/IlmariReissumies/newpsi>

*well-formed* if  $|\tilde{x}| = |\tilde{T}|$  and  $\tilde{x}$  are pairwise distinct. Unless otherwise specified, we only consider well-formed substitutions.

**Definition 2.2** (Static equivalence). Two assertions  $\Psi, \Psi'$  are *statically equivalent*, written  $\Psi \simeq \Psi'$ , if  $\forall \varphi. \Psi \vdash \varphi \Leftrightarrow \Psi' \vdash \varphi$ .

**Definition 2.3** (Valid parameters). A psi-calculus is *valid* if  $(\mathbf{A}/\simeq, \otimes, \mathbf{1})$  form an abelian monoid.

Note that since the abelian monoid is closed, static equivalence is preserved by composition. Henceforth we will only consider valid psi-calculi. The original presentation of psi-calculi had  $\leftrightarrow$  for channel equivalence in place of our  $\rightarrow$ , and required that channel equivalence be symmetric (formally,  $\Psi \vdash M \leftrightarrow K$  iff  $\Psi \vdash K \leftrightarrow M$ ) and transitive.

**Definition 2.4** (Process syntax). The *processes* (or *agents*)  $\mathbf{P}$ , ranged over by  $P, Q, R$ , are inductively defined by the grammar

$P := \mathbf{0}$	(nil)
$(\Psi)$	(assertion)
$\overline{M} N.P$	(output)
$\underline{M}(\lambda\tilde{x})N.P$	(input)
<b>case</b> $\tilde{\varphi} : \tilde{P}$	(case)
$P \mid Q$	(parallel composition)
$(\nu x)P$	(restriction)
$!P$	(replication)

A process is *assertion-guarded* if all assertions occur underneath an input or output prefix. We require that in  $!P$ ,  $P$  is guarded; that in **case**  $\tilde{\varphi} : \tilde{P}$ , all  $\tilde{P}$  are guarded; and that in  $\underline{M}(\lambda\tilde{x})N.P$  it holds that  $\tilde{x} \subseteq \mathfrak{n}(N)$ . We will use  $P_G, Q_G$  to range over assertion-guarded processes. A process  $P$  is *prefix-guarded* if its outermost operator is an input or output prefix.

Restriction, replication and parallel composition are standard. We lift restriction to sequences of names by letting  $(\nu\tilde{a})P$  abbreviate  $(\nu a_0)(\nu a_1)\dots(\nu a_i)P$ ; in particular,  $(\nu\epsilon)P = P$ .  $\overline{M} N.P$  is a process ready to send the message  $N$  on channel  $M$ , and then continue as  $P$ . Similarly,  $\underline{M}(\lambda\tilde{x})N.P$  is a process ready to receive a message on channel  $M$  that matches the pattern  $(\lambda\tilde{x})N$ . We sometimes write  $\overline{M} N$  to stand for  $\overline{M} N.0$ , and similarly for input. We elide the object  $N$  when it is unimportant.

The process  $(\Psi)$  asserts a fact  $\Psi$  about the environment. Intuitively,  $(\Psi) \mid P$  means that  $P$  executes in an environment where all conditions entailed by  $\Psi$  hold.  $P$  may itself contain assertions that add or retract conditions. Environments can evolve dynamically: as a process reduces, assertions may become unguarded and thus added to the environment. **case**  $\tilde{\varphi} : \tilde{P}$  is a process that may act as any  $P_i$  whose guard  $\varphi_i$  is entailed by the environment. For a discussion of why replication and case must be assertion-guarded we refer to [BJPV11, JBPV10]. We use  $\square$  to denote composition of **case** statements, so e.g. **case**  $\varphi : P \square \varphi' : Q$  desugars to **case**  $\varphi, \varphi' : P, Q$ .

The assertion environment of a process is described by its *frame*:

**Definition 2.5** (Frames). The *frame* of  $P$ , written  $\mathcal{F}(P) = (\nu \tilde{b}_P)\Psi_P$  where  $\tilde{b}_P$  bind into  $\Psi_P$ , is defined as

$$\begin{aligned} \mathcal{F}(\langle \Psi \rangle) &= (\nu \epsilon)\Psi \\ \mathcal{F}(P \mid Q) &= \mathcal{F}(P) \otimes \mathcal{F}(Q) \\ \mathcal{F}(\nu x P) &= (\nu x)\mathcal{F}(P) \\ \mathcal{F}(P) &= \mathbf{1} \quad \text{otherwise} \end{aligned}$$

where name-binding and composition of frames is defined as  $(\nu x)(\nu \tilde{b}_P)\Psi_P = (\nu x, \tilde{b}_P)\Psi_P$ , and, if  $\tilde{b}_P \# \tilde{b}_Q, \Psi_Q$  and  $\tilde{b}_Q \# \Psi_P$ ,

$$(\nu \tilde{b}_P)\Psi_P \otimes (\nu \tilde{b}_Q)\Psi_Q = (\nu \tilde{b}_P, \tilde{b}_Q)(\Psi_P \otimes \Psi_Q)$$

where  $\nu$  binds stronger than  $\otimes$ . We overload  $\Psi$  to denote the frame  $(\nu \epsilon)\Psi$ .

We extend entailment to frames as follows:  $\mathcal{F}(P) \vdash \varphi$  holds if, for some  $\tilde{b}_P, \Psi_P$  such that  $\mathcal{F}(P) = (\nu \tilde{b}_P)\Psi_P$  and  $\tilde{b}_P \# \varphi, \Psi_P \vdash \varphi$ . The freshness side-condition  $\tilde{b}_P \# \varphi$  is important because it allows assertions to be used for representing local state. By default, the assertion environment is effectively a form of global non-monotonic state, which is not always appropriate for modelling distributed processes. With  $\nu$ -binding we can recover locality by writing e.g.  $(\nu x)(\langle x = M \rangle \mid P)$  for a process  $P$  with a local variable  $x$ .

The notion of *provenance* is the main novelty of our semantics. It is the key technical device used to make our semantics compositional:

**Definition 2.6** (Provenances). The *provenances*  $\Pi$ , ranged over by  $\pi$ , are either  $\perp$  or of form  $(\nu \tilde{x}; \tilde{y})M$ , where  $M$  is a term, and  $\tilde{x}, \tilde{y}$  bind into  $M$ .

We write  $M$  for  $(\nu \epsilon; \epsilon)M$ . When  $\tilde{x}, \tilde{y} \# \tilde{x}', \tilde{y}'$  and  $\tilde{x} \# \tilde{y}$ , we interpret the expression  $(\nu \tilde{x}; \tilde{y})(\nu \tilde{x}'; \tilde{y}')M$  as  $(\nu \tilde{x} \tilde{x}'; \tilde{y} \tilde{y}')M$ . Furthermore, we equate  $(\nu \tilde{x}; \tilde{y})\perp$  and  $\perp$ . Let  $\pi \downarrow$  denote the result of moving all binders from the outermost binding sequence to the innermost; that is,  $(\nu \tilde{x}; \tilde{y})M \downarrow = (\nu \epsilon; \tilde{x}, \tilde{y})M$ . Similarly,  $\pi \downarrow \tilde{z}$  denotes the result of inserting  $\tilde{z}$  at the end of the outermost binding sequence: formally,  $(\nu \tilde{x}; \tilde{y})M \downarrow \tilde{z} = (\nu \tilde{x}, \tilde{z}; \tilde{y})M$ .

Intuitively, a provenance describes the origin of an input or output transition. For example, if an output transition is annotated with  $(\nu \tilde{x}; \tilde{y})M$ , the sender is an output prefix with subject  $M$  that occurs underneath the  $\nu$ -binders  $\tilde{x}, \tilde{y}$ . For technical reasons, these binders are partitioned into two distinct sequences. The intention is that  $\tilde{x}$  are the frame binders, while  $\tilde{y}$  contains binders that occur underneath case and replication; these are not part of the frame, but may nonetheless bind into  $M$ . We prefer to keep them separate because the  $\tilde{x}$  binders are used for deriving  $\vdash$  judgements, but  $\tilde{y}$  are not (cf. Definition 2.5).

**Definition 2.7** (Labels). The *labels*  $\mathbf{L}$ , ranged over by  $\alpha, \beta$ , are:

$$\begin{array}{ll} \overline{M} (\nu \tilde{x})N & \text{(output)} \\ \underline{M} N & \text{(input)} \\ \tau & \text{(silent)} \end{array}$$

The bound names of  $\alpha$ , written  $\text{bn}(\alpha)$ , is  $\tilde{x}$  if  $\alpha = \overline{M} (\nu \tilde{x})N$  and  $\epsilon$  otherwise. The subject of  $\alpha$ , written  $\text{subj}(\alpha)$ , is  $M$  if  $\alpha = \overline{M} (\nu \tilde{x})N$  or  $\alpha = \underline{M} N$ . Analogously, the object of  $\alpha$ , written  $\text{obj}(\alpha)$ , is  $N$  if  $\alpha = \overline{M} (\nu \tilde{x})N$  or  $\alpha = \underline{M} N$ .

While the provenance describes the origin of a transition, a label describes how it can interact. For example, a transition labelled with  $\underline{M} N$  indicates readiness to receive a message  $N$  from an output prefix with subject  $M$ .

$\text{IN} \frac{\Psi \vdash K \dot{\rightarrow} M}{\Psi \triangleright \underline{M}(\lambda \tilde{y})N.P \xrightarrow[\underline{M}]{\underline{KN}[\tilde{y}:=\tilde{L}]} P[\tilde{y}:=\tilde{L}]}$	$\text{OUT} \frac{\Psi \vdash M \dot{\rightarrow} K}{\Psi \triangleright \overline{M}N.P \xrightarrow[\underline{M}]{\overline{KN}} P}$
$\text{PARL} \frac{\Psi_Q \otimes \Psi \triangleright P \xrightarrow[\pi]{\alpha} P'}{\Psi \triangleright P \mid Q \xrightarrow[\pi \downarrow \tilde{b}_Q]{\alpha} P' \mid Q} \text{bn}(\alpha) \# Q$	$\text{PARR} \frac{\Psi_P \otimes \Psi \triangleright Q \xrightarrow[\pi]{\alpha} Q'}{\Psi \triangleright P \mid Q \xrightarrow[(\nu \tilde{b}_P)\pi]{\alpha} P \mid Q'} \text{bn}(\alpha) \# P$
$\text{COM} \frac{\Psi_Q \otimes \Psi \triangleright P \xrightarrow[(\nu \tilde{b}_P; \tilde{x})K]{\overline{M}(\nu \tilde{a})N} P' \quad \Psi_P \otimes \Psi \triangleright Q \xrightarrow[(\nu \tilde{b}_Q; \tilde{y})M]{\underline{KN}} Q'}{\Psi \triangleright P \mid Q \xrightarrow[\perp]{\tau} (\nu \tilde{a})(P' \mid Q')} \tilde{a} \# Q$	
$\text{CASE} \frac{\Psi \triangleright P_i \xrightarrow[\pi]{\alpha} P' \quad \Psi \vdash \varphi_i}{\Psi \triangleright \text{case } \tilde{\varphi} : \tilde{P} \xrightarrow[\pi \downarrow]{\alpha} P'}$	$\text{SCOPE} \frac{\Psi \triangleright P \xrightarrow[\pi]{\alpha} P'}{\Psi \triangleright (\nu b)P \xrightarrow[(\nu b)\pi]{\alpha} (\nu b)P'} b \# \alpha, \Psi$
$\text{OPEN} \frac{\Psi \triangleright P \xrightarrow[\pi]{\overline{M}(\nu \tilde{a})N} P' \quad b \# \tilde{a}, \Psi, M}{\Psi \triangleright (\nu b)P \xrightarrow[(\nu b)\pi]{\overline{M}(\nu \tilde{a} \cup \{b\})N} P'} b \in \text{n}(N)$	$\text{REP} \frac{\Psi \triangleright P \mid !P \xrightarrow[\pi]{\alpha} P'}{\Psi \triangleright !P \xrightarrow[\pi \downarrow]{\alpha} P'}$

Table 1: Structured operational semantics. A symmetric version of COM is elided. In the rule COM we assume that  $\mathcal{F}(P) = (\nu \tilde{b}_P)\Psi_P$  and  $\mathcal{F}(Q) = (\nu \tilde{b}_Q)\Psi_Q$  where  $\tilde{b}_P$  is fresh for  $\Psi$  and  $Q$ ,  $\tilde{x}$  is fresh for  $\Psi, \Psi_Q, P$ , and  $\tilde{b}_Q, \tilde{y}$  are similarly fresh. In rule PARL we assume that  $\mathcal{F}(Q) = (\nu \tilde{b}_Q)\Psi_Q$  where  $\tilde{b}_Q$  is fresh for  $\Psi, P, \pi$  and  $\alpha$ . PARR has the same freshness conditions but with the roles of  $P, Q$  swapped. In OPEN the expression  $\tilde{a} \cup \{b\}$  means the sequence  $\tilde{a}$  with  $b$  inserted anywhere.

