Psi-Calculi Revisited: Connectivity and Compositionality

Johannes Åman Pohjola[0000−0002−6406−7875]

Data61/CSIRO, Sydney, Australia
University of New South Wales, Sydney, Australia
`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 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.

Keywords: Process algebra · Psi-calculi · Nominal logic · Interactive theorem proving · Bisimulation

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 [18], channel connectivity is syntactic identity: in the process

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

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 connnectivity. The explicit fusion calculus pi-F [9] 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 $(a = b)$

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

then communication is possible. Other examples may be found in e.g. calculi for wireless communication [19], where channel connectivity can be used to directly model the network's topology.

Psi-calculi [2] 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 [3] 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. [2, 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 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[1] in Nominal Isabelle [24]. (Section 3.1)
– We prove, again using Nominal Isabelle, that this paper's developments constitute a conservative extension of the original psi-calculi. (Section 3.2)
– We further validate our semantics by defining a reduction semantics and strong barbed congruence, and showing that they agree with their labelled counterparts. (Section 3.2)

----

[1] Isabelle proofs are available at `https://github.com/IlmariReissumies/newpsi`

- We capture a pi-calculus with preorders by Hirschkoff et al. [11], 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 Hirschkoff 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)

For lack of space we elide proofs; please see the associated technical report [1]. The final publication is available at Springer via `https://doi.org/10.1007/978-3-030-21759-4_1`.

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 [2]. We will highlight the places where the two differ.

We assume a countable set of *names* $\mathcal{N}$ ranged over by $a, b, c, \ldots, x, y, z$. A *nominal set* [8] 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 occurences of the name x swapped for y and vice versa, is also an element of $\mathbf{X}$. $\mathsf{n}(X)$ (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 $a \notin \mathsf{n}(X)$. A nominal set $\mathbf{X}$ has *finite support* if for every $X \in \mathbf{X}$, $\mathsf{n}(X)$ is finite. A function symbol f is *equivariant* if $p \cdot f(x) = f(p \cdot x)$; 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. $(\nu x)\overline{a}\langle x\rangle = (\nu y)\overline{a}\langle y\rangle$ if $x, y\#a$.

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

Definition 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* Ψ; *and*
3. $\mathbf{C}$, *the* conditions, *ranged over by* φ.

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:*

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

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 [2] for the details.

Definition 2 (Static equivalence). *Two assertions Ψ, Ψ' are statically equivalent, written $\Psi \simeq \Psi'$, if $\forall \varphi.\, \Psi \vdash \varphi \Leftrightarrow \Psi' \vdash \varphi$.*

Definition 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 $\dot{\rightarrow}$, and required that channel equivalence be symmetric (formally, $\Psi \vdash M \leftrightarrow K$ iff $\Psi \vdash K \leftrightarrow M$) and transitive.

Definition 4 (Process syntax). *The* processes *(or* agents*)* $\mathbf{P}$, *ranged over by* P, Q, R, *are inductively defined by the grammar*

$$
\begin{array}{llll}
P := & \mathbf{0} & \text{(nil)} & (\!|\Psi|\!) & \text{(assertion)} \\
& \overline{M}\,N.P & \text{(output)} & \underline{M}(\lambda\widetilde{x})N.P & \text{(input)} \\
& \mathbf{case}\ \widetilde{\varphi} : \widetilde{P} & \text{(case)} & P \mid Q & \text{(parallel composition)} \\
& (\nu x)P & \text{(restriction)} & !P & \text{(replication)}
\end{array}
$$

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

Restriction, replication and parallel composition are standard. $\overline{M}\,N.P$ is a process ready to send the message N on channel M, and then continue as P. Similarly, $\underline{M}(\lambda\widetilde{x})N.P$ is a process ready to receive a message on channel M that matches the pattern $(\lambda\widetilde{x})N$. The process $(\!|\Psi|\!)$ asserts a fact Ψ about the environment. Intuitively, $(\!|\Psi|\!) \mid P$ means that P executes in an environment where all conditions entailed by Ψ 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. $\mathbf{case}\ \widetilde{\varphi} : \widetilde{P}$ is a process that may act as any P_i whose guard φ_i is entailed by the environment. For discussion of why replication and case must be guarded we refer to [2,15].

