Bisimulations Respecting Duration and Causality for the Non-interleaving Applied π-Calculus

Clément Aubert
Augusta University, USA
caubert@augusta.edu

Ross Horne
University of Luxembourg, Luxembourg
ross.horne@uni.lu

Christian Johansen
NTNU, Norway
christian.johansen@ntnu.no

This paper shows how we can make use of an asynchronous transition system, whose transitions are labelled with events and which is equipped with a notion of independence of events, to define non-interleaving semantics for the applied π-calculus. The most important notions we define are: Start-Termination or ST-bisimilarity, preserving duration of events; and History-Preserving or HP-bisimilarity, preserving causality. We point out that corresponding similarity preorders expose clearly distinctions between these semantics. We draw particular attention to the distinguishing power of HP failure similarity, and discuss how it affects the attacker threat model against which we verify security and privacy properties. We also compare existing notions of located bisimilarity to the definitions we introduce.

1 Introduction

Non-interleaving semantics is sometimes referred to as true concurrency. This reflects the idea that parallel composition has a semantically distinct status from its interleavings obtained by allowing each parallel process to preform actions one-by-one in any order. In this work, we explore a spectrum of non-interleaving semantics for the applied π-calculus, which is motivated by some recent works on modelling and verifying security and privacy properties of cryptographic protocols [9, 21]. The definitions introduced are operational in style, bypassing denotations such as event structures.

We build on our recent work [4] that introduced a non-interleaving Structural Operational Semantics (SOS) for the applied π-calculus that generates Labelled Asynchronous Transition Systems (LATS). Compared with standard transition systems, whose transitions are labelled with actions, a LATS labels its transitions with richer *events*, and is equipped with a notion of *independence* over adjacent events (concurrently enabled or enabled one after another). A LATS allows independent events to be permuted and hence techniques such as partial-order reduction to be applied. This work is part of a research agenda where we wish to lay a foundation for exploring questions such as whether verification techniques are enabled by adopting a semantics that is naturally compatible with an independence relation used for partial-order reduction. Another research question is whether adopting a non-interleaving semantics impacts the attacker model for certain problems. In particular, armed with our definitions, we may ask whether our non-interleaving semantics may detect attacks that may be missed if we employ an interleaving semantics.

The contribution of this paper towards addressing the questions above is the introduction of non-interleaving equivalences and similarities that can be defined for the applied π-calculus equipped with a LATS [4]. A well understood starting point is how to generate "located" equivalences [6] for CCS [7, 24] and the π-calculus [27]. The former approach makes direct use of the LATS for CCS, while the latter uses a cut down located transition system for the π-calculus which accounts for locations but does not satisfy all properties of a LATS. We go further since, given our LATS, we can generate in an operational

V. Castiglioni and C. A. Mezzina (Eds): Combined Workshop on Expressiveness in Concurrency and Structural Operational Semantics 2022 (EXPRESS/SOS 2022).
EPTCS 368, 2022, pp. 3–22, doi:10.4204/EPTCS.368.1

Terminology	*Remarks*	*Def.*
i-similarity	"Interleaving"-similarity is the notion of similarity most commonly explored in the literature.	Def. 8
ST-similarity	"Start-Terminate"-similarity accounts for the fact that events have duration. It uses events to distinguish between actions with the same label, and to ensure that two "terminate" events correspond to the same "start" event.	Def. 11
HP-similarity	"History-Preserving"-similarity preserves the causal dependencies between events.	Def. 12
I-similarity	"Independence"-similarity are parametrised by some notion of independence I. We obtain "located bisimilarities" using the structural independence relation I_ℓ that considers only if two events are in different locations.	Def. 16

Table 1: Strategies in the interleaving/non-interleaving spectrum explored for the applied π-calculus.

style other notions of non-interleaving semantics, particularly those that preserve duration of events (Start-Termination or ST semantics) [15] and those that preserve causality (History-Preserving or HP semantics) [12, 26]. Since we cover the applied π-calculus, of course, we encompass the π-calculus, where the later surprisingly benefits from adopting a modern applied π-calculus style when handling *link causality* – the causal relationship between outputs and inputs that depend upon them. Our operational approach also avoids the need to unfold to event structures [10, 30] or configuration structures [11] that would track entire histories of causal dependencies; instead, we consider only what is happening or enabled at a particular point in time.

We include in Tables 1 and 2 a glossary, including key standard and non-standard terminology employed in this paper. We emphasise similarity rather than bisimilarity for two reasons. Firstly, similarity exposes more clearly than bisimilarity the differences between non-interleaving semantics as it allows clearer separating examples. Secondly, similarity is known to have compelling attacker models in terms of probabilistic may testing [13], and it is standard in computational security to consider probabilistic attackers [8]. Table 1 presents the notions of similarity that we discuss in the interleaving/non-interleaving spectrum we explore. Along this spectrum the attacker has different powers for observing concurrency.

While we draw attention to similarity, we are also interested in non-interleaving *bi*similarity and other notions in the linear-time/branching-time spectrum [16]. Indeed, all the notions in Table 1 also exist in their other variants in the linear-time/branching time spectrum listed in Table 2, such as failure similarity. Along this spectrum the observer has more or less power to observe and make choices. We also use the term *mutual*, e.g., mutual ST-similarity, when some notion of similarity holds in both directions.

There are further spectra that could be explored: for the π-calculus there is the open/early spectrum, including notions such as early, late, quasi-open [29], and open [28] variants of equivalences. This work considers only *early* and *strong* semantics: early semantics means that the message input is chosen at the moment the event starts, whereas the other variants allow different degrees of laziness in learning what message was input retrospectively. This choice is made since the majority of equivalences for the applied π-calculus in the literature are early, and early bisimilarity coincides with notions of testing via concurrent processes [1]. Since our semantics are strong, every τ-transition is matched by exactly one τ-transition in all our strategies. Many security and privacy problems that motivate us can be reduced to a strong equivalence problem. However, the main reason for these choices is simply to focus on the interleaving/non-interleaving spectrum. For example, it would be easy to define quasi-open variants of our non-interleaving semantics, which coincide with a testing semantics making use of all contexts [22].

Terminology	*Remarks*	*Symb.*
X-bisimilarity	An equivalence ranging over all strategies of a particular type X.	$\sim_X$
X-similarity	The preorder arising when we assume one player leads throughout a strategy (except when testing equations, as explained around Def. 8).	$\preceq_X$
X-presimilarity	A notion of similarity we introduce in this paper (Def. 7) to emphasise the testing power of inequalities in the applied π-calculus.	$\sqsubseteq_X$
Xf-similarity	X "failure" similarity is one of many variants of similarity in the linear-time/branching-time spectrum, and is chosen due to its testing model allowing us to test if something is not enabled. In particular, we look at STf-similarity (Def. 13) and HPf-similarity (Def. 14).	$\preceq_{Xf}$

Table 2: Notions in the linear-time/branching-time spectrum explored for the applied π-calculus.

After briefly recalling our non-interleaving SOS generating a LATS (Sect. 2), we use interleaving semantics to illustrate and motivate the genericity of *static* equivalences (Sect. 3). Sect. 4 is the core of our proposal: it starts by introducing and stressing the importance of the independence relation (Sect. 4.1), which is used throughout the rest of the article. ST and HP-similarities are then defined in Sect. 4.2 and 4.3 and compared in the context of privacy in Sect. 4.4. Sect. 4.5 discusses failure semantics for HP- and ST-similarities. Some design decisions are justified in light of located bisimulations in Sect. 5.

2 Background: A Non-interleaving SOS for the Applied π-Calculus

This section recalls a non-interleaving structural operational semantics for the applied π-calculus. The design decisions are discussed extensively in a companion paper [4]. What we present below is intended only as a condensed summary of that operational semantics for ease of reference.

All variables x,y,z are the same syntactic category, but are distinct from *aliases*. Aliases range over α,β,γ and consist of an alias variable, say λ, prefixed with a string $s \in \{0,1\}^*$, i.e., $\alpha = s\lambda$. *Messages* range over M,N,K, built from a signature of function symbols Σ. As standard, a *substitution* σ,θ or ρ is a function with a domain ($\mathrm{dom}(\sigma) = \{\alpha\colon \alpha \neq \alpha\sigma\}$) and a range ($\mathrm{ran}(\sigma) = \{\alpha\sigma\colon \alpha \in \mathrm{dom}(\sigma)\}$) that are applied in suffix form. The *identity substitution* is denoted id and composition $\sigma\circ\theta$.

Processes are denoted by P,Q,R, and in $\nu x.P$ and $a(x).P$ occurrences of x in P are bound. Sequences of names $\nu\vec{x}.P$ abbreviate multiple name binders defined inductively such that $\nu\varepsilon.P = P$ and $\nu x,\vec{y}.P = \nu x.\nu\vec{y}.P$, where ε is the empty sequence. *Active substitutions*, denoted σ, θ, map aliases in their finite domain to messages containing no aliases, and appear in *extended processes*, ranging over A,B,C. We assume a *normal form*, where aliases do not appear in processes, and an *equational theory* E containing equalities on messages, e.g., $\mathtt{dec}(\{M\}_K,K) =_E M$. Figs. 1 and 2 give the syntax and semantics.