**Definition 2.8** (Operational semantics). The transition relation  $\longrightarrow \subseteq \mathbf{A} \times \mathbf{P} \times \mathbf{L} \times \Pi \times \mathbf{P}$  is inductively defined by the rules in Table 1. We write  $\Psi \triangleright P \xrightarrow[\pi]{\alpha} P'$  for  $(\Psi, P, \alpha, \pi, P') \in \longrightarrow$ . In transitions,  $\text{bn}(\alpha)$  binds into  $\text{obj}(\alpha)$  and  $P'$ .

Note that to avoid clutter, the freshness conditions for some of the rules are stated in the caption of Table 1.

The operational semantics differs from [BJPV11] mainly by the inclusion of provenances: anything underneath the transition arrows is novel.

The OUT rule states that in an environment where  $M$  is connected to  $K$ , the prefix  $\overline{M}N$  may send a message  $N$  from  $M$  to  $K$ . The IN rule is dual to OUT, but also features pattern-matching. If the message is an instance of the pattern, as witnessed by a substitution, that substitution is applied to the continuation  $P$ .

In the COM rule, we see how provenances are used to determine when two processes can interact. Specifically, a communication between  $P$  and  $Q$  can be derived if  $P$  can send a message to  $M$  using prefix  $K$ , and if  $Q$  can receive a message from  $K$  using prefix  $M$ . Because names occurring in  $M$  and  $K$  may be local to  $P$  and  $Q$  respectively, we must be

careful not to conflate the local names of one with the other; this is why the provenance records all binding names that occur above  $M, K$  in the process syntax. Note that even though we identify frames and provenances up to alpha, the COM rule uses syntactically identical binding sequences  $\tilde{b}_P, \tilde{b}_Q$  in two roles: as frame binders  $\mathcal{F}(P) = (\nu\tilde{b}_P)\Psi_P$ , and as the outermost provenance binders. By thus insisting that these binding sequences are chosen to coincide, we ensure that the  $K$  on  $Q$ 's label really is the same as the  $K$  in  $P$ 's provenance.

It is instructive to compare our COM rule with the original:

$$\text{COM-OLD} \frac{\Psi_Q \otimes \Psi \triangleright P \xrightarrow{\overline{M}(\nu\tilde{a})N} P' \quad \Psi_P \otimes \Psi \triangleright Q \xrightarrow{KN} Q' \quad \Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \dot{\leftrightarrow} K}{\Psi \triangleright P \mid Q \xrightarrow{\tau} (\nu\tilde{a})(P' \mid Q')} \tilde{a}\#Q$$

where  $\mathcal{F}(P) = (\nu\tilde{b}_P)\Psi_P$  and  $\mathcal{F}(Q) = (\nu\tilde{b}_Q)\Psi_Q$  and  $\tilde{b}_P\#\Psi, \tilde{b}_Q, Q, M, P$  and  $\tilde{b}_Q\#\Psi, \tilde{b}_Q, Q, K, P$ . Here we have no way of knowing if  $M$  and  $K$  are able to synchronise other than making a channel equivalence judgement. Hence any derivation involving COM-OLD makes three channel equivalence judgements: once each in IN, OUT and COM-OLD. With COM we only make one — or more accurately, we make the exact same judgement twice, in IN resp. OUT. Eliminating the redundant judgements is crucial: the reason COM-OLD needs associativity and commutativity is to stitch these three judgements together, particularly when one end of a communication is swapped for a bisimilar process that allows the same interaction via different prefixes.

Note also that COM has fewer freshness side-conditions. A particularly unintuitive aspect of COM-OLD is that it requires  $\tilde{b}_P\#M$  and  $\tilde{b}_Q\#K$ , but not  $\tilde{b}_P\#K$  and  $\tilde{b}_Q\#M$ : we would expect that all bound names can be chosen to be distinct from all free names, but adding the missing freshness conditions makes scope extension unsound [Joh10, pp. 56-57]. With COM, it becomes clear why: because  $\tilde{b}_Q$  binds into  $M$ .

All the other rules can fire independently of what the provenance of the premise is. They manipulate the provenance, but only for bookkeeping purposes: in order for the COM rule to be sound, we maintain the invariant that if  $\Psi \triangleright P \xrightarrow[\pi]{\alpha} P'$ , the outer binders of  $\pi$  are precisely the binders of  $\mathcal{F}(P)$ . Otherwise, the rules are exactly the same as in the original psi-calculi.

The reader may notice a curious asymmetry between the treatment of provenance binders in the PARL and PARR rules. This is to ensure that the order of the provenance binders coincides with the order of the frame binders, and in the frame  $\mathcal{F}(P \mid Q)$ , the binders of  $P$  occur syntactically outside the binders of  $Q$  (cf. Definition 2.5).

**Example 2.9.** To illustrate how subjects and provenances interact, we consider a psi-calculus where terms are names, assertions are (finite) sets of names, composition is union, and channel connectivity is defined as follows:

$$\Psi \vdash x \dot{\leftrightarrow} y \quad \text{iff} \quad x, y \in \Psi$$

The intuition here is that there exists a single, shared communication medium through which all communication happens. Processes are allowed to declare aliases for this shared medium by adding them to the assertion environment.

Consider the following processes, where  $(\{x\})$  abbreviates  $(\{x\})$  and  $x \neq y$ :

$$P = (\nu x)(\bar{x} \mid \langle x \rangle) \qquad Q = (\nu y)(\underline{y} \mid \langle y \rangle)$$

Here  $P$  and  $Q$  are sending and receiving, respectively, via locally scoped aliases of the shared communication medium. This example has been used previously [BJPV11], to illustrate why the original psi-calculi needs channel equivalence in all three of the IN, OUT and COM rules. Up to scope extension  $P \mid Q$  is equivalent to

$$(\nu x, y)(\bar{x} \mid \langle x \rangle \mid \underline{y} \mid \langle y \rangle)$$

in which a communication between  $x$  and  $y$  is clearly possible, because  $x$  and  $y$  are connected in the environment  $\{x, y\}$ . Hence a communication must also be possible in  $P \mid Q$ . But the two processes share no free names that can be used as communication subjects;  $P$  cannot do an output action with subject  $x$  because  $x$  is bound, and similarly,  $Q$  cannot do an input with subject  $y$ . The only available option is for each of  $P$  and  $Q$  to derive transitions labelled with the other's prefix:

$$\begin{array}{c} \text{OUT} \frac{\{x, y\} \vdash x \dot{\rightarrow} y}{\{x, y\} \triangleright \bar{x} \xrightarrow{\bar{y}} 0} \\ \text{PAR-L} \frac{}{\{y\} \triangleright \bar{x} \mid \langle x \rangle \xrightarrow{\bar{y}} 0 \mid \langle x \rangle} \\ \text{SCOPE} \frac{}{\{y\} \triangleright P \xrightarrow{(\nu x)\bar{y}} (\nu x)(0 \mid \langle x \rangle)} \end{array} \qquad \begin{array}{c} \text{IN} \frac{\{x, y\} \vdash x \dot{\rightarrow} y}{\{x, y\} \triangleright \underline{y} \xrightarrow{\underline{x}} 0} \\ \text{PAR-L} \frac{}{\{x\} \triangleright \underline{y} \mid \langle y \rangle \xrightarrow{\underline{x}} 0 \mid \langle y \rangle} \\ \text{SCOPE} \frac{}{\{x\} \triangleright Q \xrightarrow{(\nu y)\underline{x}} (\nu y)(0 \mid \langle y \rangle)} \end{array}$$

In the original psi-calculi—where the exact same input and output transitions can be derived, but without the provenance annotations—it is clear that without the extra channel equivalence check in the COM-OLD rule, we could not derive a communication between  $P$  and  $Q$ .

With our provenance semantics the COM rule applies immediately. Note that we have  $\mathcal{F}(P) = (\nu x)\{x\}$  and  $\mathcal{F}(Q) = (\nu y)\{y\}$ , and that  $P$ 's transition matches  $Q$ 's provenance and vice versa:

$$\text{COM} \frac{\{y\} \triangleright P \xrightarrow{(\nu x)\bar{y}} (\nu x)(0 \mid \langle x \rangle) \quad \{x\} \triangleright Q \xrightarrow{(\nu y)\underline{x}} (\nu y)(0 \mid \langle y \rangle)}{\{\} \triangleright P \mid Q \xrightarrow[\perp]{\tau} (\nu x)(0 \mid \langle x \rangle) \mid (\nu y)(0 \mid \langle y \rangle)}$$

**Example 2.10.** This example is intended to illustrate how and why we maintain the invariant that frame and provenance binders coincide, and why it matters that they coincide in the COM rule. To this end, we will consider the process  $P \mid Q$ , where  $P$  and  $Q$  are defined as follows

$$P = (\nu x)((\nu y)(\langle \Psi_P \rangle \mid !(\nu z)\bar{x}z) \quad Q = (\nu ab)(\underline{a}(\lambda c)c.R \mid \underline{b}(\lambda c)c.S \mid \langle \Psi_Q \rangle)$$

and where the composition  $\Psi_P \otimes \Psi_Q$  entails the connectivity judgements  $x \dot{\rightarrow} a$  and  $y \dot{\rightarrow} b$ , but not  $x \dot{\rightarrow} b$  or  $y \dot{\rightarrow} a$ . We also assume  $x, y \# \Psi_Q$  and  $a, b \# \Psi_P$ . Concretely, this can be realised by e.g. extending the setup from Example 2.9 to use a pair of sets instead of a single set, and letting connectivity be membership in the same set.

Let us focus on how we can derive a communication between the subjects  $x$  and  $a$ .



We have that  $\mathcal{F}(P) = (\nu xy)\Psi_P$  and  $\mathcal{F}(Q) = (\nu ab)\Psi_Q$ . In the environment  $\Psi_P$ , we have the following derivation of an input transition from  $Q$ :

$$\begin{array}{c} \text{IN} \frac{\Psi_P \otimes \Psi_Q \vdash x \dot{\rightarrow} a}{\Psi_P \otimes \Psi_Q \triangleright \underline{a}(\lambda c)c.R \xrightarrow[a]{xz} R[c := z]} \\ \text{PAR-L} \frac{}{\Psi_P \triangleright \underline{a}(\lambda c)c.R \mid \underline{b}(\lambda c)c.S \mid (\Psi_Q) \xrightarrow[a]{xz} R[c := z] \mid \underline{b}(\lambda c)c.S \mid (\Psi_Q)} \\ \text{SCOPE} \times 2 \frac{}{\Psi_P \triangleright Q \xrightarrow[(\nu a, b; \epsilon)a]{xz} (\nu ab)(R[c := z] \mid \underline{b}(\lambda c)c.S \mid (\Psi_Q))} \end{array}$$

The corresponding output transition in  $Q$  is derived as follows:

$$\begin{array}{c} \text{OUT} \frac{\Psi_P \otimes \Psi_Q \vdash x \dot{\rightarrow} a}{\Psi_P \otimes \Psi_Q \triangleright \bar{x}z \xrightarrow[x]{\bar{a}z} 0} \\ \text{OPEN} \frac{}{\Psi_P \otimes \Psi_Q \triangleright (\nu z)\bar{x}z \xrightarrow[(\nu z; \epsilon)x]{\bar{a}(\nu z)z} 0} \\ \text{PAR-L} \frac{}{\Psi_P \otimes \Psi_Q \triangleright (\nu z)\bar{x}z \mid !(\nu z)\bar{x}z \xrightarrow[(\nu z; \epsilon)x]{\bar{a}(\nu z)z} 0 \mid !(\nu z)\bar{x}z} \\ \text{REP} \frac{}{\Psi_P \otimes \Psi_Q \triangleright !(\nu z)\bar{x}z \xrightarrow[(\nu \epsilon; z)x]{\bar{a}(\nu z)z} 0 \mid !(\nu z)\bar{x}z} \\ \text{PAR-R} \frac{}{\Psi_Q \triangleright (\nu y)(\Psi_P) \mid !(\nu z)\bar{x}z \xrightarrow[(\nu y; z)x]{\bar{a}(\nu z)z} (\nu y)(\Psi_P) \mid 0 \mid !(\nu z)\bar{x}z} \\ \text{SCOPE} \frac{}{\Psi_Q \triangleright P \xrightarrow[(\nu x, y; z)x]{\bar{a}(\nu z)z} (\nu x)((\nu y)(\Psi_P) \mid 0 \mid !(\nu z)\bar{x}z)} \end{array}$$

In the derivation above, notice how the provenance evolves throughout the derivation to maintain the correspondence between the outer provenance binders and the frame binders. Two rule applications are particularly noteworthy. First, the **REP** rule pushes  $z$  from the outer binders to the inner binders, because binders underneath the replication operator are not considered part of the frame (cf. Definition 2.5). Second, the **PAR-R** rule adds  $y$ , the frame binder of the leftmost parallel component, to the outer binders.