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

Definition 5 (Frames). *The* frame *of P, written* $\mathcal{F}(P) = (\nu\widetilde{b}_P)\Psi_P$ *where* $\widetilde{b}_P$ *bind into* Ψ_P, *is defined as*

$$
\mathcal{F}((\!|\Psi|\!)) = (\nu\epsilon)\Psi \qquad \mathcal{F}(P \mid Q) = \mathcal{F}(P) \otimes \mathcal{F}(Q) \qquad \mathcal{F}((\nu x)P) = (\nu x)\mathcal{F}(P)
$$

$$
\mathcal{F}(P) = \mathbf{1}\ otherwise
$$

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

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

We extend entailment to frames as follows: $\mathcal{F}(P) \vdash \varphi$ holds if, for some $\widetilde{b}_P, \Psi_P$ such that $\mathcal{F}(P) = (\nu \widetilde{b}_P)\Psi_P$ and $\widetilde{b}_P \# \varphi$, $\Psi_P \vdash \varphi$. The freshness side-condition $\widetilde{b}_P \# \varphi$ 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 ν-binding we recover locality by writing e.g. $(\nu x)((\!|x = M|\!) \mid P)$ 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 6 (Provenances). *The* provenances Π*, ranged over by* π*, are either* $\perp$ *or of form* $(\nu \widetilde{x}; \widetilde{y})M$*, where* M *is a term, and* $\widetilde{x}, \widetilde{y}$ *bind into* M*.*

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

Intuitively, a provenance describes the origin of an input or output transition. For example, if an output transition is annotated with $(\nu \widetilde{x}; \widetilde{y})M$, the sender is an output prefix with subject M that occurs underneath the ν-binders $\widetilde{x}, \widetilde{y}$. For technical reasons, these binders are partitioned into two distinct sequences. The intention is that $\widetilde{x}$ are the frame binders, while $\widetilde{y}$ 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 $\widetilde{x}$ binders are used for deriving $\vdash$ judgements, but $\widetilde{y}$ are not (cf. Definition 5).

Definition 7 (Labels). *The* labels $\mathbf{L}$*, ranged over by* α, β*, are:*

$$\overline{M}\,(\nu \widetilde{x})N \ \textit{(output)} \qquad \underline{M}\,N \ \textit{(input)} \qquad \tau \ \textit{(silent)}$$

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

Definition 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,* $\mathrm{bn}(\alpha)$ *binds into* $\mathrm{obj}(\alpha)$ *and* P'*.*

The operational semantics differs from [2] mainly by the inclusion of provenances: anything underneath the transition arrows is novel.

$$\text{In } \frac{\Psi \vdash K \stackrel{.}{\to} M}{\Psi \,\triangleright\, \underline{M}(\lambda \widetilde{y})N . P \xrightarrow[M]{K\,N[\widetilde{y}:=\widetilde{L}]} P[\widetilde{y}:=\widetilde{L}]} \qquad\qquad \text{Out } \frac{\Psi \vdash M \stackrel{.}{\to} K}{\Psi \,\triangleright\, \overline{M}\,N . P \xrightarrow[M]{\overline{K}N} P}$$

$$\text{ParL } \frac{\Psi_Q \otimes \Psi \,\triangleright\, P \xrightarrow[\pi]{\alpha} P'}{\Psi \,\triangleright\, P \mid Q \xrightarrow[\pi\downarrow\widetilde{b}_Q]{\alpha} P' \mid Q} \; \mathrm{bn}(\alpha)\#Q$$

$$\text{ParR } \frac{\Psi_P \otimes \Psi \,\triangleright\, Q \xrightarrow[\pi]{\alpha} Q'}{\Psi \,\triangleright\, P \mid Q \xrightarrow[(\nu\widetilde{b}_P)\pi]{\alpha} P \mid Q'} \; \mathrm{bn}(\alpha)\#P$$

$$\text{Com } \frac{\Psi_Q \otimes \Psi \,\triangleright\, P \xrightarrow[(\nu\widetilde{b}_P;\widetilde{x})K]{\overline{M}(\nu\widetilde{a})N} P' \qquad \Psi_P \otimes \Psi \,\triangleright\, Q \xrightarrow[(\nu\widetilde{b}_Q;\widetilde{y})M]{K\,N} Q'}{\Psi \,\triangleright\, P \mid Q \xrightarrow[\bot]{\tau} (\nu\widetilde{a})(P' \mid Q')} \; \widetilde{a}\#Q$$

$$\text{Case } \frac{\Psi \,\triangleright\, P_i \xrightarrow[\pi]{\alpha} P' \qquad \Psi \vdash \varphi_i}{\Psi \,\triangleright\, \mathbf{case}\ \widetilde{\varphi} : \widetilde{P} \xrightarrow[\pi\downarrow]{\alpha} P'} \qquad \text{Scope } \frac{\Psi \,\triangleright\, P \xrightarrow[\pi]{\alpha} P'}{\Psi \,\triangleright\, (\nu b)P \xrightarrow[(\nu b)\pi]{\alpha} (\nu b)P'} \; b\#\alpha,\Psi$$