Definition 1 (freshness, α-equivalence, etc.)**.** *A variable x (resp. an alias α) is* free in a message M *if* $x \in \mathrm{fv}(M)$ *(resp.* $\alpha \in \mathrm{fa}(M)$*) for*

$$\mathrm{fv}(f(M_1,\ldots M_n)) = \cup_{i=1}^n \mathrm{fv}(M_i) \qquad \mathrm{fv}(x) = \{x\} \qquad \mathrm{fv}(\alpha) = \emptyset$$
$$\mathrm{fa}(f(M_1,\ldots M_n)) = \cup_{i=1}^n \mathrm{fa}(M_i) \qquad \mathrm{fa}(x) = \emptyset \qquad \mathrm{fa}(\alpha) = \{\alpha\}\,.$$

The fv *function extends in the standard way to (extended) processes, letting* $\mathrm{fv}(\nu x.P) = \mathrm{fv}(P)\setminus\{x\}$ *and* $\mathrm{fv}(M(x).P) = \mathrm{fv}(M)\cup(\mathrm{fv}(P)\setminus\{x\})$*, and similarly for* $\mathrm{fv}(A)$*. The functions for free variables and free*

PROCESSES:

$P,Q,R ::=$	0		deadlock
	$\mid$	$\nu x.P$	new
	$\mid$	$P \mid Q$	parallel
	$\mid$	G	guarded process
	$\mid$	$!P$	replication

GUARDED PROCESSES:

$G,H ::=$	$M(x).P$		input prefix
	$\mid$	$\overline{M}\langle N\rangle.P$	output prefix
	$\mid$	$[M=N]G$	match
	$\mid$	$[M\neq N]G$	mismatch
	$\mid$	$G+H$	choice

EXTENDED PROCESSES:

$A,B ::=$	$\sigma \mid P$		active process
	$\mid$	$\nu x.A$	new

MESSAGES:

$M,N ::=$	x		variable
	$\mid$	α	alias
	$\mid$	$f(M_1,\ldots,M_n)$	function

EARLY ACTION LABELS:

$\pi ::=$	MN		free input
	$\mid$	$\overline{M}(\alpha)$	output
	$\mid$	τ	interaction

Figure 1: Syntax of extended processes with guarded choices, where $f \in \Sigma$.

aliases extend to labels as follows.

$$\mathrm{fv}(\pi) = \begin{cases} \mathrm{fv}(M)\cup\mathrm{fv}(N) & \text{if } \pi = MN \\ \mathrm{fv}(M) & \text{if } \pi = \overline{M}(\alpha) \\ \emptyset & \text{if } \pi = \tau \end{cases} \qquad \mathrm{fa}(\pi) = \begin{cases} \mathrm{fa}(M)\cup\mathrm{fa}(N) & \text{if } \pi = MN \\ \mathrm{fa}(M) & \text{if } \pi = \overline{M}(\alpha) \\ \emptyset & \text{if } \pi = \tau \end{cases}$$

We say a variable x is fresh for a message M (resp. process P, extended process A), *written $x \# M$ (resp. $x \# P$, $x \# A$) whenever $x \notin \mathrm{fv}(M)$ (resp. $x \notin \mathrm{fv}(P)$, $x \notin \mathrm{fv}(A)$), and similarly for aliases. Freshness extends point-wise to lists of entities, i.e., $x_1,x_2,\ldots x_m \# M_1,M_2,\ldots,M_n$, denotes the conjunction of all $x_i \# M_j$ for all $1 \le i \le m$ and $1 \le j \le n$.*

We define α-equivalence *(denoted $\equiv_\alpha$) for variables only (not aliases which are fixed "addresses") as the least congruence (a reflexive, transitive, and symmetric relation preserved in all contexts) such that, whenever $z \# \nu x.P$, we have $\nu x.P \equiv_\alpha \nu z.(P\{^z/_x\})$ and $M(x).P \equiv_\alpha M(z).(P\{^z/_x\})$. Similarly, for extended processes, we have the least congruence such that, whenever $z \# \nu x.A$, we have $\nu x.A \equiv_\alpha \nu z.(A\{^z/_x\})$. Restriction is such that $\theta\restriction_{\vec{\alpha}}(x) = \theta(x)$ if $x \in \vec{\alpha}$ and x otherwise.*

Capture-avoiding substitutions *are defined for processes such that $(M(x).P)\sigma \equiv_\alpha M\sigma(z).P\{^z/_x\}\sigma$ and $(\nu x.P)\sigma \equiv_\alpha \nu z.P\{^z/_x\}\sigma$ for $z \# \mathrm{dom}(\sigma),\mathrm{ran}(\sigma),\nu x.P$. For extended processes, it is defined such that $(\nu x.A)\rho \equiv_\alpha \nu z.(A(\{^z/_x\}\circ\rho))$ and $(\sigma \mid P)\rho = (\sigma\circ\rho\restriction_{\mathrm{dom}(\sigma)} \mid P\rho)$, for $z \# \mathrm{dom}(\rho),\mathrm{ran}(\rho),\nu x.A$.*

Definition 2 (structural congruence)**.** *Our minimal* structural congruence *(denoted $\equiv$) is the least equivalence relation on extended processes extending α-equivalence such that whenever $\sigma = \theta$, $P \equiv_\alpha Q$ and $A \equiv B$, we have: $\sigma \mid P \equiv \theta \mid Q$, $\nu x.A \equiv \nu x.B$ and $\nu x.\nu z.A \equiv \nu z.\nu x.A$.*

Definition 3 (location labels)**.** *A* location *ℓ is of the form $s[t]$, where $s \in \{0,1\}^*$ and $t \in \{0,1\}^*$. If s or t is empty, we omit it (hence, we write $\varepsilon[\varepsilon]$ as $[]$). A* location label *u is either a location ℓ or a pair of locations (ℓ_0,ℓ_1), and we let $c(\ell_0,\ell_1) = (c\ell_0,c\ell_1)$ for $c \in \{0,1\}$.*

3 Handling located aliases, explained using interleaving similarities

Although the objective of this paper is to explore non-interleaving semantics, we begin by defining an interleaving semantics. The reason is that we wish to expose clearly which parts of our definitions are generic to any type of semantics, and which are specific to non-interleaving semantics.

$$\frac{M =_E K}{K(x).P \xrightarrow[\square]{MN} \mathrm{id} \mid P\{^N/_x\}}\ \textsc{Inp} \qquad \frac{M =_E K}{\overline{K}\langle N\rangle.P \xrightarrow[\square]{\overline{M}(\lambda)} \{^N/_\lambda\} \mid P}\ \textsc{Out}$$

$$\frac{P \xrightarrow[u]{\pi} \nu\vec{x}.(\sigma \mid R) \quad \vec{x}\#Q}{P \mid Q \xrightarrow[0u]{\pi} \nu\vec{x}.(\sigma \mid R \mid Q)}\ \textsc{Par-L} \qquad \frac{Q \xrightarrow[u]{\pi} \nu\vec{x}.(\sigma \mid R) \quad \vec{x}\#P}{P \mid Q \xrightarrow[1u]{\pi} \nu\vec{x}.(\sigma \mid P \mid R)}\ \textsc{Par-R}$$

$$\frac{P\{^z/_x\} \xrightarrow[u]{\pi} A \qquad z\#\mathrm{fv}(\pi), \nu x.P}{\nu x.P \xrightarrow[u]{\pi} \nu z.A}\ \textsc{Extrude} \qquad \frac{A \xrightarrow[u]{\pi} B \qquad x\#\mathrm{fv}(\pi)}{\nu x.A \xrightarrow[u]{\pi} \nu x.B}\ \textsc{Res}$$

$$\frac{P \xrightarrow[s[s']]{\overline{M\sigma}(\lambda)} \nu\vec{x}.\left(\{^N/_\lambda\} \mid Q\right) \quad \vec{x}\#\mathrm{ran}(\sigma) \quad \mathrm{fa}(M) \subseteq \mathrm{dom}(\sigma) \quad s\lambda\#\mathrm{dom}(\sigma)}{\sigma \mid P \xrightarrow[s[s']]{\overline{M}(s\lambda)} \nu\vec{x}.\left(\sigma \circ \{^N/_{s\lambda}\} \mid Q\right)}\ \textsc{Alias-out}$$

$$\frac{P \xrightarrow[u]{\pi\sigma} \nu\vec{x}.(\mathrm{id} \mid Q) \quad \vec{x}\#\mathrm{ran}(\sigma) \quad \mathrm{fa}(\pi) \subseteq \mathrm{dom}(\sigma)}{\sigma \mid P \xrightarrow[u]{\pi} \nu\vec{x}.(\sigma \mid Q)}\ \textsc{Alias-free}$$

$$\frac{G \xrightarrow[[t]]{\pi} A}{G+H \xrightarrow[[0t]]{\pi} A}\ \textsc{Sum-L} \qquad \frac{H \xrightarrow[[t]]{\pi} A}{G+H \xrightarrow[[1t]]{\pi} A}\ \textsc{Sum-R} \qquad \frac{P \mid !P \xrightarrow[u]{\pi} A}{!P \xrightarrow[u]{\pi} A}\ \textsc{Bang}$$

$$\frac{P \xrightarrow[u]{\pi} A \qquad M =_E N}{[M=N]P \xrightarrow[u]{\pi} A}\ \textsc{Mat} \qquad \frac{P \xrightarrow[u]{\pi} A \qquad M \neq_E N}{[M\neq N]P \xrightarrow[u]{\pi} A}\ \textsc{Mismat}$$

$$\frac{P \xrightarrow[\ell_0]{\overline{M}(\lambda)} \nu\vec{y}.\left(\{^N/_\lambda\} \mid P'\right) \qquad Q \xrightarrow[\ell_1]{MN} \nu\vec{w}.\left(\mathrm{id} \mid Q'\right) \qquad \vec{y}\#Q \qquad \vec{w}\#P,\vec{y}}{P \mid Q \xrightarrow[(0\ell_0,1\ell_1)]{\tau} \nu\vec{y},\vec{w}.\left(\mathrm{id} \mid P' \mid Q'\right)}\ \textsc{Close-L}$$

$$\frac{P \xrightarrow[\ell_0]{MN} \nu\vec{y}.\left(\mathrm{id} \mid P'\right) \qquad Q \xrightarrow[\ell_1]{\overline{M}(\lambda)} \nu\vec{w}.\left(\{^N/_\lambda\} \mid Q'\right) \qquad \vec{w}\#P \qquad \vec{y}\#Q,\vec{w}}{P \mid Q \xrightarrow[(0\ell_0,1\ell_1)]{\tau} \nu\vec{y},\vec{w}.\left(\mathrm{id} \mid P' \mid Q'\right)}\ \textsc{Close-R}$$

Figure 2: An early non-interleaving structural operational semantics.