Because both derivations have matching provenances and subjects, and the frame binders and provenance binders used are the same, the **COM** rule allows a derivation as follows:

$$\text{COM} \frac{\Psi_Q \triangleright P \xrightarrow[(\nu x, y; z)x]{\bar{a}(\nu z)z} \dots \quad \Psi_P \triangleright Q \xrightarrow[(\nu a, b; \epsilon)a]{xz} \dots}{\mathbf{1} \triangleright P \mid Q \xrightarrow[\perp]{\tau} (\nu z)(\dots \mid \dots)}$$

In order for this rule to be sound, it is important that the frame binders of  $\mathcal{F}(P)$ , and the provenance binders of the transition from  $P$ , have the same ordering. To see this, suppose we have a version of the **COM** rule which allows transitions to be derived when frame and provenance binders are equal up to reordering. Call this alternative rule **COM'**. We will now argue that **COM'** is unsound, because we lose the ability to distinguish synchronisations between  $x$  and  $a$  from synchronisations between  $x$  and  $b$ .

First, note that since we identify frames up to alpha, we have  $\mathcal{F}(P) = (\nu yx)((x y) \cdot \Psi_P)$ . By equivariance of  $\vdash$  and  $\dot{\rightarrow}$  we have

$$((x y) \cdot \Psi_P) \otimes \Psi_Q \vdash x \dot{\rightarrow} b \qquad ((x y) \cdot \Psi_P) \otimes \Psi_Q \vdash y \dot{\rightarrow} a$$

In this permuted frame, we can derive an input where  $b$  receives from  $x$ :

$$(x y) \cdot \Psi_P \triangleright Q \xrightarrow[(\nu a, b; \epsilon)_b]{xz} (\nu ab)(\underline{a}(\lambda c)c.R \mid S[c := z] \mid (\Psi_Q))$$

Since we identify provenances up to alpha, we also have

$$(x y) \cdot \Psi_P \triangleright Q \xrightarrow[(\nu b, a; \epsilon)_a]{xz} (\nu ab)(\underline{a}(\lambda c)c.R \mid S[c := z] \mid (\Psi_Q))$$

Using this transition, and the same derivation of a transition from  $P$  as above, we can now apply COM' to derive a synchronisation between  $x$  and  $b$ , despite the fact that  $x$  and  $b$  are not connected:

$$\text{COM}' \frac{\Psi_Q \triangleright P \xrightarrow[(\nu x, y; z)_x]{\bar{a}(\nu z)z} \dots \quad (x y) \cdot \Psi_P \triangleright Q \xrightarrow[(\nu b, a; \epsilon)_a]{xz} \dots}{\mathbf{1} \triangleright P \mid Q \xrightarrow[\perp]{\tau} (\nu z)(\dots \mid \dots)}$$

If we push all binders in  $P \mid Q$  to the top level, no such derivation is possible. Thus scope extension fails to hold with COM'.

With COM, the existence of this alternative transition from  $Q$  is unproblematic: it cannot synchronise with any transition from  $P$  unless that transition too uses the permuted frame.

With the same counterexample, we can also see why it is important that the provenance retains all of the frame binders, even the vacuous ones: the provenances  $(\nu x)x$  and  $(\nu y)y$  are equal, so the provenance would contain no information about which prefix the transition originates from.

### 3. META-THEORY

In this section, we will derive the standard algebraic and congruence laws of strong and weak bisimulation, develop an alternative formulation of strong bisimulation in terms of a reduction relation and barbed congruence, and show that our extension of psi-calculi is conservative.

**3.1. Strong bisimulation.** We write  $\Psi \triangleright P \xrightarrow{\alpha} P'$  as shorthand for  $\exists \pi. \Psi \triangleright P \xrightarrow[\pi]{\alpha} P'$ . Bisimulation is then defined exactly as in the original psi-calculi:

**Definition 3.1** (Strong bisimulation). A symmetric relation  $\mathcal{R} \subseteq \mathbf{A} \times \mathbf{P} \times \mathbf{P}$  is a *strong bisimulation* iff for every  $(\Psi, P, Q) \in \mathcal{R}$

- (1)  $\Psi \otimes \mathcal{F}(P) \simeq \Psi \otimes \mathcal{F}(Q)$  (static equivalence)
- (2)  $\forall \Psi'. (\Psi \otimes \Psi', P, Q) \in \mathcal{R}$  (extension of arbitrary assertion)
- (3) If  $\Psi \triangleright P \xrightarrow{\alpha} P'$  and  $\text{bn}(\alpha) \# \Psi, Q$ , then there exists  $Q'$  such that  $\Psi \triangleright Q \xrightarrow{\alpha} Q'$  and  $(\Psi, P', Q') \in \mathcal{R}$  (simulation)

We let *bisimilarity*  $\dot{\sim}$  be the largest bisimulation. We write  $P \dot{\sim}_\Psi Q$  to mean  $(\Psi, P, Q) \in \dot{\sim}$ , and  $P \dot{\sim} Q$  for  $P \dot{\sim}_1 Q$ .

Clause 3 is the same as for pi-calculus bisimulation. Clause 1 requires that two bisimilar processes expose statically equivalent assertion environments. Clause 2 states that if two processes are bisimilar in an environment, they must be bisimilar in every extension of that environment. Without this clause, bisimulation is not preserved by parallel composition.

This definition might raise some red flags for the experienced concurrency theorist. We allow the matching transition from  $Q$  to have any provenance, irrespective of what  $P$ 's provenance is. Hence the COM rule uses information that is ignored for the purposes of bisimulation, which in most cases would result in a bisimilarity that is not preserved by the parallel operator.

Before showing that bisimilarity is nonetheless compositional, we will argue that bisimilarity would be too strong if Clause 4 required transitions with matching provenances. Consider two distinct terms  $M, N$  that are connected to the same channels; that is, for all  $\Psi, K$  we have  $\Psi \vdash M \dot{\rightarrow} K$  iff  $\Psi \vdash N \dot{\rightarrow} K$ . We would expect  $\overline{M}.0$  and  $\overline{N}.0$  to be bisimilar because they offer the same interaction possibilities. With our definition, they are. But if bisimulation cared about provenance they would be distinguished, because transitions originating from  $\overline{M}.0$  will have provenance  $M$  while those from  $\overline{N}.0$  will have  $N$ .

The key intuition is that what matters is not which provenance a transition has, but which channels the provenance is connected to. The latter is preserved by Clause 3, as this key technical lemma hints at:

**Lemma 3.2.** (*Find connected provenance*)

- (1) If  $\Psi \triangleright P \xrightarrow[\pi]{M N} P'$  and  $C$  is a finitely supported nominal set, then there exists  $\tilde{b}_P, \Psi_P, \tilde{x}, K$  such that  $\mathcal{F}(P) = (\nu \tilde{b}_P) \Psi_P$  and  $\pi = (\nu \tilde{b}_P; \tilde{x}) K$  and  $\tilde{b}_P \# \Psi, P, M, N, P', C, \tilde{x}$  and  $\tilde{x} \# \Psi, P, N, P', C$  and  $\Psi \otimes \Psi_P \vdash M \dot{\rightarrow} K$ .
- (2) A similar property for output transitions (*elided*).

*Proof.* Formally proven in Isabelle, by a routine induction. □

In words, the provenance of a transition is always connected to its subject, and the frame binders can always be chosen sufficiently fresh for any context. This simplifies the proof that bisimilarity is preserved by parallel: in the original psi-calculi, one of the more challenging aspects of this proof is finding sufficiently fresh subjects to use in the COM-OLD rule, and then using associativity and symmetry to connect them (cf. [BJPV11, Lemma 5.11]). By Lemma 3.2 we already have a sufficiently fresh subject: our communication partner's provenance.

**Theorem 3.3** (Congruence properties of strong bisimulation).

- (1)  $P \dot{\sim}_\Psi Q \Rightarrow P \mid R \dot{\sim}_\Psi Q \mid R$
- (2)  $P \dot{\sim}_\Psi Q \Rightarrow (\nu x) P \dot{\sim}_\Psi (\nu x) Q$  if  $x \# \Psi$
- (3)  $P_G \dot{\sim}_\Psi Q_G \Rightarrow !P_G \dot{\sim}_\Psi !Q_G$
- (4)  $\forall i. P_i \dot{\sim}_\Psi Q_i \Rightarrow \mathbf{case} \tilde{\varphi} : \tilde{P} \dot{\sim}_\Psi \mathbf{case} \tilde{\varphi} : \tilde{Q}$  if  $\tilde{P}, \tilde{Q}$  are assertion-guarded
- (5)  $P \dot{\sim}_\Psi Q \Rightarrow \overline{M N}. P \dot{\sim}_\Psi \overline{M N}. Q$

*Proof.* Formally proven in Isabelle. All proofs are by coinduction. The most interesting cases are parallel and replication, where Lemma 3.2 features prominently. We briefly outline

a COM subcase of the replication case, where the candidate relation is

$$\{(\Psi, R \mid !P, R \mid !Q).P \dot{\sim}_{\Psi} Q \wedge P, Q \text{ are assertion guarded}\}$$

Suppose  $P \dot{\sim} Q$  and that  $!P$  derives a  $\tau$  transition from communication between two unfolded copies of  $P$ , with input subject  $M$  and output subject  $K$ . We need to mimic the same communication between two copies of  $Q$ , but after using  $P \dot{\sim}_{\Psi} Q$  to obtain a matching input transition, the subject  $M$  is not useful to derive a communication since it is  $P$ 's provenance, not  $Q$ 's. However, we can obtain eligible subjects  $M', K'$  by repeatedly applying Lemma 3.2.  $\square$

In Theorem 3.3.4,  $P_i$  is the  $i$ :th element of  $\tilde{P}$ , and similarly for  $Q_i$ . The index variable  $i$  ranges over the length of the sequences  $\tilde{\varphi}, \tilde{P}, \tilde{Q}$ , which we assume are equal.

**Theorem 3.4** (Algebraic laws of strong bisimulation).

- (1)  $P \dot{\sim}_{\Psi} P \mid \mathbf{0}$
- (2)  $P \mid (Q \mid R) \dot{\sim}_{\Psi} (P \mid Q) \mid R$
- (3)  $P \mid Q \dot{\sim}_{\Psi} Q \mid P$
- (4)  $(\nu a)\mathbf{0} \dot{\sim}_{\Psi} \mathbf{0}$
- (5)  $P \mid (\nu a)Q \dot{\sim}_{\Psi} (\nu a)(P \mid Q)$  if  $a \# P$
- (6)  $\overline{M} N.(\nu a)P \dot{\sim}_{\Psi} (\nu a)\overline{M} N.P$  if  $a \# M, N$
- (7)  $\underline{M}(\lambda \tilde{x})N.(\nu a)P \dot{\sim}_{\Psi} (\nu a)\underline{M}(\lambda \tilde{x})N.P$  if  $a \# \tilde{x}, M, N$
- (8)  $!P \dot{\sim}_{\Psi} P \mid !P$
- (9) **case**  $\tilde{\varphi} : (\nu a)P \dot{\sim}_{\Psi} (\nu a)$  **case**  $\tilde{\varphi} : \tilde{P}$  if  $a \# \tilde{\varphi}$
- (10)  $(\nu a)(\nu b)P \dot{\sim}_{\Psi} (\nu b)(\nu a)P$

*Proof.* Formally proven in Isabelle. All proofs are by coinduction.  $\square$

Note that bisimilarity is not preserved by input, for the same reasons as the pi-calculus. As in the pi-calculus, we can define *bisimulation congruence* as the substitution closure of bisimilarity, and thus obtain a true congruence which satisfies all the algebraic laws above. We have verified this in Nominal Isabelle, following [BJPV11].

**3.2. Weak bisimulation.** We have also proved the standard algebraic and congruence properties of weak bisimulation. The results in this section were established for the original psi-calculi by Johansson et al. [JBPV10]; our contribution is to lift them to psi-calculi without channel symmetry and transitivity. As for strong bisimulation, it turns out that we may disregard provenances for the purposes of weak bisimulation, so we can reuse the original definitions verbatim.

The definition of weak bisimulation is technically complicated in psi-calculi because of the delicate interplay between assertions and reductions. For example, in the pi-calculus weak bisimulation equates  $P$  and  $\tau.P$ , but in psi-calculi this equation cannot be admitted:  $P$  may contain top-level assertions that disable interaction possibilities in parallel processes. Hence there may be situations where an observable action originating from  $Q$  is available in  $Q \mid \tau.P$  (where  $P$  has not yet disabled it) but unavailable in  $Q \mid P$ .

For a comprehensive motivation of the definitions, we refer to Johansson et al. [JBPV10]; below we restate the pertinent definitions for completeness.