$$\text{Open } \frac{\Psi \,\triangleright\, P \xrightarrow[\pi]{\overline{M}(\nu\widetilde{a})N} P'}{\Psi \,\triangleright\, (\nu b)P \xrightarrow[(\nu b)\pi]{\overline{M}(\nu\widetilde{a}\cup\{b\})N} P'} \begin{array}{l} b\#\widetilde{a},\Psi,M \\ b \in \mathrm{n}(N) \end{array} \qquad \text{Rep } \frac{\Psi \,\triangleright\, P \mid !P \xrightarrow[\pi]{\alpha} P'}{\Psi \,\triangleright\, !P \xrightarrow[\pi\downarrow]{\alpha} P'}$$

Table 1. Structured operational semantics. A symmetric version of Com is elided. In the rule Com we assume that $\mathcal{F}(P) = (\nu\widetilde{b}_P)\Psi_P$ and $\mathcal{F}(Q) = (\nu\widetilde{b}_Q)\Psi_Q$ where $\widetilde{b}_P$ is fresh for Ψ and Q, $\widetilde{x}$ is fresh for Ψ, Ψ_Q, P, and $\widetilde{b}_Q, \widetilde{y}$ are similarly fresh. In rule ParL we assume that $\mathcal{F}(Q) = (\nu\widetilde{b}_Q)\Psi_Q$ where $\widetilde{b}_Q$ is fresh for Ψ, P, π and α. ParR has the same freshness conditions but with the roles of P, Q swapped. In Open the expression $\widetilde{a} \cup \{b\}$ means the sequence $\widetilde{a}$ with b inserted anywhere.

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 subtitution 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 occuring 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 insists that we consider

alpha-variants such that the frame binders and the outermost provenance binders coincide. This ensures that the K on Q's label really is the same as the K in the provenance.

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

$$\textsc{Com-Old} \ \frac{\Psi_P \otimes \Psi \ \triangleright \ Q \ \xrightarrow{K\,N} \ Q' \qquad \Psi \otimes \Psi_P \otimes \Psi_Q \vdash M \overset{\cdot}{\leftrightarrow} K}{\Psi \ \triangleright \ P \mid Q \ \xrightarrow{\tau} \ (\nu\widetilde{a})(P' \mid Q')} \ \widetilde{a}\#Q$$

with the top premise $\Psi_Q \otimes \Psi \ \triangleright \ P \ \xrightarrow{\overline{M}\,(\nu\widetilde{a})N} \ P'$

where $\mathcal{F}(P) = (\nu\widetilde{b}_P)\Psi_P$ and $\mathcal{F}(Q) = (\nu\widetilde{b}_Q)\Psi_Q$ and $\widetilde{b}_P\#\Psi, \widetilde{b}_Q, Q, M, P$ and $\widetilde{b}_Q\#\Psi, \widetilde{b}_Q, Q, K, P$. 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 $\widetilde{b}_P\#M$ and $\widetilde{b}_Q\#K$, but not $\widetilde{b}_P\#K$ and $\widetilde{b}_Q\#M$: 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 [14, pp. 56-57]. With COM, it becomes clear why: because $\widetilde{b}_Q$ 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 π 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 5).

3   Meta-theory

In this section, we will derive the standard algebraic and congruence laws of strong 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. While weak equivalences are beyond the scope of the present paper, we believe it is possible (if tedious) to adapt the results about weak bisimilarity from [15] to our setting.

3.1   Bisimulation

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

Definition 9 (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 $\Psi \rhd P \xrightarrow{\alpha} P'$ and $\mathrm{bn}(\alpha)\#\Psi, Q$, then there exists Q' such that $\Psi \rhd Q \xrightarrow{\alpha} Q'$ and $(\Psi, P', Q') \in \mathcal{R}$ (simulation)*

   We let bisimilarity $\overset{\cdot}{\sim}$ *be the largest bisimulation. We write $P \overset{\cdot}{\sim}_\Psi Q$ to mean $(\Psi, P, Q) \in \overset{\cdot}{\sim}$, and $P \overset{\cdot}{\sim} Q$ for $P \overset{\cdot}{\sim}_{\mathbf{1}} Q$*

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, irrespectively 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 Ψ, K we have $\Psi \vdash M \overset{.}{\to} K$ iff $\Psi \vdash N \overset{.}{\to} K$. 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—formally proven in Isabelle, by a routine induction—hints at:

Lemma 1. *(Find connected provenance)*

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

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. [2, Lemma 5.11]). By Lemma 1 we already have a sufficiently fresh subject: our communication partner's provenance.