The first shared trait by all equivalences for the applied π-calculus is that they make use of a *static equivalence*. Its role is to prevent the attacker from using the data they know to form a test for one process that does not hold for another process. In an extended process, one can think of the active substitution as a record of the information available to an attacker observing messages communicated on public channels. The attacker can then combine that information in various ways to try to pass a test, e.g., hashing the first message and checking whether it is equal to the second message. We find it insightful to break down static equivalence into simpler definitions, that we will employ to achieve the same effect. In particular, we start with the following satisfaction relation.

Definition 4 (satisfaction)**.** *Satisfaction* $\vDash$ *is defined inductively as:*

- $\nu x.A \vDash M = N$ *whenever, for* $y\#\nu x.A,M,N$*, we have* $A\{^y/_x\} \vDash M = N$*, and also*
- $\theta \mid P \vDash M = N$ *whenever* $M\theta =_E N\theta$*.*

The above ensures that the private names in an extended process do not appear directly in M or N, leaving only the possibility of using aliases in the domain of the active substitution in M and N to indirectly refer to private names. That is, M and N are recipes that must produce the same message, up to the equational theory E, given the information recorded in the active substitution of the extended process. As a simple example, we have $\nu x.\left(\{^{x}/_{0\lambda}\} \circ \{^{h(x)}/_{1\lambda}\} \mid P\right) \vDash h(0\lambda) = 1\lambda$.

Now we can make a generic point about all reasonable notions of equivalence based on our structural operational semantics. As explained in related work [4], each alias has a location prefix, allowing each location to have its unique pool of aliases, thus ensuring that the choice of alias is localised and not impacted by choices of aliases made by concurrent threads. For example, the following process has two transitions, labelled with $(\overline{a}(0\lambda), 0[])$ and $(\overline{b}(1\lambda), 1[])$ (cf. Def. 9 for a formal definition of those *events*):

$$\nu x.\left(\{^{x}/_{0\lambda}\} \mid 0 \mid \overline{b}\langle h(x)\rangle\right) \xleftarrow[0[]]{\overline{a}(0\lambda)} \mathrm{id} \mid \nu x.\left(\overline{a}\langle x\rangle \mid \overline{b}\langle h(x)\rangle\right) \xrightarrow[1[]]{\overline{b}(1\lambda)} \nu x.\left(\left\{^{h(x)}/_{1\lambda}\right\} \mid \overline{a}\langle x\rangle \mid 0\right)$$

Clearly, any reasonable semantics should equate the above process with the one below, where the only difference is that the parallel processes $\overline{a}\langle x\rangle$ and $\overline{b}\langle h(x)\rangle$ have been permuted (e.g., exchanged their locations).

$$\nu x.\left(\{^{x}/_{1\lambda}\} \mid \overline{b}\langle h(x)\rangle \mid 0\right) \xleftarrow[1[]]{\overline{a}(1\lambda)} \mathrm{id} \mid \nu x.\left(\overline{b}\langle h(x)\rangle \mid \overline{a}\langle x\rangle\right) \xrightarrow[0[]]{\overline{b}(0\lambda)} \nu x.\left(\left\{^{h(x)}/_{0\lambda}\right\} \mid 0 \mid \overline{a}\langle x\rangle\right)$$

Notice that the events labelling the transitions differ only in the prefix string 0 or 1, but that this change impacts the domain of the active substitutions. Therefore, when defining any notion of equivalence using this operational semantics, we must keep track of a substitution between aliases (which should be a bijection), thereby allowing for differences in prefixes and making the particular choice of alias irrelevant when performing equivalence checking.

Definition 5 (alias substitution)**.** *Alias substitutions ρ extend to labels such that $(MN)\rho = M\rho N\rho$ and $(M(\alpha))\rho = M\rho(\alpha\rho)$, and $\tau\rho = \tau$.*

The following function is just a convenience to pick out the domain of an active substitution. This is useful since the domain remembers the set of aliases that have already been extruded.

Definition 6. *We extend the domain function to extended processes such that* $\mathrm{dom}(\nu\vec{x}.(\theta \mid A)) = \mathrm{dom}(\theta)$.

We make use of aliases substitution even for interleaving equivalences and similarities. For example, the following[1] defines a notion of interleaving "presimilarity" (a term coined here to distinguish it from "similarity", introduced in Def. 8) that disregards the locations but requires the aliases to be substituted.

Definition 7 (interleaving presimilarity)**.** *Let $\mathscr{R}$ be a relation between pairs of extended processes and ρ be an alias substitution. We say $\mathscr{R}$ is an i-presimulation whenever if $A\ \mathscr{R}^{\rho}\ B$, then:*

- *If $A \xrightarrow[u]{\pi} A'$ then there exists ρ', B', u', π' s.t. $\rho\restriction_{\mathrm{dom}(A)} = \rho'\restriction_{\mathrm{dom}(A)}$, $B \xrightarrow[u']{\pi'} B'$, $\pi\rho' = \pi'$ and $A'\ \mathscr{R}^{\rho'}\ B'$.*
- *If $A \vDash M = N$, then $B \vDash M\rho = N\rho$.*

We say process P i-presimulates Q, and write $P \sqsubseteq_i Q$, whenever there exists a i-presimulation $\mathscr{R}$ such that $\mathrm{id} \mid P\ \mathscr{R}^{\mathrm{id}}\ \mathrm{id} \mid Q$.

Notice that i-presimilarity $\sqsubseteq_i$ is defined on processes: defining it on extended processes A and B require bijective alias substitutions ρ such that $\mathrm{dom}(A)\rho = \mathrm{dom}(B)$ that complicate later definitions.

Now consider again the processes examined above $\nu x.\left(\overline{a}\langle x\rangle \mid \overline{b}\langle h(x)\rangle\right)$ and $\nu x.\left(\overline{b}\langle h(x)\rangle \mid \overline{a}\langle x\rangle\right)$. They are mutually i-presimilar, i.e., there exist two i-presimulations that relate them in each direction. These

[1] We color what we want to stress or the "diff" with the previous definition or a definition indicated in footnote.

presimulations involve building up a bijection on aliases ρ such that $\rho : 0\lambda \mapsto 1\lambda$ and $\rho : 1\lambda \mapsto 0\lambda$. By applying this bijection to the labels of each of the transitions presented above, indeed the actions of both processes, $\overline{a}(0\lambda)$ and $\overline{a}(1\lambda)$ map to each other. Observe also that the final states these processes reach are $A = \nu x.\left(\{^{x}/_{0\lambda}\} \circ \{^{h(x)}/_{1\lambda}\} \mid 0 \mid 0\right)$ and $B = \nu x.\left(\{^{x}/_{1\lambda}\} \circ \{^{h(x)}/_{0\lambda}\} \mid 0 \mid 0\right)$. Since $A \vDash h(0\lambda) = 1\lambda$, we also want this test to be satisfied by B, modulo the alias substitution ρ that has been built by the presimilarity, i.e., $B \vDash (h(0\lambda))\rho = (1\lambda)\rho$, which indeed holds. Notice that it is necessary to apply ρ to the messages when checking that equality tests are preserved, and that it must be applied before the active substitution.

One may ask whether it is possible to simply have a permutation of location prefixes, keeping alias variables the same. Such an approach would not be sufficiently flexible to capture relations such as

$$\nu x.\left(\overline{b}\langle h(x)\rangle.\overline{a}\langle x\rangle\right) \sqsubseteq_i \nu x.\left(\overline{b}\langle h(x)\rangle \mid \overline{a}\langle x\rangle\right) \qquad \text{and} \qquad \nu x.(\overline{a}\langle x\rangle \mid \overline{x}\langle h(x)\rangle) \sqsubseteq_i \nu x.(\overline{a}\langle x\rangle.\overline{x}\langle h(x)\rangle).$$

In both examples, on one side there are two locations, and on the other there is only one location. This helps explain why we employ a bijection between aliases and not only between locations.