**Definition 3.5** (Weak transitions).  $\Psi \triangleright P \Longrightarrow P'$  means that either  $P = P'$  or there exists  $P''$  such that  $\Psi \triangleright P \xrightarrow{\tau} P''$  and  $\Psi \triangleright P'' \Longrightarrow P'$ . We write  $\Psi \triangleright P \xrightarrow{\alpha} P'$

to mean that there exists  $P'', P'''$  such that  $\Psi \triangleright P \implies P''$  and  $\Psi \triangleright P'' \xrightarrow{\alpha} P'''$  and  $\Psi \triangleright P''' \implies P'$ .

**Definition 3.6.**  $P$  statically implies  $Q$  in the environment  $\Psi$ , written  $P \leq_{\Psi} Q$ , if

$$\forall \varphi. \Psi \otimes \mathcal{F}(P) \vdash \varphi \implies \Psi \otimes \mathcal{F}(Q) \vdash \varphi$$

If  $\Psi = \mathbf{1}$  we may write  $P \leq Q$ .

**Definition 3.7** (Weak bisimulation). A *weak bisimulation*  $\mathcal{R}$  is a ternary relation between assertions and pairs of agents such that  $\mathcal{R}(\Psi, P, Q)$  implies all of

(1) Weak static implication:

$$\begin{aligned} & \forall \Psi' \exists Q'', Q'. \\ & \Psi \triangleright Q \implies Q'' \quad \wedge \quad P \leq_{\Psi} Q'' \quad \wedge \\ & \Psi \otimes \Psi' \triangleright Q'' \implies Q' \quad \wedge \quad \mathcal{R}(\Psi \otimes \Psi', P, Q') \end{aligned}$$

(2) Symmetry:  $\mathcal{R}(\Psi, Q, P)$

(3) Extension of arbitrary assertion:

$$\forall \Psi'. \mathcal{R}(\Psi \otimes \Psi', P, Q)$$

(4) Weak simulation: for all  $\alpha, P'$  such that  $\text{bn}(\alpha) \# \Psi, Q$  and  $\Psi \triangleright P \xrightarrow{\alpha} P'$  it holds

$$\begin{aligned} & \text{if } \alpha = \tau : \exists Q'. \Psi \triangleright Q \implies Q' \quad \wedge \quad \mathcal{R}(\Psi, P', Q') \\ & \text{if } \alpha \neq \tau : \forall \Psi' \exists Q'', Q'''. \\ & \Psi \triangleright Q \implies Q''' \quad \wedge \quad P \leq_{\Psi} Q''' \quad \wedge \\ & \Psi \triangleright Q''' \xrightarrow{\alpha} Q'' \quad \wedge \\ & \exists Q'. \Psi \otimes \Psi' \triangleright Q'' \implies Q' \quad \wedge \quad \mathcal{R}(\Psi \otimes \Psi', P', Q') \end{aligned}$$

We define  $P \dot{\approx}_{\Psi} Q$  to mean that there exists a weak bisimulation  $\mathcal{R}$  such that  $\mathcal{R}(\Psi, P, Q)$  and write  $P \dot{\approx} Q$  for  $P \dot{\approx}_{\mathbf{1}} Q$ .

Weak bisimulation thus defined includes strong bisimulation, and thus satisfies all the usual structural laws. It is not preserved by **case** and input, for the same reasons as  $+$  and input do not preserve weak bisimulation in the pi-calculus. We employ the standard solution to obtain a congruence: all initial  $\tau$  steps must be simulated by at least one  $\tau$  step, and we furthermore close the relation under all substitutions.

**Definition 3.8** (Weak congruence).  $P$  and  $Q$  are weakly  $\tau$ -bisimilar, written  $\Psi \triangleright P \dot{\approx}_{\text{tau}} Q$ ,

if  $P \dot{\approx}_{\Psi} Q$  and the following holds: for all  $P'$  such that  $\Psi \triangleright P \xrightarrow{\tau} P'$  there exists  $Q'$  such that  $\Psi \triangleright Q \xrightarrow{\tau} Q' \wedge P' \dot{\approx}_{\Psi} Q'$ , and similarly with the roles of  $P$  and  $Q$  exchanged. We define  $P \approx Q$  to mean that for all  $\Psi$ , and for all well-formed substitution sequences  $\tilde{\sigma}$ , it holds that  $\Psi \triangleright P \tilde{\sigma} \dot{\approx}_{\text{tau}} Q \tilde{\sigma}$ .

The following theorems have been formally proven in Nominal Isabelle:

**Theorem 3.9.**  $\dot{\approx}_{\text{tau}}$  satisfies all the algebraic laws of  $\dot{\approx}$  established in Theorem 3.4.

*Proof.* Formally proven in Isabelle. The proof relies on the fact that  $\dot{\approx} \subseteq \dot{\approx}_{\text{tau}}$ , which we show by coinduction, using  $\dot{\approx}$  as a candidate relation.  $\square$

**Theorem 3.10.**  $\dot{\approx}$  satisfies all the congruence properties of  $\dot{\sim}$  established in Theorem 3.3 except 3.3.4.

*Proof.* Formally proven in Isabelle, by coinduction. □

**Theorem 3.11.** Weak congruence  $\approx$  is a congruence wrt. all operators of psi-calculi.

*Proof.* Formally proven in Isabelle, using Theorem 3.10 where applicable. □

**3.3. Motivating the design.** We have added provenance annotations to an operational semantics that had no shortage of annotations and side-conditions to begin with. The end result may strike the reader as somewhat unparsimonious. Previously, psi-calculi had one label component—the channel subjects—for keeping track of connectivity. We now have two; do we really need both? In this section, we will explore the consequences of removing either channel subjects or provenances from the semantics. The short answer is that while we *can* do without either one, the end result is not greater parsimony.

**3.3.1. Do we need provenances?** The fact that bisimilarity is compositional yet ignores provenances suggests that the semantics could be reformulated without provenance annotations on labels. To achieve this, what is needed is a side-condition  $S$  for the COM rule which, given an input and an output with subjects  $M, K$ , determines if the input transition could have been derived from prefix  $K$ , and vice versa:

$$\frac{\Psi_Q \otimes \Psi \triangleright P \xrightarrow{\overline{M}(\nu\tilde{a})N} P' \quad \Psi_P \otimes \Psi \triangleright Q \xrightarrow{\underline{K}N} Q' \quad S}{\Psi \triangleright P \mid Q \xrightarrow{\tau} (\nu\tilde{a})(P' \mid Q')}$$

But we already have such an  $S$ : the semantics *with* provenances! So we can let

$$S = \Psi_Q \otimes \Psi \triangleright P \xrightarrow[\substack{(\nu\tilde{b}_P;\tilde{x})K}{\overline{M}(\nu\tilde{a})N}} P' \wedge \Psi_P \otimes \Psi \triangleright Q \xrightarrow[\substack{(\nu\tilde{b}_Q;\tilde{y})M}{\underline{K}N}] Q'$$

Of course, this definition is not satisfactory: the provenances are still there, just swept under the carpet. Worse, we significantly complicate the definitions by effectively introducing a stratified semantics. Thus the interesting question is not whether such an  $S$  exists (it does), but whether  $S$  can be formulated in a way that is significantly simpler than the semantics with provenances. The author believes the answer is negative:  $S$  is a property about the roots of the proof trees used to derive the transitions from  $P$  and  $Q$ . The provenance records just enough information about the proof trees to show that  $M$  and  $K$  are connected; with no provenances, it is not clear how this information could be obtained without essentially reconstructing the proof tree.

Another alternative is to use the proof tree itself as the transition label [BC88, DP92]. This makes the necessary information available, at the expense of making labels even more complicated.

3.3.2. *Do we need channel subjects?* While we have chosen to test for channel connectivity in the IN and OUT rules, a semantics without channel subjects would defer the connectivity check until the COM rule.<sup>2</sup> Let us call the former approach *early connectivity*, and the latter *late connectivity*. The rules for late connectivity—eliding freshness conditions for readability, and using ! and ? to distinguish outputs from inputs—would be:

$$\begin{array}{c}
\text{IN-LATE} \frac{}{\Psi \triangleright \underline{M}(\lambda\tilde{y})N.P \xrightarrow[M]{?N[\tilde{y}:=\tilde{L}]} P[\tilde{y}:=\tilde{L}]}} \quad \text{OUT-LATE} \frac{}{\Psi \triangleright \overline{M}N.P \xrightarrow[M]{!N} P} \\
\\
\text{COM-LATE} \frac{\Psi \otimes \Psi_P \otimes \Psi_Q \vdash K \dot{\rightarrow} M \quad \Psi_Q \otimes \Psi \triangleright P \xrightarrow[(\nu\tilde{b}_P;\tilde{x})K]{!(\nu\tilde{a})N} P' \quad \Psi_P \otimes \Psi \triangleright Q \xrightarrow[(\nu\tilde{b}_Q;\tilde{y})M]{?N} Q'}{\Psi \triangleright P | Q \xrightarrow[\perp]{\tau} (\nu\tilde{a})(P' | Q')} \tilde{a}\#Q
\end{array}$$

It is pleasant that this formulation allows the IN-LATE and OUT-LATE rules to have no side-conditions. Save for the provenance bookkeeping, all questions of connectivity are handled entirely in COM-LATE, which results in a pleasing separation of concerns. This may seem more parsimonious at first glance, but it introduces two issues that makes the trade-off seem unfavourable: more complicated bisimulation and spurious transitions.

- (1) More complicated bisimulation. Consider a psi-calculus where channel connectivity is syntactic equality; that is, where  $\Psi \vdash M \dot{\rightarrow} K$  holds iff  $M = K$ . Fix  $M, K$  such that  $M \neq K$ . Using bisimilarity as defined in Definition 3.1, we can show that  $\overline{M}.0 \dot{\sim} \overline{K}.0$ : without subjects, these processes emit identical labels save for the provenance, which is ignored by bisimulation. Hence bisimilarity fails to be preserved by the parallel operator: consider these processes in parallel with a process that can receive on  $M$ . Then  $\overline{M}.0$  can communicate but  $\overline{K}.0$  cannot.

The takeaway is that with late connectivity, a compositional notion of bisimulation needs to be more careful with which provenance the mimicking transition may use. The intuition is that rather than admitting any provenance, we admit provenances that are connected to the same channels. We conjecture that the necessary adaptation is:

**Definition 3.12** (Late-connectivity bisimulation). A symmetric relation  $\mathcal{R} \subseteq \mathbf{A} \times \mathbf{P} \times \mathbf{P}$  is a *channel-free bisimulation* iff for every  $(\Psi, P, Q) \in \mathcal{R}$

- (a)  $\Psi \otimes \mathcal{F}(P) \simeq \Psi \otimes \mathcal{F}(Q)$  (static equivalence)
- (b)  $\forall \Psi'. (\Psi \otimes \Psi', P, Q) \in \mathcal{R}$  (extension of arbitrary assertion)
- (c) If  $\Psi \triangleright P \xrightarrow[\pi]{\alpha} P'$  and  $\text{bn}(\alpha) \# \Psi, Q$ , then
  - (i) If  $\alpha = \tau$  then there exists  $Q'$  such that  $\Psi \triangleright Q \xrightarrow[\perp]{\tau} Q'$  and  $(\Psi, P', Q') \in \mathcal{R}$
  - (ii) For all  $M, K, N, \tilde{x}, \tilde{b}_P, \tilde{b}_Q, \Psi_P, \Psi_Q$  such that  $\alpha = ?N$  and  $\pi = (\nu\tilde{b}_P; \tilde{x})K$  and  $\mathcal{F}(P) = (\nu\tilde{b}_P)\Psi_P$  and  $\mathcal{F}(Q) = (\nu\tilde{b}_Q)\Psi_Q$  and  $\tilde{b}_P, \tilde{b}_Q \# \Psi, P, Q, M, \tilde{x}$  and  $\tilde{x} \# \Psi$

<sup>2</sup>This is similar to a design first proposed by Magnus Johansson in an unpublished draft. Johansson's design does not use provenances, but obtains a similar effect by including bound subjects and bound assertions in labels. By partitioning provenance binders in two sequences, we can recover frame binders from labels and thus found no need to include bound assertions.

and  $\Psi \otimes \Psi_P \vdash M \dot{\rightarrow} K$ , then there exists  $\tilde{y}, K', Q'$  such that  $\tilde{y} \# \Psi$  and  $\Psi \otimes \Psi_Q \vdash M \dot{\rightarrow} K'$  and  $\Psi \triangleright Q \xrightarrow[\text{(\nu}\tilde{b}_Q; \tilde{y})K']{?N} Q'$  and  $(\Psi, P', Q') \in \mathcal{R}$

(iii) (A similar clause for output transitions)

In words, for every channel  $M$  that the transition from  $P$ 's provenance is connected to, there must be a transition from  $Q$  with a provenance that is also connected to  $M$ . Note that static equivalence is not sufficient to imply preservation of connectivity: the conditions may be distinct, and even if equal may be obscured by the frame binders.