Theorem 1 (Congruence properties of strong bisimulation).

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

Theorem 2 (Algebraic laws of strong bisimulation).

$$P \stackrel{\cdot}{\sim}_{\Psi} P \mid \mathbf{0} \qquad P \mid (Q \mid R) \stackrel{\cdot}{\sim}_{\Psi} (P \mid Q) \mid R \qquad P \mid Q \stackrel{\cdot}{\sim}_{\Psi} Q \mid P \qquad (\nu a)\mathbf{0} \stackrel{\cdot}{\sim}_{\Psi} \mathbf{0}$$

$$P \mid (\nu a)Q \stackrel{\cdot}{\sim}_{\Psi} (\nu a)(P \mid Q) \text{ if } a\#P \qquad \overline{M} \, N.(\nu a)P \stackrel{\cdot}{\sim}_{\Psi} (\nu a)\overline{M} \, N.P \text{ if } a\#M, N$$

$$\underline{M}(\lambda \widetilde{x})N.(\nu a)P \stackrel{\cdot}{\sim}_{\Psi} (\nu a)\underline{M}(\lambda \widetilde{x})N.P \text{ if } a\#\widetilde{x}, M, N \qquad !P \stackrel{\cdot}{\sim}_{\Psi} P \mid !P$$

$$\mathbf{case} \ \widetilde{\varphi} : \widetilde{(\nu a)P} \stackrel{\cdot}{\sim}_{\Psi} (\nu a)\mathbf{case} \ \widetilde{\varphi} : \widetilde{P} \text{ if } a\#\widetilde{\varphi} \qquad (\nu a)(\nu b)P \stackrel{\cdot}{\sim}_{\Psi} (\nu b)(\nu a)P$$

The proofs of Theorems 1 and 2 have been mechanised in Nominal Isabelle. 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 [2].

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:

$$\frac{\Psi_Q \otimes \Psi \rhd P \xrightarrow{\overline{M} \, (\nu \widetilde{a})N} P' \qquad \Psi_P \otimes \Psi \rhd Q \xrightarrow{K \, N} Q' \qquad S}{\Psi \rhd P \mid Q \xrightarrow{\tau} (\nu \widetilde{a})(P' \mid Q')}$$

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

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

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.

3.2   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. [2], i.e., without provenances and with the COM-OLD rule discussed in Section 2. The following result has been mechanised in Nominal Isabelle:

Theorem 3 (Conservativity). *When $\dot{\rightarrow}$ is symmetric and transitive we have* $\dot{\sim}_o = \dot{\sim}$ *and* $\longrightarrow_o = \longrightarrow$.

Our reduction semantics departs from standard designs [4,17] by relying on reduction contexts [7] instead of structural rules, for two reasons. First, standard formulations tend to include rules like these:

$$\frac{P \ \longrightarrow \ P'}{P \mid Q \ \longrightarrow \ P' \mid Q} \qquad \frac{}{\alpha.P + Q \mid \overline{\alpha}.R + S \ \longrightarrow \ P \mid R}$$

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

$$\begin{aligned} C := \ &P_G \quad \textit{(process)} \quad &&[\,] &&\textit{(hole)} \\ &C \mid C \ \textit{(parallel)} \quad &&\textbf{case } \widetilde{\varphi} : \widetilde{P_G} \ [\!] \ \varphi' : C \ [\!] \ \widetilde{\varphi''} : \widetilde{Q_G} \quad &&\textit{(case)} \end{aligned}$$

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

$$\text{STRUCT} \dfrac{P \equiv Q \qquad Q \longrightarrow Q' \qquad Q' \equiv P'}{P \longrightarrow P'} \qquad\qquad \text{SCOPE} \dfrac{P \longrightarrow Q}{(\nu a)P \longrightarrow (\nu a)Q}$$

$$\text{CTXT} \dfrac{\widetilde{\Psi} \vdash M \overset{.}{\to} N \qquad K = L[\widetilde{x} := \widetilde{T}] \qquad \forall \varphi \in \mathrm{conds}(C).\ \widetilde{\Psi} \vdash \varphi}{\widetilde{(\!|\Psi|\!)} \mid C[\overline{M}\,K.P,\ \underline{N}(\lambda \widetilde{x})L.Q] \ \longrightarrow\ \widetilde{(\!|\Psi|\!)} \mid P \mid Q[\widetilde{x} := \widetilde{T}] \mid \mathrm{ppr}(C)}$$