The above definition is an aesthetic preorder in that we always match a positive test on the left with a positive test on right. The clause concerning equality tests effectively defines "static implication" proposed in related work on applied process calculi [25]. However, there is a small gap compared to the standard simulation we expect for the π-calculus. Indeed, the definition of presimilarity lets the following hold:

$$\nu y.(\overline{a}\langle x\rangle + \overline{a}\langle y\rangle) \sqsubseteq_i \overline{a}\langle x\rangle$$

Therefore the above processes are mutually presimilar, since the other direction holds trivially. The reason the above relation holds is that there is no equality that can distinguish the message x from the private name y. That is, both

$$\mathrm{id} \mid \nu y.(\overline{a}\langle x\rangle + \overline{a}\langle y\rangle) \xrightarrow[[0]]{\overline{a}(\lambda)} \nu y.(\{^{x}/_{\lambda}\} \mid 0) \qquad \text{and} \qquad \mathrm{id} \mid \nu y.(\overline{a}\langle x\rangle + \overline{a}\langle y\rangle) \xrightarrow[[1]]{\overline{a}(\lambda)} \nu y.(\{^{y}/_{\lambda}\} \mid 0)$$

can only be matched by $\mathrm{id} \mid \overline{a}\langle x\rangle \xrightarrow[[]]{\overline{a}(\lambda)} \{^{x}/_{\lambda}\} \mid 0$, and there is no M and N such that $\nu y.(\{^{y}/_{\lambda}\} \mid 0) \vDash M = N$ and $\{^{x}/_{\lambda}\} \mid 0 \nvDash M = N$. Notice this is despite the fact that $\{^{x}/_{\lambda}\} \mid 0 \vDash \lambda = x$, but $\nu y.(\{^{y}/_{\lambda}\} \mid 0) \nvDash \lambda = x$, which would amount to $\nu y.(\{^{y}/_{\lambda}\} \mid 0)$ satisfying the inequality $\lambda \neq x$; hence such negative distinguishing tests are not picked up on by presimilarity.

Intuitively, one can think of the above example modelling, with the left process, an "unreliable" channel (i.e., output on channel $\overline{a}$ can either be the intended message x or anything else as y); whereas the right process is a reliable channel where the receiver would always get the intended message x. Since we expect that in a conservative extension of the π-calculus the above processes can be distinguished, we strengthen presimilarity to obtain "similarity". This strengthening amounts to demanding static equivalence, even when considering similarity preorders.

Definition 8 (interleaving similarity)**.** *Let $\mathscr{R}$ be a relation between pairs of extended processes and ρ be an alias substitution. We say $\mathscr{R}$ is a i-simulation whenever if $A\ \mathscr{R}^{\rho}\ B$, then:*

- *If $A \xrightarrow[u]{\pi} A'$ then there exists ρ', B', u', π' s.t. $\rho\restriction_{\mathrm{dom}(A)} = \rho'\restriction_{\mathrm{dom}(A)}$, $B \xrightarrow[u']{\pi'} B'$, $\pi\rho' = \pi'$ and $A'\ \mathscr{R}^{\rho'}\ B'$.*
- *$A \vDash M = N$ iff $B \vDash M\rho = N\rho$.*

We say process P i-simulates Q, and write $P \preceq_i Q$, whenever there exists an i-simulation $\mathscr{R}$ such that $\mathrm{id} \mid P\ \mathscr{R}^{\mathrm{id}}\ \mathrm{id} \mid Q$. If in addition the relation is symmetric, e.g., $A\ \mathscr{R}^{\rho}\ B$ iff $B\ \mathscr{R}^{\rho^{-1}}\ A$, then P and Q are i-bisimilar, written $P \sim_i Q$.

The notions of bisimilarity obtained from presimilarity and similarity concide, hence we see similarity as presimilarity with a little of the power of bisimilarity for equating tests. Note that $\nu y.(\overline{a}\langle x\rangle+\overline{a}\langle y\rangle)$ and $\overline{a}\langle x\rangle$ are not i-similar, since there is a $\overline{a}(\lambda)$-transition after which only the right side satisfies $\lambda = x$.

Definitions in related work on the applied π-calculus do not require an alias substitution, as in the definition above. Those papers [1, 21] allow the alias to be freely chosen, without indicating the location. Notice also the location under the labelled transition is never used in these interleaving semantics. The located aliases and location labels are however important for our non-interleaving equivalences, and for concurrency diamonds required to extend techniques such as POR to the full applied π-calculus.

4 Using LATS to define semantics preserving duration or causality

We now make the transition from interleaving to non-intereaving semantics. The border between interleaving and non-interleaving semantics was heavily debated in the early 1990's. A common argument at the time was that problems concerning non-interleaving semantics could be reduced to a problem in terms of an interleaving semantics, since processes such as $\nu x.(\overline{a}\langle x\rangle \mid \overline{a}\langle x\rangle)$ and $\nu x.(\overline{a}\langle x\rangle.\overline{a}\langle x\rangle)$ could be distinguished by splitting each output actions into a "begin output" and "end output" action and then considering the interleavings. This view was eventually dispelled by van Glabbeek and Vaandrager [18] (based on works, such as [3, 19, 32]), who showed that, no matter how many times actions are split, one cannot obtain an interleaving semantics that preserves desirable properties of a non-interleaving semantics.

Their key example, translated here to the π-calculus, is that there is an interleaving simulation relating the following processes.

$$\nu c,d.\big(\big(\overline{d}\langle d\rangle|\ \nu n.\overline{a}\langle n\rangle.d(z).n(x)\big) \mid (\overline{c}\langle c\rangle|\ c(y))\big) \preceq_i \nu c,d.\big(\big(\overline{d}\langle d\rangle|\ \nu n.\overline{a}\langle n\rangle.d(z)\big) \mid (\overline{c}\langle c\rangle|\ c(y).n(x))\big) \quad (1)$$

Furthermore, even if we were to enhance similarity with the power to split actions, these processes would still be related. What is happening here is that when a τ-transition both starts and terminates while another τ-transition is running, the end of the longer and shorter τ-transition can be swapped, resulting in a behaviour that can be simulated on the right. Such "swapping" semantics were investigated by Vogler [32], when investigating the coarsest language theory robust against splitting.

Although the above example preserves event splitting, allowing it to hold can be considered problematic since we confuse the beginning and end of two distinct events that happen to be labelled in the same way. A notion of similarity allowing the above example to hold, neither preserves the duration of events, nor the causal dependencies between events. To see why, observe that the process on the left above has a τ-transition that can start before any other event and terminate after all events have finished, but there is no τ-transition on the right that can match that timing history. In this section, we lift two truly non-interleaving semantics (ST and HP) to the applied π-calculus that do preserve such properties.

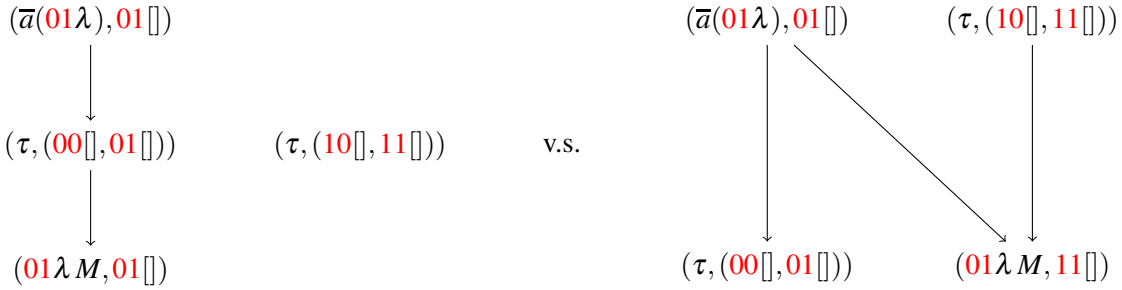
4.1 Independence and permutations of events

To define non-interleaving equivalences we make use of independence relations. Structural independence, that looks only at the locations, is sufficient for calculi such as CCS. However, for the π-calculus and its extensions, in addition, so called *link causality* should be accounted for to determine whether an output must occur first before a subsequent event occurs.

Definition 9 (independence)**.** *Define $\mathscr{L}oc$ a function on location labels (Def. 3) such that $\mathscr{L}oc(\ell) = \{\ell\}$ and $\mathscr{L}oc(\ell_0,\ell_1) = \{\ell_0,\ell_1\}$. The structural independence relation I_ℓ on location labels is the least relation*

defined by $u_0\ I_\ell\ u_1$ *whenever for all locations* $\ell_0 \in \mathscr{L}oc(u_0)$ *and* $\ell_1 \in \mathscr{L}oc(u_1)$*, there exist a string* $s \in \{0,1\}^*$ *and locations* ℓ_0', ℓ_1'*, such that either:* $\ell_0 = s0\ell_0'$ *and* $\ell_1 = s1\ell_1'$*; or* $\ell_0 = s1\ell_0'$ *and* $\ell_1 = s0\ell_1'$*. Events* (π, u) *are pairs of action labels* π *and location labels* u*. The independence relation* $\frown$ *on events is the least symmetric relation such that* $(\pi_0, u_0) \frown (\pi_1, u_1)$ *whenever* $u_0\ I_\ell\ u_1$ *and if* $\pi_0 = \overline{M}(\alpha)$*, then* $\alpha \# \pi_1$*.*

Consider again Eq. 1, where we present its executions as a graph where the events are nodes and edges represent dependencies (i.e., the absence of independence). Note M is any message such that $\mathrm{fa}(M) \subseteq \{01\lambda\}$, and results from an input.



On the left above, observe that the rightmost τ-transition is independent from all other transitions, while all other events in that diagram are dependent on each other. In contrast, on the right above, both τ-transitions are dependent on only one other event, and independent of the others. In what follows, we make precise what it means for the processes producing these events to be incomparable.

4.2 ST-similarity and ST-bisimilarity, preserving duration

We define now ST semantics that preserve the duration of events, abstractly, without explicit time, by providing mechanisms for modelling the start and termination of events. To avoid confusion about which event terminates at a particular moment, definitions of ST equivalences make use of a device to pair events that started at the same moment, which is done by a relation over events in this work. We define some simple auxiliary functions to work with relations and sets of events.