We find this definition of bisimulation intolerably ad-hoc and complicated.

- (2) Spurious transitions. Consider the representation of the  $\pi$ -calculus in psi-calculi, where  $\mathbf{T} = \mathcal{N}$ , and where channel connectivity is syntactic equality on names. This representation is in one-to-one transition correspondence with the standard presentation of the  $\pi$ -calculus [Joh10], but if we use late connectivity, one-to-one transition correspondence is lost. The pi-calculus process  $(\nu x)\bar{x}y$  should not have any outgoing transitions, but late connectivity semantics allows the derivation of a transition as follows:

$$\text{SCOPE-LATE} \frac{\text{OUT-LATE} \frac{\Psi \triangleright \bar{x}y \xrightarrow[x]{!y} 0}}{\Psi \triangleright (\nu x)\bar{x}y \xrightarrow[(\nu x)x]{!y} (\nu x)0}}$$

The existence of this transition is more of a blemish than a real problem. It cannot be used to derive a communication because there exists no  $y$  such that  $x \neq y$  and  $x \dot{\rightarrow} y$ . It will be ignored by late-connectivity bisimulation (Definition 3.12) for the same reason, so it remains true that  $(\nu x)\bar{x}y \sim 0$ . Still, we maintain the view that a derivable transition should signify the readiness of the process to engage in some behaviour. This transition signifies nothing.

**3.4. Revisiting the counterexamples.** In the introduction, we mentioned that Bengtson et al. [BJPV11] have counterexamples to the effect that without symmetry and transitivity, scope extension is unsound. In this section we will revisit these counterexamples, with the aim of convincing the reader that they do not apply to our developments in the present paper.

We begin by quoting the counterexample used by Bengtson et al. to argue that scope extension requires channel symmetry [BJPV11, p. 14]:

Consider any psi-calculus where  $\Psi_1$  and  $\Psi_2$  are such that  $\Psi_1 \otimes \Psi_2 \vdash a \dot{\leftrightarrow} b$  and  $\Psi_1 \otimes \Psi_2 \vdash b \dot{\leftrightarrow} a$ . We shall argue that also  $\Psi_1 \otimes \Psi_2 \vdash b \dot{\leftrightarrow} a$  must hold, otherwise scope extension does not hold. Consider the agent

$$(\nu a, b)((\Psi_1) \mid (\Psi_2) \mid \bar{a}. \mathbf{0} \mid b. \mathbf{0})$$

which has an internal communication  $\tau$  using  $b$  as subjects in the premises of the COM rule. If  $b \# \Psi_1$  and  $a \# \Psi_2$ , by scope extension the agent should behave as

$$(\nu a)((\Psi_1) \mid \bar{a}. \mathbf{0}) \mid (\nu b)((\Psi_2) \mid b. \mathbf{0})$$

and therefore this agent must also have a  $\tau$  action. The left hand component cannot do an  $\bar{a}$  action, but in the environment of  $\Psi_2$  it can do a  $\bar{b}$  action.



Similarly, the right hand component cannot do a  $b$  action. The only possibility is for it to do an  $a$  action, as in

$$\Psi_1 \triangleright (\nu b)((\Psi_2) \mid b.\mathbf{0}) \xrightarrow{a} \dots$$

and this requires  $\Psi_1 \otimes \Psi_2 \vdash b \leftrightarrow a$ .

This counterexample is only valid because the authors were not careful about the orientation of channel equivalence judgements in the operational rules—understandably so, because they were designing a calculus with symmetric connectivity. To see this clearly, consider the preconditions to the rules for input and output in the original psi-calculi:

$$\text{IN-OLD} \frac{\Psi \vdash M \leftrightarrow K}{\Psi \triangleright \underline{M}(\lambda \tilde{y})N.P \xrightarrow{\underline{K}N[\tilde{y}:=\tilde{L}]} P[\tilde{y}:=\tilde{L}]}} \quad \text{OUT-OLD} \frac{\Psi \vdash M \leftrightarrow K}{\Psi \triangleright \overline{M}N.P \xrightarrow{\overline{K}N} P}$$

Note the inconsistency that in IN-OLD, the input channel occurs on the LHS of  $\leftrightarrow$ , whereas in OUT-OLD the output channel occurs on the LHS. This of course makes no difference when  $\leftrightarrow$  is symmetric, but for an asymmetric connectivity relation it is important to use it consistently, with the input channel always going on the same side of  $\leftrightarrow$ . Simply reorienting the channel equivalence judgement in IN-OLD suffices to make this counterexample inapplicable even to the original psi-calculi. This should not be taken to mean that the original psi-calculi do not require symmetry: we only mean to say that the reason for the symmetry requisite is not clear from this counterexample.

For transitivity, Bengtson et al. give the following counterexample [BJPV11, p. 14]:

Let  $\mathbf{1}$  entail  $a \leftrightarrow a$  for all names  $a$ , and let  $\Psi$  be an assertion with support  $\{a, b, c\}$  that additionally entails the two conditions  $a \leftrightarrow b$  and  $b \leftrightarrow c$ , but not  $a \leftrightarrow c$ , and thus does not satisfy transitivity of channel equivalence. If  $\Psi$  entails no other conditions then  $(\nu b)\Psi \simeq \mathbf{1}$ , and we expect  $(\nu b)(\Psi)$  to be interchangeable with  $(\mathbf{1})$  in all contexts. Consider the agent

$$\bar{a}.\mathbf{0} \mid c.\mathbf{0} \mid (\nu b)(\Psi)$$

By scope extension it should behave precisely as

$$(\nu b)(\bar{a}.\mathbf{0} \mid c.\mathbf{0} \mid (\Psi))$$

This agent has a  $\tau$ -transition since  $\Psi$  enables an interaction between the components  $\bar{a}.\mathbf{0}$  and  $c.\mathbf{0}$ . But the agent

$$\bar{a}.\mathbf{0} \mid c.\mathbf{0} \mid (\mathbf{1})$$

has no such transition. The conclusion is that  $(\nu b)\Psi$  must entail that the components can communicate, i.e. that  $a \leftrightarrow c$ , in other words  $\Psi \vdash a \leftrightarrow c$ .

The present author agrees with this reasoning, but reaches the exact opposite conclusion about which process is at fault: the anomaly here is not that  $\bar{a}.\mathbf{0} \mid c.\mathbf{0} \mid (\mathbf{1})$  cannot reduce, but that  $(\nu b)(\bar{a}.\mathbf{0} \mid c.\mathbf{0} \mid (\Psi))$  can. If  $a$  and  $c$  are not channel equivalent, there should not be a derivable communication between the channels  $a$  and  $c$ . The original psi-calculi nonetheless admit the derivation

$$\begin{array}{c}
\text{OUT} \frac{\Psi \vdash a \dot{\leftrightarrow} b}{\Psi \triangleright \bar{a}. \mathbf{0} \xrightarrow{\bar{b}} \dots} \quad \text{IN} \frac{\Psi \vdash c \dot{\leftrightarrow} c}{\Psi \triangleright c. \mathbf{0} \xrightarrow{c} \dots} \quad \Psi \vdash b \dot{\leftrightarrow} c \\
\text{COM-OLD} \frac{\Psi \triangleright \bar{a}. \mathbf{0} \xrightarrow{\bar{b}} \dots \quad \Psi \triangleright c. \mathbf{0} \xrightarrow{c} \dots \quad \Psi \vdash b \dot{\leftrightarrow} c}{\Psi \triangleright \bar{a}. \mathbf{0} \mid c. \mathbf{0} \xrightarrow{\tau} \dots} \\
\text{PAR} \frac{\Psi \triangleright \bar{a}. \mathbf{0} \mid c. \mathbf{0} \xrightarrow{\tau} \dots}{\mathbf{1} \triangleright \bar{a}. \mathbf{0} \mid c. \mathbf{0} \mid (\Psi) \xrightarrow{\tau} \dots} \\
\text{SCOPE} \frac{\mathbf{1} \triangleright \bar{a}. \mathbf{0} \mid c. \mathbf{0} \mid (\Psi) \xrightarrow{\tau} \dots}{\mathbf{1} \triangleright (\nu b)(\bar{a}. \mathbf{0} \mid c. \mathbf{0} \mid (\Psi)) \xrightarrow{\tau} \dots}
\end{array}$$

Notice how we use three different channel equivalence judgements to string together a derivation via  $b$ , which both  $a$  and  $c$  are connected to. This is not a problem if transitivity is intended, but leads to absurd derivations if the intention is to allow non-transitive connectivity relations.

With the provenance semantics, the counterexample does not apply since no communication between  $a$  and  $c$  is possible: the only possibility is to apply the COM rule with matching subjects and provenances. This requires  $\bar{a}. \mathbf{0}$  to have an output transition with subject  $c$  and  $c. \mathbf{0}$  to have an input transition with subject  $a$ , but such transitions cannot be derived because  $a \dot{\leftrightarrow} c$  (in our notation  $a \dot{\rightarrow} c$ ) does not hold.

Finally, we observe that a slight variant of the counterexample for transitivity illustrates the need for symmetry. This time, let  $\Psi$  be an assertion with support  $\{a, b, c, d\}$  that entails the three conditions  $a \dot{\leftrightarrow} b$  and  $d \dot{\leftrightarrow} b$  and  $c \dot{\leftrightarrow} d$  and none other. The agent

$$\bar{a}. \mathbf{0} \mid c. \mathbf{0} \mid \mathbf{1}$$

will have no transitions, but this agent will have a  $\tau$  transition:

$$(\nu b, d)(\bar{a}. \mathbf{0} \mid c. \mathbf{0} \mid (\Psi))$$

We conclude by the same reasoning as above that  $\Psi \vdash a \dot{\leftrightarrow} c$  must hold, or in other words, that channel equivalence must satisfy the law

$$a \dot{\leftrightarrow} b \wedge d \dot{\leftrightarrow} b \wedge c \dot{\leftrightarrow} d \Rightarrow a \dot{\leftrightarrow} c$$

This awkward-looking algebraic law is weaker than symmetry and transitivity, and together with reflexivity it implies both. It may well be that this weaker law is sufficient for the original psi-calculi to be compositional (assuming a channel equivalence reorientation in the IN-OLD rule). However, we find it difficult to imagine a useful connectivity relation that satisfies this law but is neither reflexive, transitive nor symmetric.

**3.5. Validation.** We have defined semantics and bisimulation, and showed that bisimilarity satisfies the expected laws. But how do we know that they are the right semantics, and the right bisimilarity? This section provides two answers to this question. First, we show that our developments constitute a conservative extension of the original psi-calculi. Second, we define a reduction semantics and barbed bisimulation that are in agreement with our (labelled) semantics and (labelled) bisimilarity.

Let  $\longrightarrow_o$  and  $\dot{\sim}_o$  denote semantics and bisimilarity as defined by Bengtson et al. [BJPV11], i.e., without provenances and with the COM-OLD rule discussed in Section 2. Along the same lines, let  $\dot{\approx}_o$  and  $\dot{\approx}_{\text{tau}_o}$  and  $\approx_o$  denote weak bisimulation, weak  $\tau$ -bisimilarity and weak congruence as defined by Johansson et al. [JBPV10]. Then conservativity can be stated as follows:

**Theorem 3.13** (Conservativity). *When  $\rightarrow$  is symmetric and transitive we have*

$$\dot{\sim}_o = \dot{\sim} \qquad \rightarrow_o = \rightarrow \qquad \dot{\approx}_o = \dot{\approx} \qquad \dot{\tau}_{\text{tau}_o} = \dot{\tau}_{\text{tau}} \qquad \approx_o = \approx$$

*Proof.* Formally proven in Isabelle. The bulk of the proof is to show that  $\rightarrow_o = \rightarrow$ .

The  $\Leftarrow$  direction is by induction on the derivation of the  $\rightarrow$  judgement, using symmetry to reorient the connectivity judgement in the IN case. In the COM case, associativity and Lemma 3.2 are used to reconstruct the missing channel equivalence judgement

The  $\Rightarrow$  direction is by induction on the  $\rightarrow_o$  judgement, and is more involved; in particular, the COM-OLD case requires relabelling the transitions obtained from the induction hypotheses with the provenance of the other.  $\square$

Our reduction semantics departs from standard designs [BB90, Mil90] by relying on reduction contexts [FH92] instead of structural rules, for two reasons. First, standard formulations tend to include rules like these:

$$\frac{P \rightarrow P'}{P \mid Q \rightarrow P' \mid Q} \qquad \frac{}{\alpha.P + Q \mid \bar{\alpha}.R + S \rightarrow P \mid R}$$

A parallel rule like the above would be unsound because  $Q$  might contain assertions that retract some conditions needed to derive  $P$ 's reduction. The reduction axiom assumes prefix-guarded choice. We want our semantics to apply to the full calculus, without limiting the syntax to prefix-guarded **case** statements.