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

We let *structural congruence* $\equiv$ be the smallest equivalence relation on processes derivable using Theorems 1 and 2. The *conditions* $\mathrm{conds}(C)$ and *parallel processes* $\mathrm{ppr}(C)$ 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:

$$\mathrm{ppr}(P_G) = P_G \qquad \mathrm{ppr}([\,]) = \mathbf{0} \qquad \mathrm{ppr}(C_1 \mid C_2) = \mathrm{ppr}(C_1) \mid \mathrm{ppr}(C_2)$$

$$\mathrm{ppr}(\mathbf{case}\ \widetilde{\varphi} : \widetilde{P_G} \,[\!]\, \varphi' : C \,[\!]\, \widetilde{\varphi''} : \widetilde{Q_G}) = \mathrm{ppr}(C) \qquad \mathrm{conds}(P_G) = \emptyset$$

$$\mathrm{conds}([\,]) = \emptyset \qquad \mathrm{conds}(C_1 \mid C_2) = \mathrm{conds}(C_1) \cup \mathrm{conds}(C_2)$$

$$\mathrm{conds}(\mathbf{case}\ \widetilde{\varphi} : \widetilde{P_G} \,[\!]\, \varphi' : C \,[\!]\, \widetilde{\varphi''} : \widetilde{Q_G}) = \{\varphi'\} \cup \mathrm{conds}(C)$$

Definition 11 (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 $\mathrm{ppr}(C)$ in the reduct makes sure any processes in parallel to the holes are preserved.

Theorem 4. $P \longrightarrow P'$ *iff there is* P'' *such that* $\mathbf{1} \rhd P \xrightarrow{\ \tau\ } P''$ *and* $P'' \equiv P'$

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. [15] 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 $P \downarrow_{\overline{M}(\nu \widetilde{a})N}$ (P exposes $\overline{M}(\nu \widetilde{a})N$) to mean $\exists P'.\ \mathbf{1} \rhd P \xrightarrow{\ \overline{M}(\nu \widetilde{a})N\ } P'$. We write $P \downarrow_{\overline{M}}$ for $\exists \widetilde{a}, N.P \downarrow_{\overline{M}(\nu \widetilde{A})N}$, and $P \Downarrow_{\alpha}$ for $P \xrightarrow{\ \tau\ }^{\star} \downarrow_{\alpha}$. Second, we let observers use *static* contexts, i.e. ones built from parallel and restriction.

Definition 12 (Barbed bisimilarity). Barbed bisimilarity, *written* $\overset{\cdot}{\sim}_{\mathrm{barb}}$, *is the largest equivalence on processes such that* $P \overset{\cdot}{\sim}_{\mathrm{barb}} Q$ *implies*

1. *If* $P \downarrow_{\overline{M}(\nu\widetilde{a})N}$ *and* $\widetilde{a}\#Q$ *then* $Q \downarrow_{\overline{M}(\nu\widetilde{a})N}$ *(barb similarity)*
2. *If* $P \longrightarrow P'$ *then there exists* Q' *such that* $Q \longrightarrow Q'$ *and* $P' \overset{\cdot}{\sim}_{\mathrm{barb}} Q'$ *(reduction simulation)*
3. $(\nu\widetilde{a})(P \mid R) \overset{\cdot}{\sim}_{\mathrm{barb}} (\nu\widetilde{a})(Q \mid R)$ *(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, and psi-calculi where no transitions whatsoever are possible.

Definition 13. *A psi-calculus is* observational *if:*

1. *For all* P *there are* M_P, K_P *such that* $\mathcal{F}(P) \vdash M_P \overset{\cdot}{\to} K_P$ *and not* $P \Downarrow_{\overline{K_p}}$.
2. *If* $N = (\widetilde{x}\ \widetilde{y}) \cdot M$ *and* $\widetilde{y}\#M$ *and* $\widetilde{x}, \widetilde{y}$ *are distinct then* $M[\widetilde{x} := \widetilde{y}] = N$.

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. [15]. Ours are given a slightly weaker formulation.

We can now state the main result of this section:

Theorem 5. *In all observational psi-calculi,* $P \overset{\cdot}{\sim}_{\mathrm{barb}} Q$ *iff* $P \overset{\cdot}{\sim}_1 Q$.