Definition 10 (auxiliary functions)**.** *Given a relation over events* S*, we write* $\mathrm{dom}(\mathsf{S})$ *and* $\mathrm{ran}(\mathsf{S})$ *the sets of events forming the domain and range of* S*, respectively. Given an event* e *and set of events* E *we write* $e \frown E$ *whenever for all* $e' \in E$ *we have* $e \frown e'$*.*

Our definition of ST-similarity below enhances the definition of interleaving similarity such that we not only preserve the transitions, but also respect the fact that some events may have started already and are running concurrently with the new event. This is captured by ensuring that we only consider a transition labelled with event (π, u) if the condition $(\pi, u) \frown \mathrm{dom}(\mathsf{S})$ holds, which ensures that all events currently running in S are independent of (π, u). We then demand that the corresponding transition, labelled with (π', u'), is also independent of all events currently started, which is ensured by the condition $(\pi', u') \frown \mathrm{ran}(\mathsf{S})$. Notice that the relation on events strongly associate (π, u) and (π', u'), and thus, when we appeal to the second clause below they will be removed from the relation simultaneously.[2] This models the termination of the events. Thus we only record in relation S those events that are concurrently running now, which is suited to our independence relation that is only well-defined on transitions enabled in the same state or subsequent states.

Definition 11 (ST-similarity)**.** *Let* $\mathscr{R}$ *be a relation between pairs of extended processes,* ρ *be an alias substitution, and* S *be a relation over events. We say* $\mathscr{R}$ *is an ST-simulation whenever if* $A\ \mathscr{R}^{\rho,\mathsf{S}}\ B$*, then:*

[2] Using a relation has the same effect as employing a bijection between the labels of events in other formulations of ST-bisimilarity [15, p. 14].

- *If* $A \xrightarrow[u]{\pi} A'$ *and* $(\pi,u) \smile \mathrm{dom}(\mathsf{S})$ *then there exists* ρ', B', u', *and* π' *s.t.* $\rho\restriction_{\mathrm{dom}(A)} = \rho'\restriction_{\mathrm{dom}(A)}$, $B \xrightarrow[u']{\pi'} B'$, $\pi\rho' = \pi'$, $(\pi',u') \smile \mathrm{ran}(\mathsf{S})$, *and* $A' \mathscr{R}^{\rho',\mathsf{S}\cup\{((\pi,u),(\pi',u'))\}} B'$.
- *If* $\mathsf{S}'\subseteq\mathsf{S}$ *then* $A \mathscr{R}^{\rho,\mathsf{S}'} B$.
- $A \vDash M = N$ *iff* $B \vDash M\rho = N\rho$.

We say process P ST-simulates Q, and write $P \preceq_{ST} Q$, whenever there exists a ST-simulation $\mathscr{R}$ s.t. $\mathrm{id} \mid P \mathscr{R}^{\mathrm{id},\emptyset} \mathrm{id} \mid Q$. *If in addition $\mathscr{R}$ is symmetric, e.g., $A \mathscr{R}^{\rho,\mathsf{S}} B$ iff $B \mathscr{R}^{\rho^{-1},\mathsf{S}^{-1}} A$, then P and Q are ST-bisimilar, written $P \sim_{ST} Q$.*

Consider the following, which are i-bisimilar, but can be distinguished by ST-similarity.

$$\nu x.\overline{a}\langle x\rangle \mid \nu x.\overline{a}\langle x\rangle \not\preceq_{ST} \nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle$$

To see why the above does not hold, observe that two events can be concurrently started on the left, but the second cannot be matched on the right. That is, when playing the ST-simulation game, we reach the following states, where $\rho : 0\lambda \mapsto \lambda'$ and $(\overline{a}(0\lambda), 0[]) \ \mathsf{S}\ (\overline{a}(\lambda'), [])$.

$$\nu y.(\{^{y}/_{0\lambda}\} \mid 0 \mid \nu x.\overline{a}\langle x\rangle) \ R^{\rho,\mathsf{S}} \ \nu y.(\{^{y}/_{\lambda'}\} \mid \nu x.\overline{a}\langle x\rangle)$$

Now observe that the extended process on the left can perform an event $(\overline{a}(1\lambda), 1[])$ independent of $\mathrm{dom}(\mathsf{S})$, but the process on the right cannot perform any action independent of $\mathrm{ran}(\mathsf{S})$. From this we conclude that the above processes cannot be related by any ST-simulation.

We still however obtain many relations that also hold according to interleaving semantics. For example, observe that the following holds.

$$\nu x,y,z.(\overline{a}\langle x\rangle.(\overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle)) \preceq_{ST} \nu x,y,z.(\overline{a}\langle x\rangle.\overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle). \tag{2}$$

Indeed, the left term's only transition

$$\mathrm{id} \mid \nu x,y,z.(\overline{a}\langle x\rangle.(\overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle)) \xrightarrow[[]]{\overline{a}(\lambda)} \nu x,y,z.(\{^{x}/_{\lambda}\} \mid \overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle)$$

can easily be matched by the right term

$$\mathrm{id} \mid \nu x,y,z.(\overline{a}\langle x\rangle.\overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle) \xrightarrow[0[]]{\overline{a}(0\lambda)} \nu x,y,z.(\{^{x}/_{0\lambda}\} \mid \overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle)$$

and $\rho : \lambda \mapsto 0\lambda$, $\mathsf{S} = \{((\overline{a}(\lambda), []), (\overline{a}(0\lambda), 0[]))\}$ satisfies our definition. Then, one needs to show that the resulting two terms are in $\mathscr{R}^{\rho',\mathsf{S}}$ *and* $\mathscr{R}^{\rho',\emptyset}$. For $\mathscr{R}^{\rho',\mathsf{S}}$, since $\nu x,y,z.(\{^{x}/_{\lambda}\} \mid \overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle)$'s only transitions (with events $(\overline{b}(0\lambda'), 0[])$ and $(\overline{c}(1\lambda'), 1[])$) are *not* independent with $\mathrm{dom}(S) = (\overline{a}(\lambda), [])$, they do not need to be matched by $\nu x,y,z.(\{^{x}/_{0\lambda}\} \mid \overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle)$. For $\mathscr{R}^{\rho',\emptyset}$, it is straightforward to pair $(\overline{b}(0\lambda'), 0[])$ and $(\overline{c}(1\lambda'), 1[])$ with themselves, and to map $0\lambda'$ and $1\lambda'$ to themselves.

Interestingly, two processes that are unrelated by ST-similarity can be in the limit identified even by ST-bisimilarity. Consider for example the following.

$$\nu x.\overline{a}\langle x\rangle \mid \nu x.\overline{a}\langle x\rangle \not\preceq_{ST} \nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle \quad \text{and yet} \quad !\nu x.\overline{a}\langle x\rangle \sim_{ST} !(\nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle)$$

To establish the equation on the right above, we construct the relation below and prove that it is an ST-bisimulation by checking that each condition holds. Firstly, S is downward closed, since it is not required to be defined for all $i \in \phi \cup \psi$. When the right side leads, it can either start an action in a component that has not fired (in L or greater than n), or it can start a second component that is not blocked (i.e., in ϕ, such

that $(\overline{a}(1^i0\lambda), 1^i0[]) \notin \mathrm{ran}(\mathsf{S})$), either of which can be matched on the left by starting a new independent component. When the left side leads it can only fire a new component, which can be matched by starting a new component on the right. Those transitions are preserved by $\mathscr{R}^{\rho,\mathsf{S}}$; notably, there can never be more concurrently started actions on the left than there are started components on the right. Let $\mathscr{R}$ be the least symmetric relation containing the following (upto $\equiv$).

$$\nu\vec{z}.(\theta \mid Q_0 \mid \ldots (Q_m \mid !\nu x.\overline{a}\langle x\rangle)\ldots) \;\mathscr{R}^{\rho,\mathsf{S}}\; \nu\vec{y}.(\sigma \mid P_0 \mid \ldots (P_n \mid !\nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle)\ldots)$$

$\{0,\ldots m\} = \chi \cup J$ with χ and J disjoint and $m \notin J$

$$Q_i = \begin{cases} 0 & \text{if } i \in \chi \\ \nu x.\overline{a}\langle x\rangle & \text{if } i \in J \end{cases} \qquad \vec{z}_i = \begin{cases} z_i & \text{if } i \in \chi \\ \varepsilon & \text{if } i \in J \end{cases} \qquad \vec{z} = \textstyle\bigcup_{i=0}^{m} \vec{z}_i$$

$$\theta_i = \begin{cases} \{^{z_i}/_{1^i0\lambda}\} & \text{if } i \in \chi \\ \mathrm{id} & \text{if } i \in J \end{cases} \qquad \theta = \textstyle\prod_{i=0}^{m} \theta_i$$

$\{0,\ldots n\} = \psi \cup \phi \cup L$ with ψ, ϕ and L disjoint and $n \notin L$

$$P_i = \begin{cases} 0 & \text{if } i \in \psi \\ \nu x.\overline{a}\langle x\rangle & \text{if } i \in \phi \\ \nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle & \text{if } i \in L \end{cases} \qquad \vec{y}_i = \begin{cases} x_i, y_i & \text{if } i \in \psi \\ x_i & \text{if } i \in \phi \\ \varepsilon & \text{if } i \in L \end{cases} \qquad \vec{y} = \textstyle\bigcup_{i=0}^{n} \vec{y}_i$$

$$\sigma_i = \begin{cases} \{^{x_i}/_{1^i0\lambda}\} \circ \{^{y_i}/_{1^i0\lambda'}\} & \text{if } i \in \psi \\ \{^{x_i}/_{1^i0\lambda}\} & \text{if } i \in \phi \\ \mathrm{id} & \text{if } i \in L \end{cases} \qquad \sigma = \textstyle\prod_{i=0}^{n} \sigma_i$$

with $\rho : \mathrm{dom}(\theta) \to \mathrm{dom}(\sigma)$ any bijection such that $(1^{f(i)}0\lambda)\rho = \begin{cases} 1^i0\lambda & \text{if } i \in \phi \\ 1^i0\lambda' & \text{if } i \in \psi \end{cases}$, for