But first, a few auxiliary definitions. The *reduction contexts* are the contexts in which communicating processes may occur:

**Definition 3.14** (Reduction contexts). The *reduction contexts*, ranged over by  $C$ , are generated by the grammar

$$\begin{aligned} C & ::= P_G && \text{(process)} \\ & \quad [] && \text{(hole)} \\ & \quad C \mid C && \text{(parallel)} \\ & \quad \mathbf{case} \ \tilde{\varphi} : \tilde{P}_G \ \square \ \varphi' : C \ \square \ \tilde{\varphi}'' : \tilde{Q}_G && \text{(case)} \end{aligned}$$

Let  $H(C)$  denote the number of holes in  $C$ .  $C[\tilde{P}_G]$  denotes the process that results from filling each hole of  $C$  with the corresponding element of  $\tilde{P}_G$ , where holes are numbered from left to right; if  $H(C) \neq |\tilde{P}_G|$ ,  $C[\tilde{P}_G]$  is undefined.

We do not need restriction contexts—instead, we rely on structural rules to pull all restrictions to the top level. To this end, we let *structural congruence*  $\equiv$  be the smallest equivalence relation on processes derivable using Theorems 3.3 and 3.4. The *conditions*  $\text{conds}(C)$  and *parallel processes*  $\text{ppr}(C)$  of a context  $C$  are, respectively, the conditions in  $C$  that guard the holes, and the processes of  $C$  that are parallel to the holes:

$$\begin{array}{c}
\text{STRUCT} \frac{P \equiv Q \quad Q \longrightarrow Q' \quad Q' \equiv P'}{P \longrightarrow P'} \qquad \text{SCOPE} \frac{P \longrightarrow Q}{(\nu a)P \longrightarrow (\nu a)Q} \\
\text{CTXT} \frac{\tilde{\Psi} \vdash M \dot{\rightarrow} N \quad K = L[\tilde{x} := \tilde{T}] \quad \forall \varphi \in \text{conds}(C). \tilde{\Psi} \vdash \varphi}{(\tilde{\Psi}) \mid C[\overline{M} K.P, \underline{N}(\lambda \tilde{x})L.Q] \longrightarrow (\tilde{\Psi}) \mid P \mid Q[\tilde{x} := \tilde{T}] \mid \text{ppr}(C)}
\end{array}$$

Table 2: Reduction semantics. Here  $\tilde{\Psi}$  abbreviates the composition  $\Psi_1 \otimes \Psi_2 \otimes \dots$ , and  $(\tilde{\Psi})$  abbreviates the parallel composition  $(\Psi_1) \mid (\Psi_2) \mid \dots$ —for empty sequences they are taken to be  $\mathbf{1}$  and  $\mathbf{0}$  respectively.

$$\begin{array}{l}
\text{ppr}(P_G) = P_G \\
\text{ppr}([\ ] ) = \mathbf{0} \\
\text{ppr}(C_1 \mid C_2) = \text{ppr}(C_1) \mid \text{ppr}(C_2) \\
\text{ppr}(\mathbf{case} \tilde{\varphi} : \tilde{P}_G \parallel \varphi' : C \parallel \tilde{\varphi}'' : \tilde{Q}_G) = \text{ppr}(C) \\
\text{conds}(P_G) = \emptyset \\
\text{conds}([\ ] ) = \emptyset \\
\text{conds}(C_1 \mid C_2) = \text{conds}(C_1) \cup \text{conds}(C_2) \\
\text{conds}(\mathbf{case} \tilde{\varphi} : \tilde{P}_G \parallel \varphi' : C \parallel \tilde{\varphi}'' : \tilde{Q}_G) = \{\varphi'\} \cup \text{conds}(C)
\end{array}$$

**Definition 3.15** (Reduction semantics). The reduction relation  $\longrightarrow \subseteq \mathbf{P} \times \mathbf{P}$  is defined inductively by the rules of Table 2.

In words, CTXT states that if an input and output prefix occur in a reduction context, we may derive a reduction if the following holds: the prefixes are connected in the current assertion environment, the message matches the input pattern, and all conditions guarding the prefixes are entailed by the environment. The  $\text{ppr}(C)$  in the reduct makes sure any processes in parallel to the holes are preserved.

Note that even though an unrestricted parallel rule would be unsound in a psi-calculus with non-monotonic composition, the following is valid as a derived rule:

$$\frac{P \longrightarrow P'}{P \mid Q_G \longrightarrow P' \mid Q_G}$$

**Theorem 3.16.**  $P \longrightarrow P'$  iff there is  $P''$  such that  $\mathbf{1} \triangleright P \xrightarrow{\tau} P''$  and  $P'' \equiv P'$

*Proof.* A full proof is available in the technical report [ÅP19b]. The  $\Leftarrow$  direction is by induction on the derivation of  $P \longrightarrow P'$ . The  $\Rightarrow$  direction is via reduction to normal form, showing that for every process  $P$  there are  $\tilde{x}, \tilde{\Psi}, P_G$  such that

$$P \equiv (\nu \tilde{x})(\tilde{\Psi}) \mid P_G \quad \square$$

For barbed bisimulation, we need to define what the observables are, and what contexts an observer may use. We follow previous work by Johansson et al. [JBPV10] on weak barbed bisimilarity for the original psi-calculi on both counts. First, we take the barbs to be the

output labels a process can exhibit: we define  $P \downarrow_{\overline{M}(\nu\tilde{a})N}$  ( $P$  exposes  $\overline{M}(\nu\tilde{a})N$ ) to mean  $\exists P'. 1 \triangleright P \xrightarrow{\overline{M}(\nu\tilde{a})N} P'$ . We write  $P \downarrow_{\overline{M}}$  for  $\exists \tilde{a}, N. P \downarrow_{\overline{M}(\nu\tilde{a})N}$ , and  $P \downarrow_{\alpha}$  for  $P \xrightarrow{\tau}^* \downarrow_{\alpha}$ . Second, we let observers use *static* contexts, i.e. ones built from parallel and restriction.

**Definition 3.17** (Barbed bisimilarity). *Barbed bisimilarity*, written  $\overset{\sim}{\text{barb}}$ , is the largest equivalence on processes such that  $P \overset{\sim}{\text{barb}} Q$  implies

- (1) If  $P \downarrow_{\overline{M}(\nu\tilde{a})N}$  and  $\tilde{a}\#Q$  then  $Q \downarrow_{\overline{M}(\nu\tilde{a})N}$  (barb similarity)
- (2) If  $P \longrightarrow P'$  then there exists  $Q'$  such that  $Q \longrightarrow Q'$  and  $P' \overset{\sim}{\text{barb}} Q'$  (reduction simulation)
- (3)  $(\nu\tilde{a})(P \mid R) \overset{\sim}{\text{barb}} (\nu\tilde{a})(Q \mid R)$  (closure under static contexts)

Our proof that barbed and labelled bisimilarity coincides only considers psi-calculi with a certain minimum of sanity and expressiveness. This rules out some degenerate cases: psi-calculi where there are messages that can be sent but not received,<sup>3</sup> and psi-calculi where no transitions whatsoever are possible.

**Definition 3.18.** A psi-calculus is *observational* if:

- (1) For all  $P$  there are  $M_P, K_P$  such that  $\mathcal{F}(P) \vdash M_P \dot{\rightarrow} K_P$  and not  $P \downarrow_{\overline{K_P}}$ .
- (2) If  $N = (\tilde{x} \tilde{y}) \cdot M$  and  $\tilde{y}\#M$  and  $\tilde{x}, \tilde{y}$  are distinct then  $M[\tilde{x} := \tilde{y}] = N$ .

The first clause means that no process can exhaust the set of barbs. Hence observing contexts can signal success or failure without interference from the process under observation. For example, in the pi-calculus  $M_P, K_P$  can be any name  $x$  such that  $x\#P$ . The second clause states that for swapping of distinct names, substitution and permutation have the same behaviour. Any standard definition of simultaneous substitution should satisfy this requirement. These assumptions are present, explicitly or implicitly, in the work of Johansson et al. [JBPV10]. Ours are given a slightly weaker formulation.

We can now state the main result of this section:

**Theorem 3.19.** *In all observational psi-calculi,  $P \overset{\sim}{\text{barb}} Q$  iff  $P \overset{\sim}{\text{barb}}_1 Q$ .*

*Proof.* A full proof is available in the technical report [ÅP19b]. Soundness is by coinduction on the definition of barbed bisimilarity, using  $\overset{\sim}{\text{barb}}_1$  as candidate relation, and using Theorems 3.16 and 3.3, and the fact that  $\equiv$  is a strong bisimulation. Completeness is by showing that  $\{(\Psi, P, Q) : P \mid (\Psi) \overset{\sim}{\text{barb}} Q \mid (\Psi)\}$  is a bisimulation relation.  $\square$

#### 4. EXPRESSIVENESS

In this section, we study two examples of the expressiveness gained by dropping symmetry and transitivity.

<sup>3</sup>More precisely, we refer to messages that can occur as objects of output transitions, but not as the objects of input transitions. This can happen in psi-calculi where the substitution function on terms is not surjective, because the rule IN requires the message to be the result of a substitution.

**4.1. Pi-calculus with preorders.** Recall that pi-F [WG05] extends the pi-calculus with name equalities ( $x = y$ ) as first-class processes. Communication in pi-F gives rise to equalities rather than substitutions, so e.g.  $xy.P \mid \bar{x}z.Q$  reduces to  $y = z \mid P \mid Q$ : the input and output objects are fused. Hirschhoff et al. [HMS13] observe that fusion and subtyping are fundamentally incompatible, and propose a generalisation of pi-F called the *pi-calculus with preorders* or  $\pi P$  to resolve the issue.

We are interested in  $\pi P$  because its channel connectivity is not transitive. The equalities of pi-F are replaced with *arcs*  $a/b$  (“ $a$  is above  $b$ ”) which act as one-way fusions: anything that can be done with  $b$  can be done with  $a$ , but not the other way around. The effect of a communication is to create an arc with the output subject above the input subject, so  $x(y).P \mid \bar{x}(z).Q$  reduces to  $(\nu yz)(z/y \mid P \mid Q)$ . We write  $A \vdash x \prec y$  to mean that  $x$  and  $y$  are related by the reflexive and transitive closure of the set of arcs  $A$ .  $A$  is usually the set of top-level arcs in the process under consideration, and will often be left implicit. Two names  $x, y$  are considered *joinable* for the purposes of synchronisation if some name  $z$  is above both of them: formally, we write  $x \Upsilon y$  for  $\exists z. x \prec z \wedge y \prec z$ .

Hirschhoff et al. conclude by saying that “[it] could also be interesting to study the representation of  $\pi P$  into Psi-calculi. This may not be immediate because the latter make use of on an equivalence relation on channels, while the former uses a preorder” [HMS13, p. 387]. Having lifted the constraint that channels form an equivalence relation, we happily accept the challenge. We write  $\Psi P$  for the psi-calculus we use to embed  $\pi P$ . We follow the presentation of  $\pi P$  from [HMX15a, HMX15b], where the behavioural theory is most developed.

**Definition 4.1.** The psi-calculus  $\Psi P$  is defined with the following parameters:

$$\begin{aligned}
\mathbf{T} &\triangleq \mathcal{N} \\
\mathbf{C} &\triangleq \{x \prec y : x, y \in \mathcal{N}\} \cup \{x \Upsilon y : x, y \in \mathcal{N}\} \\
\mathbf{A} &\triangleq \mathcal{P}_{\text{fin}}(\{x \prec y : x, y \in \mathcal{N}\}) \\
\mathbf{1} &\triangleq \{\} \\
\otimes &\triangleq \cup \\
\dot{\rightarrow} &\triangleq \Upsilon \\
\vdash &\triangleq \text{the relation denoted } \vdash \text{ in [HMX15b]}.
\end{aligned}$$

The prefix operators of  $\pi P$  are different from those of psi-calculi: objects are always bound, communication gives rise to an arc rather than a substitution, and a conditional silent prefix  $[\varphi]\tau.P$  is included. The full syntax, ignoring protected prefixes, is as follows:<sup>4</sup>

**Definition 4.2** (Syntax of  $\pi P$ ).

$$\begin{aligned}
P &:= a/b && \text{(arc)} \\
&\quad \Sigma_{i \in I} \pi_i.P_i && \text{(prefix-guarded choice)} \\
&\quad P \mid Q && \text{(parallel)} \\
&\quad (\nu x)P && \text{(restriction)} \\
\pi &:= a(x) && \text{(input)} \\
&\quad \bar{a}(y) && \text{(output)} \\
&\quad [\varphi]\tau && \text{(conditional silent prefix)} \\
\varphi &:= x \prec y \\
&\quad x \Upsilon y
\end{aligned}$$

<sup>4</sup>We ignore protected prefixes because they are redundant, cf. Remark 1 of [HMX15a].