4   Expressiveness

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

4.1   Pi-calculus with preorders

Recall that pi-F [25] 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. $xy.P \mid \overline{x}z.Q$ reduces to $y = z \mid P \mid Q$: the input and output objects are fused. Hirschkoff et al. [11] observed that fusion and subtyping are fundamentally incompatible, and propose a generalisation of pi-F called the *pi-calculus with preorders* or πP to resolve the issue.

We are interested in πP 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 $x(y).P \mid \overline{x}(z).Q$ reduces to $(\nu xy)(z/y \mid P \mid Q)$. We write $\prec$ for the reflexive and transitive closure of the "is above" relation. Two names x, y are considered *joinable* for the purposes of synchronisation if some name z is above both of them: formally, we write $x \curlyvee y$ for $\exists z.x \prec z \wedge y \prec z$.

Hirschkoff et al. conclude by saying that "[it] could also be interesting to study the representation of πP 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" [11, p. 387]. Having lifted the constraint that channels form an equivalence relation, we happily accept the challenge. We write ΨP for the psi-calculus we use to embed πP. We follow the presentation of πP from [12,13], where the behavioural theory is most developed.

Definition 14. *The psi-calculus ΨP is defined with the following parameters:*

$$\mathbf{T} \triangleq \mathcal{N} \qquad \mathbf{C} \triangleq \{x \prec y : x, y \in \mathcal{N}\} \cup \{x \curlyvee y : x, y \in \mathcal{N}\}$$

$$\mathbf{A} \triangleq \mathcal{P}_{fin}(\{x \prec y : x, y \in \mathcal{N}\}) \qquad \mathbf{1} \triangleq \{\} \qquad \otimes \triangleq \cup$$

$$\dot{\rightarrow} \triangleq \curlyvee \qquad \vdash \triangleq \text{the relation denoted } \vdash \text{ in [13].}$$

The prefix operators of πP 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.[2] These are encodable as follows:

Definition 15 (Encoding of prefixes). *The encoding $\llbracket \_ \rrbracket$ from πP to ΨP is homomorphic on all operators except prefixes and arcs, where it is defined by*

$$\llbracket a/b \rrbracket = (\!| b \prec a |\!) \qquad \llbracket \overline{a}(y).P \rrbracket = (\nu xy)(\overline{a}x.((\!| x \prec y |\!) \mid \llbracket P \rrbracket)) \text{ where } x \# y, P$$

$$\llbracket a(y).P \rrbracket = (\nu y)(\underline{a}(\lambda x)x.((\!| y \prec x |\!) \mid \llbracket P \rrbracket)) \text{ where } x \# y, P$$

$$\llbracket [\varphi]\tau.P \rrbracket = \mathbf{case}\ \varphi : (\nu x)(\underline{x}(\lambda x)x.0 \mid \overline{x}x.\llbracket P \rrbracket) \text{ where } x \# P$$

This embedding of πP in psi-calculi comes with a notion of bisimilarity per Definition 9. We show that it coincides with the labelled bisimilarity for πP (written $\sim$) introduced in [12,13].

Theorem 6. $P \sim Q$ iff $\llbracket P \rrbracket \stackrel{.}{\sim} \llbracket Q \rrbracket$

Thus our encoding validates the behavioural theory of πP 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

---

[2] We ignore protected prefixes because they are redundant, cf. Remark 1 of [12].

designs. While we do rewriting of subjects in the prefix rules, Hirschkoff et al. instead use relabelling rules like this one (mildly edited to match our notation):

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

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.

πP also has labels $[\varphi]\tau$, meaning that a silent transition is allowed in environments where φ is true. A rule for rewriting φ 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. πP transitions of kind $P \xrightarrow{[\varphi]\tau} P'$ correspond to ΨP transitions of kind $\{\varphi\} \rhd P \xrightarrow{\tau} P'$.

Interestingly, the analogous full abstraction result fails to hold for the embedding of pi-F in psi-calculi by Bengtson et al. [2], because outputs that emit distinct but fused names are distinguished by psi-calculus bisimilarity. This issue does not arise here because πP objects are always bound; however, we believe the encoding of Bengtson et al. can be 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** $\top : P \;[\!]\; \top : Q$ for some condition $\top$ that is always entailed. *Mixed choice* means that in $P + Q$, P and Q must be prefix-guarded; that is, the outermost operators of P, Q must be input or output prefixes. 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 $\mathcal{P} = (\mathbf{T}, \mathbf{A}, \mathbf{C}, \vdash, \otimes, \mathbf{1}, \dot{\to})$ with mixed choice; this will be our source language. 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 α_x for tagging the subject of the prefix α 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*.