$f\colon \phi \cup \psi \to \chi$ any injection and $\begin{cases} (\overline{a}(1^{f(i)}0\lambda), 1^{f(i)}0[]) \;\mathsf{S}\; (\overline{a}(1^i0\lambda), 1^i0[]) & \text{only if } i \in \phi \\ (\overline{a}(1^{f(i)}0\lambda), 1^{f(i)}0[]) \;\mathsf{S}\; (\overline{a}(1^i0\lambda'), 1^i0[]) & \text{only if } i \in \psi \end{cases}$

4.3 History-Preserving similarity: preserving causality

Besides observing the duration of events as in ST semantics, History-Preserving semantics observe also the partial order of causal dependencies between events. We define here HP-similarity as a strengthening of our definition of ST-similarity such that we observe not only independence but also dependence, thereby, step-by-step, ensuring that exactly the same dependencies are satisfied by the events produced by both processes. Technically this is achieved in the definition below, by partitioning the relation representing concurrently started events S according to the firing event (π,u) into: S_1 consisting of events that are independent of the current event (i.e., $(\pi,u) \frown \mathrm{dom}(\mathsf{S}_1)$); S_2 consisting of those events that are not independent (i.e., $(\pi,u) \not\frown \mathrm{dom}(\mathsf{S}_2)$). Thus S_2 is the minimal set of events that must have terminated before the new event can proceed. This partitioning must be reflected by the matching transition on the right, thereby preserving both independence and dependence. Since only the independent events and the new event are retained at the next step, the relation over events always consists of independent events.

Definition 12 (HP-similarity[3])**.** *Let $\mathscr{R}$ be a relation between pairs of extended processes, ρ be an alias substitution, and S be a relation over events. We say $\mathscr{R}$ is an HP-simulation whenever if $A\ \mathscr{R}^{\rho,\mathsf{S}}\ B$, then:*

- *If $A \xrightarrow[u]{\pi} A'$, $\mathsf{S}_1 \cup \mathsf{S}_2 = \mathsf{S}$, $(\pi,u) \frown \mathrm{dom}(\mathsf{S}_1)$ and $(\pi,u) \not\frown \mathrm{dom}(\mathsf{S}_2)$, then there exists ρ', B', u', and π' s.t. $\rho\restriction_{\mathrm{dom}(A)} = \rho'\restriction_{\mathrm{dom}(A)}$, $B \xrightarrow[u']{\pi'} B'$, $\pi\rho' = \pi'$, $(\pi',u') \frown \mathrm{ran}(\mathsf{S}_1)$, $(\pi',u') \not\frown \mathrm{ran}(\mathsf{S}_2)$, and $A'\ \mathscr{R}^{\rho',\mathsf{S}_1\cup\{((\pi,u),(\pi',u'))\}}\ B'$.*

[3] This definition is "diffed" against Def. 11. The clause "If $\mathsf{S}' \subseteq \mathsf{S}$ then $A\ \mathscr{R}^{\rho,\mathsf{S}'}\ B$." was replaced by the partitioning of events.

- $A \vDash M = N$ *iff* $B \vDash M\rho = N\rho$.

We say process P is HP-simulated by Q, and write $P \preceq_{HP} Q$, whenever there exists an HP-simulation $\mathscr{R}$ s.t. $\mathrm{id} \mid P \; \mathscr{R}^{\mathrm{id},\emptyset} \; \mathrm{id} \mid Q$. *If in addition $\mathscr{R}$ is symmetric, then P and Q are HP-bisimilar, written $P \sim_{HP} Q$.*

When we consider similarity the difference between ST-similarity and HP-similarity is clear. For example, although Eq. 2 proved the ST-similarity of the following, they are not HP-similar.

$$\nu x,y,z.(\overline{a}\langle x\rangle.(\overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle)) \not\preceq_{HP} \nu x,y,z.(\overline{a}\langle x\rangle.\overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle)$$

To see this, observe that when attempting to construct an HP-simulation we can reach the following pair of processes, where $\rho : \lambda \mapsto 0\lambda$ and $(\overline{a}(\lambda),[]) \; \mathsf{S} \; (\overline{a}(0\lambda),0[])$.

$$\nu x,y,z.\left(\{^{x}/_{\lambda}\} \mid \overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle\right) \; \mathscr{R}^{\rho,\mathsf{S}} \; \nu x,y,z.\left(\{^{x}/_{0\lambda}\} \mid \overline{b}\langle y\rangle \mid \overline{c}\langle z\rangle\right)$$

At this moment, the left side can perform a transition on channel c that is *dependent* on $(\overline{a}(\lambda),[])$ in $\mathrm{dom}(\mathsf{S})$. Yet, although the right side can perform a transition on channel c, it cannot match the dependency, since $(\overline{c}(1\lambda),1[])$ and $(\overline{a}(0\lambda),0[])$ are independent.

When we consider bisimilarity, the gap is more subtle for finite processes. An example separating ST-bisimilarity from HP-bisimilarity is the following.

$$\nu a,b.\left((\overline{a}\langle a\rangle \mid (a(x)+b(x))) \mid \overline{c}\langle c\rangle.\overline{b}\langle b\rangle\right) \sim_{ST} \nu a.((\overline{a}\langle a\rangle \mid a(x)) \mid \overline{c}\langle c\rangle) \tag{3}$$

To see that they are unrelated by HP-similarity (hence certainly unrelated by HP-bisimilarity), observe that the two processes can perform the following transitions

$$\mathrm{id} \mid \nu a,b.\left((\overline{a}\langle a\rangle \mid (a(x)+b(x))) \mid \overline{c}\langle c\rangle.\overline{b}\langle b\rangle\right) \xrightarrow[1[]]{\overline{c}(1\lambda)} \nu a,b.\left(\{^{c}/_{1\lambda}\} \mid (\overline{a}\langle a\rangle \mid (a(x)+b(x))) \mid \overline{b}\langle b\rangle\right)$$

and

$$\mathrm{id} \mid \nu a.((\overline{a}\langle a\rangle \mid a(x)) \mid \overline{c}\langle c\rangle) \xrightarrow[1[]]{\overline{c}(1\lambda)} \nu a.(\{^{c}/_{1\lambda}\} \mid (\overline{a}\langle a\rangle \mid a(x)) \mid 0)\,.$$

The relation on events at this moment is such that $(\overline{c}(1\lambda),1[]) \; \mathsf{S} \; (\overline{c}(1\lambda),1[])$ where alises are related by the identity function. Notice now that $\nu a,b.\left(\{^{c}/_{1\lambda}\} \mid (\overline{a}\langle a\rangle \mid (a(x)+b(x))) \mid \overline{b}\langle b\rangle\right)$ can perform a transition labelled with $(\tau,(01[1],1[]))$, which is not independent from $(\overline{c}(1\lambda),1[])$; yet, although the other process can perform a τ-transition, it cannot match the dependency constraints. In contrast, since ST-similarity would not require dependency constraints to be matched, a matching τ-transition can be performed at the corresponding point in any ST-bisimulation game.

The distinction between ST and HP is less subtle when we consider replicated processes. Consider

$$!(\nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle) \not\preceq_{HP} !(\nu x.\overline{a}\langle x\rangle) \qquad \text{and yet} \qquad !(\nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle) \sim_{ST} !(\nu x.\overline{a}\langle x\rangle).$$

The latter relation above we have already established previously, p. 12. Now we attempt to construct an HP-simulation containing the relation on the left. Observe that a possible first transition can be matched by both processes as follows.

$$\mathrm{id} \mid !(\nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle) \xrightarrow[0[]]{\overline{a}(0\lambda)} \nu y.(\{^{y}/_{0\lambda}\} \mid \nu x.\overline{a}\langle x\rangle \mid !(\nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle))$$

$$\mathrm{id} \mid !(\nu x.\overline{a}\langle x\rangle) \xrightarrow[1^{n}0[]]{\overline{a}(1^{n}0\lambda)} \nu y.(\{^{y}/_{1^{n}0\lambda}\} \mid 0 \mid (\nu x.\overline{a}\langle x\rangle \ldots (\nu x.\overline{a}\langle x\rangle \mid !\nu x.\overline{a}\langle x\rangle)))$$

At this point we have $(\overline{a}(0\lambda),0[]) \mathsf{S} (\overline{a}(1^n0\lambda),1^n0[])$ and aliases substitution such that $\rho : 0\lambda \mapsto 1^n0\lambda$. Then, $\nu y.(\{^y/_{0\lambda}\} \mid \nu x.\overline{a}\langle x\rangle \mid !(\nu x.\overline{a}\langle x\rangle.\nu x.\overline{a}\langle x\rangle))$ can perform an event $(\overline{a}(0\lambda'),0[])$ that is *not independent* of $(\overline{a}(0\lambda),0[])$, but the other process can only perform an independent transition, violating the condition of HP-similarity that the transition on the right must have the same dependencies.