Here  $I$  is a finite index set.

We can now define our encoding of  $\pi P$  prefixes:

**Definition 4.3** (Encoding of prefixes). The encoding  $\llbracket \_ \rrbracket$  from  $\pi P$  to  $\Psi P$  is homomorphic on all operators except prefixes and arcs, where it is defined by

$$\begin{aligned} \llbracket a/b \rrbracket &= (b \prec a) \\ \llbracket \bar{a}(y).P \rrbracket &= (\nu xy)(\bar{a}x.(\langle x \prec y \rangle \mid \llbracket P \rrbracket)) && \text{where } x\#y, P \\ \llbracket a(y).P \rrbracket &= (\nu y)(\underline{a}(\lambda x)x.(\langle y \prec x \rangle \mid \llbracket P \rrbracket)) && \text{where } x\#y, P \\ \llbracket [\varphi]\tau.P \rrbracket &= \mathbf{case} \varphi : (\nu x)(\underline{x}(\lambda x)x.0 \mid \bar{x}x.\llbracket P \rrbracket) && \text{where } x\#P \end{aligned}$$

For choice, we let  $\llbracket \sum_{i \in I} P_i \rrbracket = \mathbf{case} \tilde{\varphi} : \llbracket P \rrbracket$ , where each  $\varphi_i$  is a condition that is always entailed.<sup>5</sup>

This embedding of  $\pi P$  in psi-calculi comes with a notion of bisimilarity per Definition 3.1. We show that it coincides with the labelled bisimilarity for  $\pi P$  (written  $\sim$ ) introduced in [HMX15a, HMX15b].

**Theorem 4.4.**  $P \sim Q$  iff  $\llbracket P \rrbracket \dot{\sim} \llbracket Q \rrbracket$

*Proof.* A full proof is available in the technical report [ÅP19b]. We prove strong operational correspondence by a tedious induction, then prove (respectively) that  $\{(P, Q).\llbracket P \rrbracket \dot{\sim}_1 \llbracket Q \rrbracket\}$  is a bisimulation, and that  $\{(\Psi, \llbracket P \rrbracket, \llbracket Q \rrbracket).P \mid \Psi \sim Q \mid \Psi\}$  is a bisimulation up to  $\dot{\sim}$ .  $\square$

Thus our encoding validates the behavioural theory of  $\pi P$  by connecting it to our fully mechanised proofs, while also showing that a substantially different design of the LTS yields the same bisimilarity. We will briefly compare these designs. While we do rewriting of subjects in the prefix rules, Hirschhoff et al. instead use relabelling rules like this one (mildly edited to match our notation):

$$\frac{P \xrightarrow{a(x)} P' \quad \mathcal{F}(P) \vdash a \prec b}{P \xrightarrow{b(x)} P'}$$

An advantage of this rule is that it allows input and output labels to be as simple as pi-calculus labels. A comparative disadvantage is that it is not syntax-directed, and that the LTS has more rules in total. Note that this rule would not be a viable alternative to provenances in psi-calculi: since it can be applied more than once in a derivation, its inclusion assumes that the channels form a preorder wrt. connectivity.

$\pi P$  also has labels  $[\varphi]\tau$ , meaning that a silent transition is allowed in environments where  $\varphi$  is true. A rule for rewriting  $\varphi$  to a weaker condition, similar to the above rule for subject rewriting, is included. Psi-calculi does not need this because the PAR rules take the assertion environment into account.  $\pi P$  transitions of kind  $P \xrightarrow{[\varphi]\tau} P'$  correspond to  $\Psi P$  transitions of kind  $\{\varphi\} \triangleright P \xrightarrow{\tau} P'$ .

Interestingly, the analogous full abstraction result fails to hold for the embedding of pi-f in psi-calculi by Bengtson et al. [BJPV11], because outputs that emit distinct but fused names are distinguished by psi-calculus bisimilarity. This issue does not arise here because  $\pi P$  objects are always bound; however, we believe the encoding of Bengtson et al. can be

<sup>5</sup>Such a condition can either be added to the target language, or we can use e.g.  $a \prec a$  at the cost of some technical inconvenience. See the technical report for details.

made fully abstract by encoding free output with bound output, exploiting the pi-F law  $a y.Q \sim a(x)(Q \mid x = y)$ .

**4.2. Mixed choice.** This section will argue that because we allow non-transitive channel connectivity, the **case** operator of psi-calculi becomes superfluous. The formal results here will focus on encoding the special case of mixed choice. We will then briefly discuss how to generalise these results to the full **case** operator.

Choice, written  $P + Q$ , is a process that behaves as either  $P$  or  $Q$ . In psi-calculi we consider  $P + Q$  to abbreviate **case**  $\top : P \square \top : Q$  for some condition  $\top$  that is always entailed. *Mixed choice* means that in  $P + Q$ ,  $P$  and  $Q$  must be prefix-guarded. In particular, mixed choice allows choice between an input and an output. There is a straightforward generalisation to  $n$ -ary sums that, in order to simplify the presentation, we will not consider here.

Fix a psi-calculus  $\mathcal{P} = (\mathbf{T}, \mathbf{A}, \mathbf{C}, \vdash, \otimes, \mathbf{1}, \dot{\rightarrow})$  with mixed choice. This will be our source language. For technical convenience we assume that  $\mathcal{P}$  satisfies the equation  $\mathbf{1}\sigma = \mathbf{1}$  for all substitutions  $\sigma$ ; see the associated technical report [ÅP19b] for a discussion on how this assumption can be lifted. We will construct a target psi-calculus and an encoding such that the target terms make no use of the **case** operator. The target language  $\mathcal{E}(\mathcal{P})$  adds to  $\mathbf{T}$  the ability to tag a term  $M$  with a name  $x$ ; we write  $M_x$  for the tagged term. We write  $\alpha_x$  for tagging the subject of the prefix  $\alpha$  with  $x$ . Tags are used to uniquely identify which choice statement a prefix is a summand of. As the assertions of  $\mathcal{E}(\mathcal{P})$  we use  $\mathbf{A} \times \mathcal{P}_{\text{fin}}(\mathcal{N})$ , where  $\mathcal{P}_{\text{fin}}(\mathcal{N})$  are the *disabled tags*.

**Definition 4.5** (Target language). Let  $\mathcal{P} = (\mathbf{T}, \mathbf{A}, \mathbf{C}, \vdash, \otimes, \mathbf{1}, \dot{\rightarrow})$  be a psi-calculus. Then  $\mathcal{E}(\mathcal{P}) = (\mathbf{T}_{\mathcal{E}}, \mathbf{A}_{\mathcal{E}}, \mathbf{C}_{\mathcal{E}}, \vdash_{\mathcal{E}}, \otimes_{\mathcal{E}}, \mathbf{1}_{\mathcal{E}}, \dot{\rightarrow}_{\mathcal{E}})$  is a psi-calculus whose components are as follows:

$$\begin{aligned} \mathbf{T}_{\mathcal{E}} &= \mathbf{T} \uplus \{M_x : x \in \mathcal{N}, M \in \mathbf{T}_{\mathcal{E}}\} \\ \mathbf{A}_{\mathcal{E}} &= \mathbf{A} \times \mathcal{P}_{\text{fin}}(\mathcal{N}) \\ \mathbf{C}_{\mathcal{E}} &= \mathbf{C} \uplus \{M \dot{\rightarrow} N : M, N \in \mathbf{T}\} \uplus \mathcal{N} \\ (\Psi, \mathbf{N}) \otimes_{\mathcal{E}} (\Psi', \mathbf{N}') &= (\Psi \otimes \Psi', \mathbf{N} \cup \mathbf{N}') \\ \mathbf{1}_{\mathcal{E}} &= (\mathbf{1}, \emptyset) \end{aligned}$$

$$\begin{aligned} (\Psi, \mathbf{N}) \vdash_{\mathcal{E}} \varphi & \quad \text{if } \varphi \in \mathbf{C} \text{ and } \Psi \vdash \varphi \\ (\Psi, \mathbf{N}) \vdash_{\mathcal{E}} x & \quad \text{if } x \in \mathcal{N} \text{ and } x \in \mathbf{N} \\ (\Psi, \mathbf{N}) \vdash_{\mathcal{E}} M_x \dot{\rightarrow} N_y & \quad \text{if } \Psi \vdash M \dot{\rightarrow} N \text{ and } x \neq y \text{ and } x, y \notin \mathbf{N} \\ (\Psi, \mathbf{N}) \vdash_{\mathcal{E}} M_x \dot{\rightarrow} N & \quad \text{if } \Psi \vdash M \dot{\rightarrow} N \text{ and } x \notin \mathbf{N} \\ (\Psi, \mathbf{N}) \vdash_{\mathcal{E}} M \dot{\rightarrow} N_x & \quad \text{if } \Psi \vdash M \dot{\rightarrow} N \text{ and } x \notin \mathbf{N} \end{aligned}$$

We assume that the target language is constrained by a sorting system as in [BGP<sup>+</sup>13] that ensures only terms  $M \in \mathbf{T}$  can ever occur as objects of communication, and in particular, that for all substitutions  $[\tilde{x} := \tilde{T}]$ ,  $\tilde{T} \subseteq \mathbf{T}$ . The purpose of this simplification is to avoid having to consider input transitions such as

$$\Psi \triangleright \underline{M}(\lambda \tilde{x})N.P \xrightarrow{K K_x} P'$$

that may result in substitutions where tagged terms must be substituted into source-language terms or vice versa. It is possible to lift this assumption at the cost of significant technical inconvenience [ÅP19b].



The encoding  $\llbracket - \rrbracket$  from  $\mathcal{P}$  to  $\mathcal{E}(\mathcal{P})$  is homomorphic on all operators except assertion and choice, where it is defined as follows:

$$\llbracket (\Psi) \rrbracket = \llbracket (\Psi, \emptyset) \rrbracket \quad \llbracket \alpha.P + \beta.Q \rrbracket = (\nu x)(\alpha_x.(\llbracket P \rrbracket \mid \llbracket (1, \{x\}) \rrbracket) \mid \beta_x.(\llbracket Q \rrbracket \mid \llbracket (1, \{x\}) \rrbracket))$$

where  $x \# \alpha, \beta, P, Q$ . If we disregard the tag  $x$ , we see that the encoding simply offers up both summands in parallel. This clearly allows all behaviours of  $\alpha.P + \beta.Q$ , but there are two additional behaviours we must prevent: (1) communication between the summands, and (2) lingering summands firing after the other branch has already been taken. The tagging mechanism prevents both, as a consequence of how we define channel equivalence on tagged terms in  $\mathcal{E}(\mathcal{P})$ : tagged channels are connected if the underlying channel is connected. To prevent (1), Definition 4.5 requires the tags of connected channels to be different, and to prevent (2) the definition requires that the tags are not disabled. Note that this channel connectivity is not transitive, not reflexive, and not monotonic wrt. assertion composition—not even if the source language connectivity is.

**Example 4.6.** We illustrate the operational behaviour of the encoding for the process  $R = \alpha.P + \beta.Q$ . Its encoding is

$$\llbracket R \rrbracket = (\nu x)(\alpha_x.(\llbracket P \rrbracket \mid \llbracket (1, \{x\}) \rrbracket) \mid \beta_x.(\llbracket Q \rrbracket \mid \llbracket (1, \{x\}) \rrbracket))$$

where  $x$  is a fresh name. Suppose  $\alpha$  is an output prefix with subject  $M$ , and that channel connectivity is reflexive. Then we can derive the transition  $R \xrightarrow{\alpha} P$ . The corresponding derivation from  $\llbracket R \rrbracket$  uses the connectivity judgement  $M_x \dot{\rightarrow} M$  in the OUT rule to derive the following transition:

$$\llbracket R \rrbracket \xrightarrow{\alpha} (\nu x)(\llbracket P \rrbracket \mid S)$$

where

$$S = \llbracket (1, \{x\}) \rrbracket \mid b_x.(\llbracket Q \rrbracket \mid \llbracket (1, \{x\}) \rrbracket)$$

Since  $x$  is fresh in  $\llbracket P \rrbracket$ , by scope extension we have  $(\nu x)(\llbracket P \rrbracket \mid S) \dot{\sim} \llbracket P \rrbracket \mid (\nu x)S$ . Moreover, we have  $(\nu x)S \dot{\sim} 0$ . To see why, note first that  $(\nu x)S$  has no outgoing transitions: its only prefix has the tag  $x$ , which is disabled by its top-level assertion  $\llbracket (1, \{x\}) \rrbracket$ . Second, note that since this disabled tag is a local name, its disabling has no effect on the environment.