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

$$[\![(\!|\Psi|\!)]\!] = (\!|(\Psi, \emptyset)|\!) \qquad [\![\alpha.P + \beta.Q]\!] = (\nu x)(\alpha_x.([\![P]\!] \mid (\!|(\mathbf{1}, \{x\})|\!)) \mid \beta_x.([\![Q]\!] \mid (\!|(\mathbf{1}, \{x\})|\!))$$

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 $\mathcal{E}(\mathcal{P})$:

$$(\Psi, \mathbf{N}) \vdash M_x \dot{\to} N_y \quad \text{if } \Psi \vdash M \dot{\to} N \text{ and } x \neq y \text{ and } x, y \notin \mathbf{N}$$

That is, tagged channels are connected if the underlying channel is connected. To prevent (1) we require the tags to be different, and to prevent (2) we require 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.

Theorem 7 (Correctness of choice encoding).

1. *If $\Psi \rhd P \xrightarrow{\alpha} P'$ then there is P'' such that $(\Psi, \emptyset) \rhd [\![P]\!] \xrightarrow{\alpha} P''$ and $P'' \dot{\sim}_{(\Psi,\emptyset)} [\![P']\!]$.*
2. *If $(\Psi, \emptyset) \rhd [\![P]\!] \xrightarrow{\alpha} P'$ then there is P'' such that $\Psi \rhd P \xrightarrow{\alpha_\perp} P''$ and $P' \dot{\sim}_{(\Psi,\emptyset)} [\![P'']\!]$.*
3. *$P \dot{\sim}_{\mathbf{1}} Q$ iff $[\![P]\!] \dot{\sim}_{(\mathbf{1},\emptyset)} [\![Q]\!]$.*

Here $\alpha_\perp$ denote the label α with all tags removed. It is immediate from Theorem 7 and the definition of $[\![\_]\!]$ that our encoding also satisfies the other standard quality criteria [10]: 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 non-transitive connectivity. If we remove the requirement that tags are distinct, and only allow *separate* choice (where either both summands are inputs or both summands are outputs), the encoding is correct for the original psi-calculi.

These results generalise in a straightforward way to mixed CASE statements **case** $\varphi_1 : \alpha.P \; [\!] \; \varphi_2 : \beta.Q$ by additionally tagging terms with a condition, i.e. M_{x,φ_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 Hirschkoff et al. [11] is closely related in that it uses non-transitive connectivity; see Section 4.1 for an extensive discussion.

Broadcast psi-calculi [5] 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. [16] 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 [15] (see Section 3.2), 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 [10,20,21,22,23], 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 [6]. 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