Similarly, we have $!\nu x,y.\left(\overline{a}\langle x\rangle.\overline{b}\langle y\rangle + \overline{b}\langle y\rangle.\overline{a}\langle x\rangle\right) \not\sim_{HP} !\nu x.\overline{a}\langle x\rangle \mid !\nu x.\overline{b}\langle x\rangle$ which are equated by the ST similarity. We interpret these kinds of examples as follows. From the perspective of the ST-semantics, executing the processes in an interleaved manner on one server that can be duplicated is the same as executing them on two servers that can be duplicated. This is because the same duration of events can be achieved by both, and in some settings this may be the desirable effect. However, this comes at the cost of a loss of awareness in the number of servers required (seen as resources), and of a sense of partition tolerance, since the right process needs up to half as much servers as the left process requires to complete the same task. This can be problematic if an attacker has the power to partition a system, e.g., by DDoS on a connection link, thereby isolating a small number of servers from the rest. In that situation, the difference picked out by HP-similarity becomes evident, and one can notice moreover that HP-similarity behaves the same in the finite case and in the limit.

There is related work on "causal" bisimilarity for the π-calculus [5], which is strictly finer than HP-bisimilarity. This is because causal bisimilarity only accounts for structural causality and not for link causality. Thus, for example although $\nu n.(\overline{a}\langle n\rangle \mid n(x)) \sim_{HP} \nu n.(\overline{a}\langle n\rangle.n(x))$ holds, these processes are distinguished by causal bisimilarity, because "there is both a subject and an object dependency between the actions [in the former], whereas in [the latter] there is only an object dependency" [5, p. 387].

4.4 Discussion on ST and HP in the context of privacy

We now revisit the essence of a privacy problem in the literature [14, 21]. The following compares two systems containing a process ready to respond to a message sent using a one-time key k, i.e., there is only one input action capable of responding to that key. The left process allows processes in distinct locations to send a message using k, while on the right there is only one location with that capability. Letting $P_{\text{ok}} \triangleq b(x).[\mathtt{snd}(\mathtt{dec}(x,k)) = \text{hi}]\overline{a}\langle\{\text{ok}\}_k\rangle$, we have :

$$\nu k.\Big((\nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle \mid (\nu m.\overline{a}\langle m\rangle + \nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle)) \mid P_{\text{ok}}\Big) \not\simeq_i \nu k.\Big((\nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle \mid \nu m.\overline{a}\langle m\rangle) \mid P_{\text{ok}}\Big)$$

The above processes are trace equivalent, yet these processes are distinguished by interleaving similarity as indicated above. Note that we assume a standard symmetic key Dolev-Yao equational theory E such that $\mathtt{dec}(\{M\}_K,K) =_E M$, $\mathtt{fst}(\langle M,N\rangle) =_E M$ and $\mathtt{snd}(\langle M,N\rangle) =_E N$.

Now compare this example above with the example below, where we essentially replicate some of the processes, and notice that, by doing so, these processes become i-bisimilar—they are even ST-bisimilar.

$$\nu k.\Big((!\nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle \mid !\nu m.\overline{a}\langle m\rangle) \mid P_{\text{ok}}\Big) \underset{\not\simeq_{HP}}{\overset{\sim_{ST}}{}} \nu k.\Big((\nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle \mid !\nu m.\overline{a}\langle m\rangle) \mid P_{\text{ok}}\Big) \tag{4}$$

The problem is that there is no way for an observer to tell the difference between the output on channel a after the match and a parallel random output on channel a (in the finite case all such parallel actions can be used up before performing the input, so it becomes clear whether or not $\{\text{ok}\}_k$ is triggered, even without the attacker being able to read the message). Of course, creating a channel for each process can be a solution to this modelling problem [21]. But the question we ask here is different: *is the difference in locations picked up only by non-interleaving semantics?*.

The fact that the processes in Eq. 4 are ST-bisimilar shows that observing differences in the duration of events does not affect the problem. Indeed, while the output $\{\text{ok}\}_k$ can only occur after the input,

there is always another parallel action indistinguishable to the attacker ready to fire for the same duration. Therefore ST-bisimilarity is not distinguishing sufficiently the localities for this problem.

In contrast to the above, HP-similarity can detect the difference in localities. This is because $\{\text{ok}\}_k$ is triggered after the input, and HP-similarity ensures that the same dependencies are preserved on the right hand side of the simulation.

This problem is encapsulated by the following ST-bisimilar, but not mutually HP-similar, processes:

$$!\nu n.\overline{a}\langle n\rangle \mid b(x).\nu n.\overline{a}\langle h(n)\rangle \overset{\sim_{ST}}{\not\sim_{HP}} !\nu n.\overline{a}\langle n\rangle \mid b(x)$$

Hence, HP semantics is better at preserving structure, since we know that there is a success message (represented by $\{\text{ok}\}_k$ here) caused by the input action, while ST semantics confuses this with other indistinguishable messages on channel a.

4.5 Failure semantics

Considering simulations, not only bisimulation, allows to explore more of the linear-time/branching-time spectrum. For example, we can define ST failure similarity [2], which extends ST-similarity such that if an action is enabled by the process on the right, then it should be enabled on the left.

Definition 13 (STf-similarity[4])**.** *Let $\mathscr{R}$ be a relation between pairs of extended processes, ρ be an alias substitution, and S be a relation over events. We say $\mathscr{R}$ is an STf-simulation whenever if $A\ \mathscr{R}^{\rho,\mathsf{S}}\ B$, then:*

- *If $A \xrightarrow[u]{\pi} A'$ and $(\pi,u) \smile \text{dom}(\mathsf{S})$ then there exists ρ', B', u', and π' s.t. $\rho\restriction_{\text{dom}(A)} = \rho'\restriction_{\text{dom}(A)}$, $B \xrightarrow[u']{\pi'} B'$, $\pi\rho' = \pi'$, $(\pi',u') \smile \text{ran}(\mathsf{S})$, and $A'\ \mathscr{R}^{\rho',\mathsf{S}\cup\{((\pi,u),(\pi',u'))\}}\ B'$.*
- *If $B \xrightarrow[u']{\pi'} B'$ and $(\pi',u') \smile \text{ran}(\mathsf{S})$ then there exists ρ', A', u and π s.t. $\rho\restriction_{\text{dom}(A)} = \rho'\restriction_{\text{dom}(A)}$, $A \xrightarrow[u]{\pi} A'$, $\pi\rho' = \pi'$, and $(\pi,u) \smile \text{dom}(\mathsf{S})$.*
- *If $\mathsf{S}'\subseteq\mathsf{S}$ then $A\ \mathscr{R}^{\rho,\mathsf{S}'}\ B$.*
- *$A \vDash M = N$ iff $B \vDash M\rho = N\rho$.*

We say process P is STf-simulated by Q, and write $P \preceq_{STf} Q$, whenever there exists an STf-simulation $\mathscr{R}$ such that $\text{id} \mid P\ \mathscr{R}^{\text{id},\emptyset}\ \text{id} \mid Q$.

Tantalisingly, the above definition appears to preserve more dependencies than ST-similarity. Not only can we detect differences in the branching structure, as expected for interleaving failure similarity, but we can also detect the differences in the independence structure. For instance we have:

$$\nu x,y.(\overline{a}\langle x\rangle.\overline{a}\langle y\rangle) \not\preceq_{STf} \nu x,y.(\overline{a}\langle x\rangle \mid \overline{a}\langle y\rangle)$$

The distinguishing strategy is as follows. Both processes are free to perform the first output on a to reach the following indexed pair.

$$\rho : \lambda \mapsto 0\lambda \quad (\overline{a}(\lambda),[])\ \mathsf{S}\ (\overline{a}(0\lambda),0[]), \quad \nu x,y.(\{^x/_\lambda\} \mid \overline{a}\langle y\rangle)\ \mathscr{R}^{\text{id},\mathsf{S}}\ \nu x,y.(\{^x/_{0\lambda}\} \mid 0 \mid \overline{a}\langle y\rangle)$$

At this moment, the right hand side can perform a transition labelled with event $(\overline{a}(1\lambda),1[])$, since that event is independent of $(\overline{a}(0\lambda),0[])$; yet the process on the left cannot match this event. Stated

[4]This definition is "diffed" against Def. 11.

otherwise, the process on the left fails to perform the next output on a while the other output on a is still being performed, but the process on the right can. This represents a failure measurable by observing the concurrency of events. Also, $\nu x,y,z.(\overline{a}\langle x\rangle.(\overline{b}\langle y\rangle|\,\overline{c}\langle z\rangle)) \not\preceq_{STf} \nu x,y,z.(\overline{a}\langle x\rangle.\overline{b}\langle y\rangle|\,\overline{c}\langle z\rangle)$ since an action on channel c is not enabled on the left initially.

Observing failures however does not allow us to distinguish the processes in Eq. 3 nor in Eq. 4, since they are ST-bisimilar, hence mutually STf-similar.