**Theorem 4.7** (Correctness of choice encoding).

- (1) If  $\Psi \triangleright P \xrightarrow{\alpha} P'$  then there is  $P''$  such that  $(\Psi, \emptyset) \triangleright \llbracket P \rrbracket \xrightarrow{\alpha} P''$  and  $P'' \dot{\sim}_{(\Psi, \emptyset)} \llbracket P' \rrbracket$ .
- (2) If  $(\Psi, \emptyset) \triangleright \llbracket P \rrbracket \xrightarrow{\alpha} P'$  then there is  $P''$  such that  $\Psi \triangleright P \xrightarrow{\alpha} P''$  and  $P' \dot{\sim}_{(\Psi, \emptyset)} \llbracket P'' \rrbracket$ .
- (3)  $P \dot{\sim}_1 Q$  iff  $\llbracket P \rrbracket \dot{\sim}_{(1, \emptyset)} \llbracket Q \rrbracket$ .

*Proof.* A full proof is available in the technical report [ÅP19b]. Forward simulation is by induction on the derivation of  $\Psi \triangleright P \xrightarrow{\alpha} P'$ , backward simulation by structural induction on  $P$  followed by inversion on the derivation of the transition from  $\llbracket P \rrbracket$ . Full abstraction is by showing that

$$\{((\Psi, \mathbf{N}), \llbracket P \rrbracket, \llbracket Q \rrbracket) : P \dot{\sim}_{(\Psi)} Q\}$$

and

$$\{(\Psi, P, Q) : \llbracket P \rrbracket \dot{\sim}_{(\Psi, \emptyset)} \llbracket Q \rrbracket\}$$

are bisimulation relations. □

Here  $\alpha_{\perp}$  denotes the label  $\alpha$  with all tags removed. It is immediate from Theorem 4.7 and the definition of  $\llbracket \_ \rrbracket$  that our encoding also satisfies the other standard quality criteria [Gor10]: it is compositional, it is name invariant, and it preserves and reflects barbs and divergence.

In the original psi-calculi, our target language is invalid because of its non-transitive connectivity. However, if we restrict attention to *separate* choice (where either both summands are inputs or both summands are outputs), a slight modification of the scheme above yields a correct encoding in the context of the original psi-calculi. With separate choice we can drop the side-condition that tags of connected processes are distinct from Definition 4.5—the only use of this side-condition is to prevent communication between summands, and separate choice already prevents this by construction. With this modified definition of  $\rightarrow_{\mathcal{E}}$ , we have that if  $\rightarrow$  is symmetric and transitive, then so is  $\rightarrow_{\mathcal{E}}$ . Or in other words, if the source language is expressible in the original psi-calculi then so is the target language.

These results generalise in a straightforward way to mixed **case** statements

$$\mathbf{case} \varphi_1 : \alpha.P \parallel \varphi_2 : \beta.Q$$

by additionally tagging terms with a condition, i.e.  $M_{x,\varphi_1}$ , that must be entailed in order to derive connectivity judgements involving the term. The generalisation to free choice, i.e.  $P + Q$  where  $P, Q$  can be anything, is more involved and sacrifices some compositionality. The idea is to use sequences of tags, representing which branches of which (possibly nested) case statements a prefix can be found in, and disallowing communication between prefixes in distinct branches of the same **case** operator.

## 5. CONCLUSION AND RELATED WORK

We have seen how psi-calculi can be conservatively extended to allow asymmetric and non-transitive communication topologies, sacrificing none of the bisimulation meta-theory. This confers enough expressiveness to capture a pi-calculus with preorders, and makes mixed choice a derived operator.

The work of Hirschhoff et al. [HMS13] is closely related in that it uses non-transitive connectivity; see Section 4.1 for an extensive discussion.

Broadcast psi-calculi [BHJ<sup>+</sup>15] extend psi-calculi with broadcast communication in addition to point-to-point communication. There, point-to-point channels must still be symmetric and transitive, but for broadcast channels this condition is lifted, at the cost of introducing other side-conditions on how names are used: broadcast prefixes must be connected via intermediate *broadcast channels* which have no greater support than either of the prefixes it connects, precluding language features such as name fusion. We believe provenances could be used to define a version of broadcast psi-calculi that does not need this side-condition.

Kouzapas et al. [KGG14] define a similar reduction context semantics for (broadcast) psi-calculi. Their reduction contexts requires three kinds of numbered holes with complicated side-conditions on how the holes may be filled; we have attempted to simplify the presentation by having only one kind of hole. While (weak) barbed congruence for psi-calculi has been studied before [JBPV10] (see Section 3.5), barbed congruence was defined in terms of the labelled semantics rather than a reduction semantics, thus weakening its claim to independent confirmation slightly.

There is a rich literature on choice encodings for the pi-calculus [Gor10, NP00, Pal97, PN12, PNG13], with many separation and encodability results under different quality

criteria for different flavours of choice. Encodings typically require complicated protocols and tradeoffs between quality criteria. Thanks to the greater expressive power of psi-calculi, our encoding is simpler and satisfies stronger quality criteria than any choice encoding for the pi-calculus. Closest to ours is the choice encoding of CCS into the DiX calculus by Busi and Gorrieri [BG94]. DiX introduces a primitive for annotating processes with *conflict sets*, that are intended as a generalisation of choice. Processes with overlapping conflict sets cannot interact, and when a process acts, every process with an overlapping conflict set is killed. These conflict sets perform the same role in the encoding as our tags do. We believe the tagging scheme used in our choice encoding also captures DiX-style conflict sets.

#### ACKNOWLEDGEMENTS

These ideas have benefited from discussions with many people at Uppsala University, ITU Copenhagen, the University of Oslo and Data61/CSIRO, including Jesper Bengtson, Christian Johansen, Magnus Johansson and Joachim Parrow. I would also like to thank Jean-Marie Madiot and the anonymous reviewers for valuable comments on earlier versions of the paper.

#### REFERENCES

- [ÅP19a] Johannes Åman Pohjola. Psi-calculi revisited: Connectivity and compositionality. In *Formal Techniques for Distributed Objects, Components, and Systems - 39th IFIP WG 6.1 International Conference, FORTE 2019, Held as Part of the 14th International Federated Conference on Distributed Computing Techniques, DisCoTec 2019, Kongens Lyngby, Denmark, June 17-21, 2019, Proceedings*, pages 3–20, 2019.
- [ÅP19b] Johannes Åman Pohjola. Psi-calculi revisited: Connectivity and compositionality. Technical Report EP192416, CSIRO, Canberra, Australia, 2019. <http://dx.doi.org/10.21203/rs.3.rs-1001001/v1>.
- [BB90] Gerard Berry and Gerard Boudol. The chemical abstract machine. In *Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, POPL '90*, pages 81–94, New York, NY, USA, 1990. ACM.
- [BC88] Gérard Boudol and Ilaria Castellani. A non-interleaving semantics for CCS based on proved transitions. *Fundamenta Informaticae*, 11:433–452, 1988.
- [BG94] Nadia Busi and Roberto Gorrieri. Distributed conflicts in communicating systems. In Paolo Ciancarini, Oscar Nierstrasz, and Akinori Yonezawa, editors, *Object-Based Models and Languages for Concurrent Systems, ECOOP'94 Workshop on Models and Languages for Coordination of Parallelism and Distribution, Bologna, Italy, July 5, 1994, Selected Papers*, volume 924 of *Lecture Notes in Computer Science*, pages 49–65. Springer, 1994.
- [BGP<sup>+</sup>13] Johannes Borgström, Ramunas Gutkovas, Joachim Parrow, Björn Victor, and Johannes Åman Pohjola. A sorted semantic framework for applied process calculi (extended abstract). In *TGC*, pages 103–118, 2013.
- [BHJ<sup>+</sup>15] Johannes Borgström, Shuqin Huang, Magnus Johansson, Palle Raabjerg, Björn Victor, Johannes Åman Pohjola, and Joachim Parrow. Broadcast psi-calculi with an application to wireless protocols. *Software and System Modeling*, 14(1):201–216, 2015.
- [BJPV11] Jesper Bengtson, Magnus Johansson, Joachim Parrow, and Björn Victor. Psi-calculi: A framework for mobile processes with nominal data and logic. *Logical Methods in Computer Science*, 7(1), 2011.
- [BPW16] Jesper Bengtson, Joachim Parrow, and Tjark Weber. Psi-calculi in Isabelle. *J. Autom. Reasoning*, 56(1):1–47, 2016.
- [DP92] Pierpaolo Degano and Corrado Priami. Proved trees. In *Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13-17, 1992, Proceedings*, pages 629–640, 1992.

- [FH92] Matthias Felleisen and Robert Hieb. The revised report on the syntactic theories of sequential control and state. *Theor. Comput. Sci.*, 103(2):235–271, 1992.
- [Gor10] Daniele Gorla. Towards a unified approach to encodability and separation results for process calculi. *Inf. Comput.*, 208(9):1031–1053, 2010.
- [GP02] Murdoch J. Gabbay and Andrew M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2002.
- [GW00] Philippa Gardner and Lucian Wischik. Explicit fusions. In Mogens Nielsen and Branislav Rovan, editors, *Proceedings of MFCS 2000*, volume 1893, pages 373–382, 2000.
- [HMS13] Daniel Hirschhoff, Jean-Marie Madiot, and Davide Sangiorgi. Name-passing calculi: From fusions to preorders and types. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 378–387. IEEE Computer Society, 2013.
- [HMX15a] Daniel Hirschhoff, Jean-Marie Madiot, and Xian Xu. A behavioural theory for a  $\pi$ -calculus with preorders. *J. Log. Algebr. Meth. Program.*, 84(6):806–825, 2015.
- [HMX15b] Daniel Hirschhoff, Jean-Marie Madiot, and Xian Xu. A behavioural theory for a  $\pi$ -calculus with preorders. In Mehdi Dastani and Marjan Sirjani, editors, *Fundamentals of Software Engineering - 6th International Conference, FSEN 2015 Tehran, Iran, April 22-24, 2015, Revised Selected Papers*, volume 9392 of *Lecture Notes in Computer Science*, pages 143–158. Springer, 2015.
- [JBPV10] Magnus Johansson, Jesper Bengtson, Joachim Parrow, and Björn Victor. Weak equivalences in psi-calculi. In *LICS*, pages 322–331. IEEE Computer Society, 2010.
- [Joh10] Magnus Johansson. *Psi-calculi: a framework for mobile process calculi: Cook your own correct process calculus - just add data and logic*. PhD thesis, Uppsala University, Division of Computer Systems, 2010.
- [KGG14] Dimitrios Kouzapas, Ramunas Gutkovas, and Simon J. Gay. Session types for broadcasting. In Alastair F. Donaldson and Vasco T. Vasconcelos, editors, *Proceedings 7th Workshop on Programming Language Approaches to Concurrency and Communication-centric Software, PLACES 2014, Grenoble, France, 12 April 2014.*, volume 155 of *EPTCS*, pages 25–31, 2014.
- [Mil90] Robin Milner. Functions as processes. In *Proceedings of the seventeenth international colloquium on Automata, languages and programming*, pages 167–180, New York, NY, USA, 1990. Springer-Verlag New York, Inc.
- [MPW92] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, part I/II. *Inf. Comput.*, 100(1):1–77, 1992.
- [NH06] Sebastian Nanz and Chris Hankin. A framework for security analysis of mobile wireless networks. *Theoretical Computer Science*, 367(1-2):203–227, 2006.
- [NP00] Uwe Nestmann and Benjamin C. Pierce. Decoding choice encodings. *Inf. Comput.*, 163(1):1–59, 2000.
- [Pal97] Catuscia Palamidessi. Comparing the expressive power of the synchronous and the asynchronous pi-calculus. In Peter Lee, Fritz Henglein, and Neil D. Jones, editors, *Conference Record of POPL'97: The 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Papers Presented at the Symposium, Paris, France, 15-17 January 1997*, pages 256–265. ACM Press, 1997.
- [PN12] Kirstin Peters and Uwe Nestmann. Is it a "good" encoding of mixed choice? In Lars Birkedal, editor, *Foundations of Software Science and Computational Structures - 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, volume 7213 of *Lecture Notes in Computer Science*, pages 210–224. Springer, 2012.
- [PNG13] Kirstin Peters, Uwe Nestmann, and Ursula Goltz. On distributability in process calculi. In Matthias Felleisen and Philippa Gardner, editors, *Programming Languages and Systems - 22nd European Symposium on Programming, ESOP 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings*, volume 7792 of *Lecture Notes in Computer Science*, pages 310–329. Springer, 2013.
- [Urb08] Christian Urban. Nominal techniques in Isabelle/HOL. *Journal of Automated Reasoning*, 40(4):327–356, May 2008.
- [WG05] Lucian Wischik and Philippa Gardner. Explicit fusions. *Theoretical Computer Science*, 304(3):606–630, 2005.