References

1. Åman Pohjola, J.: Psi-calculi revisited: Connectivity and compositionality. Tech. Rep. EP192416, CSIRO, Canberra, Australia (2019), `https://publications.csiro.au/rpr/pub?pid=csiro:EP192416`
2. Bengtson, J., Johansson, M., Parrow, J., Victor, B.: Psi-calculi: A framework for mobile processes with nominal data and logic. Logical Methods in Computer Science **7**(1) (2011). https://doi.org/10.2168/LMCS-7(1:11)2011
3. Bengtson, J., Parrow, J., Weber, T.: Psi-calculi in Isabelle. J. Autom. Reasoning **56**(1), 1–47 (2016). https://doi.org/10.1007/s10817-015-9336-2
4. Berry, G., Boudol, G.: The chemical abstract machine. In: Proceedings of the 17th ACM SIGPLAN-SIGACT symposium on Principles of programming languages. pp. 81–94. POPL '90, ACM, New York, NY, USA (1990). https://doi.org/10.1145/96709.96717
5. Borgström, J., Huang, S., Johansson, M., Raabjerg, P., Victor, B., Åman Pohjola, J., Parrow, J.: Broadcast psi-calculi with an application to wireless protocols. Software and System Modeling **14**(1), 201–216 (2015). https://doi.org/10.1007/s10270-013-0375-z
6. Busi, N., Gorrieri, R.: Distributed conflicts in communicating systems. In: Ciancarini, P., Nierstrasz, O., Yonezawa, A. (eds.) 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. Lecture Notes in Computer Science, vol. 924, pp. 49–65. Springer (1994). https://doi.org/10.1007/3-540-59450-7_4
7. Felleisen, M., Hieb, R.: The revised report on the syntactic theories of sequential control and state. Theor. Comput. Sci. **103**(2), 235–271 (1992). https://doi.org/10.1016/0304-3975(92)90014-7
8. Gabbay, M.J., Pitts, A.M.: A new approach to abstract syntax with variable binding. Formal Aspects of Computing **13**, 341–363 (2002). https://doi.org/10.1007/s001650200016
9. Gardner, P., Wischik, L.: Explicit fusions. In: Nielsen, M., Rovan, B. (eds.) Proceedings of MFCS 2000. vol. 1893, pp. 373–382 (2000). https://doi.org/10.1007/3-540-44612-5_33
10. Gorla, D.: Towards a unified approach to encodability and separation results for process calculi. Inf. Comput. **208**(9), 1031–1053 (2010). https://doi.org/10.1016/j.ic.2010.05.002
11. Hirschkoff, D., Madiot, J., Sangiorgi, D.: 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. pp. 378–387. IEEE Computer Society (2013). https://doi.org/10.1109/LICS.2013.44
12. Hirschkoff, D., Madiot, J., Xu, X.: A behavioural theory for a π-calculus with preorders. J. Log. Algebr. Meth. Program. **84**(6), 806–825 (2015). https://doi.org/10.1016/j.jlamp.2015.07.001
13. Hirschkoff, D., Madiot, J., Xu, X.: A behavioural theory for a π-calculus with preorders. In: Dastani, M., Sirjani, M. (eds.) Fundamentals of Software Engineering - 6th International Conference, FSEN 2015 Tehran, Iran, April 22-24, 2015, Revised Selected Papers. Lecture Notes in Computer Science, vol. 9392, pp. 143–158. Springer (2015). https://doi.org/10.1007/978-3-319-24644-4_10
14. Johansson, M.: Psi-calculi: a framework for mobile process calculi: Cook your own correct process calculus - just add data and logic. Ph.D. thesis, Uppsala University, Division of Computer Systems (2010)

15. Johansson, M., Bengtson, J., Parrow, J., Victor, B.: Weak equivalences in psi-calculi. In: LICS. pp. 322–331. IEEE Computer Society (2010). https://doi.org/10.1109/LICS.2010.30
16. Kouzapas, D., Gutkovas, R., Gay, S.J.: Session types for broadcasting. In: Donaldson, A.F., Vasconcelos, V.T. (eds.) Proceedings 7th Workshop on Programming Language Approaches to Concurrency and Communication-cEntric Software, PLACES 2014, Grenoble, France, 12 April 2014. EPTCS, vol. 155, pp. 25–31 (2014). https://doi.org/10.4204/EPTCS.155.4
17. Milner, R.: Functions as processes. In: Proceedings of the seventeenth international colloquium on Automata, languages and programming. pp. 167–180. Springer-Verlag New York, Inc., New York, NY, USA (1990). https://doi.org/10.1007/BFb0032030
18. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, part I/II. Inf. Comput. **100**(1), 1–77 (1992). https://doi.org/10.1016/0890-5401(92)90008-4
19. Nanz, S., Hankin, C.: A framework for security analysis of mobile wireless networks. Theor. Comput. Sci. **367**(1-2), 203–227 (2006)
20. Nestmann, U., Pierce, B.C.: Decoding choice encodings. Inf. Comput. **163**(1), 1–59 (2000). https://doi.org/10.1006/inco.2000.2868
21. Palamidessi, C.: Comparing the expressive power of the synchronous and the asynchronous pi-calculus. In: Lee, P., Henglein, F., Jones, N.D. (eds.) 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. pp. 256–265. ACM Press (1997). https://doi.org/10.1145/263699.263731
22. Peters, K., Nestmann, U.: Is it a "good" encoding of mixed choice? In: Birkedal, L. (ed.) 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. Lecture Notes in Computer Science, vol. 7213, pp. 210–224. Springer (2012). https://doi.org/10.1007/978-3-642-28729-9_14
23. Peters, K., Nestmann, U., Goltz, U.: On distributability in process calculi. In: Felleisen, M., Gardner, P. (eds.) 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. Lecture Notes in Computer Science, vol. 7792, pp. 310–329. Springer (2013). https://doi.org/10.1007/978-3-642-37036-6_18
24. Urban, C.: Nominal techniques in Isabelle/HOL. Journal of Automated Reasoning **40**(4), 327–356 (May 2008). https://doi.org/10.1007/s10817-008-9097-2
25. Wischik, L., Gardner, P.: Explicit fusions. Theoretical Computer Science **304**(3), 606–630 (2005). https://doi.org/10.1016/j.tcs.2005.03.017