We now adapt our privacy-inspired example of Sect. 4.4 to show the power of failure similarity. The following are mutually ST-similar (and failure interleaving trace equivalent, which we do not define here), yet they are distinguished by STf-similarity. Letting $P_{\text{er}} \triangleq b(x).[\texttt{snd}(\texttt{dec}(x,k)) \neq \text{hi}]\overline{a}\langle\{\text{er}\}_k\rangle$:

$$\nu k.\Big((\nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle|\,(\nu m.\overline{a}\langle s\rangle + \nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle))\mid P_{\text{er}}\Big) \underset{\not\preceq_{STf}}{\preceq_{ST}} \nu k.\Big((\nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle|\,\nu m.\overline{a}\langle m\rangle)\mid P_{\text{er}}\Big)$$

The difference compared to the example of Sect. 4.4 is that we can detect whether the outputs from the two locations are the same by *not seeing an error* (er) after the input. This kind of negative testing is part of the vocabulary of failure semantics. However, similarly to Eq. 4, if we include replication then the processes become ST-bisimilar, and hence cannot be distinguished by STf-similarity.

$$\nu k.\Big((!\nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle|\,!\nu m.\overline{a}\langle m\rangle)\mid P_{\text{er}}\Big) \underset{\not\preceq_{HPf}}{\sim_{ST}} \nu k.\Big((\nu r.\overline{a}\langle\{r,\text{hi}\}_k\rangle|\,!\nu m.\overline{a}\langle m\rangle)\mid P_{\text{er}}\Big) \tag{5}$$

Despite the above processes being mutually STf-similar, they are distinguished using HPf-similarity:

Definition 14 (HPf-similarity[5])**.** *Let $\mathscr{R}$ be a relation between pairs of extended processes, ρ be an alias substitution, and S be a relation over events. We say $\mathscr{R}$ is an HPf-simulation whenever if $A\ \mathscr{R}^{\rho,\mathsf{S}}\ B$, then:*

- *If $A \xrightarrow[u]{\pi} A'$, $\mathsf{S}_1 \cup \mathsf{S}_2{=}\mathsf{S}$, $(\pi,u) \smile \text{dom}(\mathsf{S}_1)$ and $(\pi,u) \not\smile \text{dom}(\mathsf{S}_2)$ then there exists ρ', B', u', and π' s.t. $\rho\restriction_{\text{dom}(A)} = \rho'\restriction_{\text{dom}(A)}$ $B \xrightarrow[u']{\pi'} B'$, $\pi\rho' = \pi'$, $(\pi',u') \smile \text{ran}(\mathsf{S}_1)$, $(\pi',u') \not\smile \text{ran}(\mathsf{S}_2)$, and $A'\ \mathscr{R}^{\rho',\mathsf{S}_1\cup\{((\pi,u),(\pi',u'))\}}\ B'$.*
- *If $B \xrightarrow[u']{\pi'} B'$, $\mathsf{S}_1 \cup \mathsf{S}_2{=}\mathsf{S}$, $(\pi',u') \smile \text{ran}(\mathsf{S}_1)$ and $(\pi',u') \not\smile \text{ran}(\mathsf{S}_2)$ then there exists ρ', A', u and π s.t. $\rho\restriction_{\text{dom}(A)} = \rho'\restriction_{\text{dom}(A)}$ $A \xrightarrow[u]{\pi} A'$, $\pi\rho' = \pi'$, $(\pi,u) \smile \text{dom}(\mathsf{S}_1)$, and $(\pi,u) \not\smile \text{dom}(\mathsf{S}_2)$.*
- *$A \vDash M = N$ iff $B \vDash M\rho = N\rho$.*

We say process P is HPf-simulated by Q, and write $P \preceq_{HPf} Q$, whenever there exists an HPf-simulation $\mathscr{R}$ such that $\text{id} \mid P\ \mathscr{R}^{\text{id},\emptyset}\ \text{id} \mid Q$.

To see why HPf-similarity can be used to distinguish the processes in Eq. 5, observe that after inputting a message encrypted with k in two possible ways, we can tell that, on the right, in at least one case there will be an output message on channel *a that is dependent on the input*. Yet on the left it is possible, in both cases, that neither can perform such an output. An important part of this is the dependencies of the error message that we do not see, since all messages are indistinguishable to the attacker who does not know k, and hence cannot tell by looking at the message whether it is an error message.

Interestingly, anything coarser than HPf-similarity would not distinguish the processes in Eq. 5, since we use branching-time (so they are pomset failure trace equivalent[6]), failures (so they are HP-similar), *and* causality preservation (so they are ST-bisimilar): we need all the features of HPf-similarity.

[5] This definition is "diffed" against Def. 12.

[6] We do not define failure trace semantics in this paper, however it is easy to see how to obtain it via our approach to located aliases in Sect. 3 combined with classic definitions [2, 31].

5 Comparison to located bisimulations

This section compares our definitions to located equivalences, to help explain some less obvious design decisions. Early work on LATS for CCS defined a notion of bisimilarity preserving independence [24]. A key difference compared to our definition of HP-bisimilarity is that all events are accumulated in a history of events, whereas our definition remembers only those events that are currently active, and need not yet have terminated. Remembering all events may appear to simplify things, but we explain in this section that doing so gives rise to located equivalences that preserve the location of events, but forget about causal dependencies. To see this, consider the following processes, which are equivalent, even with respect to HP-bisimilarity.

$$L_1 \triangleq \nu b.\,\big(\overline{a}\langle a\rangle.\overline{b}\langle b\rangle \mid b(x).\overline{c}\langle c\rangle\big) \qquad \sim_{HP} \qquad \nu b.\,\big(\overline{b}\langle b\rangle \mid \overline{a}\langle a\rangle.b(x).\overline{c}\langle c\rangle\big) \triangleq L_2$$

To see why these processes are HP-bisimilar observe there are only three possible transitions for both processes, and one choice of alias substitution, as follows.

$$\mathrm{id} \mid L_1 \xrightarrow[0[]]{\overline{a}\langle 0\lambda\rangle} \xrightarrow[(0[],1[])]{\tau} \xrightarrow[1[]]{\overline{c}\langle 1\lambda\rangle} \qquad \mathrm{id} \mid L_2 \xrightarrow[1[]]{\overline{a}\langle 1\lambda\rangle} \xrightarrow[(0[],1[])]{\tau} \xrightarrow[1[]]{\overline{c}\langle 1\lambda'\rangle} \qquad \rho : 0\lambda \mapsto 1\lambda \qquad \rho : 1\lambda \mapsto 1\lambda'$$

There are no other transitions (modulo renaming λ, of course), and none of these events can be permuted. Notice that after each step the next transition is not independent of the currently started transitions, hence any started event must be removed from the set of active independent transitions S_1 for the game to continue. Therefore, we can pair the four states of these processes to form an HP-bisimulation.

In contrast, for the established located bisimilarities based on a LATS, the set of all events that have happened is accumulated in $\mathscr{E}$, and the independence of our LATS is preserved over all events. That is, we remember all pairs of events, and preserve independence everywhere, as captured by the following definition.

Definition 15 (I-consistent relation)**.** *For some symmetric relation over events I, an I-consistent relation over a set of events, say $\mathscr{E}$, is such that if $(e_0,d_0)\in\mathscr{E}$ and $(e_1,d_1)\in\mathscr{E}$ then $e_0\,I\,e_1$ iff $d_0\,I\,d_1$.*

The definition above can be instantiated with any notion of independence over events, such as I_ℓ or $\smile$ as in Def. 9, denoted here by I.

Now if we accumulate all pairs of events for our example above we obtain, after three transitions, the relation over events $\mathscr{E}$ defined as follows.

$$(\overline{a}\langle 0\lambda\rangle, 0[])\,\mathscr{E}\,(\overline{a}\langle 1\lambda\rangle, 1[]) \qquad (\tau,(0[],1[]))\,\mathscr{E}\,(\tau,(0[],1[])) \qquad (\overline{c}\langle 1\lambda\rangle, 1[])\,\mathscr{E}\,\big(\overline{c}\langle 1\lambda'\rangle, 1[]\big)$$

Taking the relation I to be $\smile$, we have that the above is not $\smile$-consistent, since $(\overline{a}\langle 0\lambda\rangle, 0[])\;I_\ell\;(\overline{c}\langle 1\lambda\rangle, 1[])$ holds but $(\overline{a}\langle 1\lambda\rangle, 1[])\;I_\ell\;(\overline{c}\langle 1\lambda'\rangle, 1[])$ does not.

An immediate consequence of the above is that the definition of bisimulation based on I-consistency, defined below, preserves the location of events more strongly than HP-bisimilarity, which preserves causal relationships. Indeed when we take I to be I_ℓ, obtaining I_ℓ-bisimilarity, we obtain a located bisimilarity and located bisimilarities and HP-bisimilarities are known to be incomparable.

Definition 16 (I-similarity)**.** *Let $\mathscr{R}$ be a relation between pairs of extended processes and ρ be an alias substitution. We say $\mathscr{R}$ is an I-simulation whenever if $A\,\mathscr{R}^{\rho,\mathscr{E}}\,B$, then:*

- *$\mathscr{E}$ is I-consistent.*

- *If* $A \xrightarrow[u]{\pi} A'$ *then there exists* ρ', B', u', *and* π' *s.t.* $\rho\restriction_{\mathrm{dom}(A)} = \rho'\restriction_{\mathrm{dom}(A)}$, $B \xrightarrow[u']{\pi'} B'$, $\pi\rho' = \pi'$, *and* $A' \mathscr{R}^{\rho',\mathscr{E}\cup\{((\pi,u),(\pi',u'))\}} B'$.
- $A \vDash M = N$ *iff* $B \vDash M\rho = N\rho$.

We say process P I-simulates Q, and write $P \preceq_I Q$, *whenever there exists an I-simulation* $\mathscr{R}$ *s.t.* $\mathrm{id} \mid P \mathscr{R}^{\mathrm{id},\emptyset} \mathrm{id} \mid Q$. *If in addition* $\mathscr{R}$ *is symmetric, then P and Q are I-bisimilar, written* $P \sim_I Q$.

In a sense, it is just a coincidence that for CCS, the above definition exploits nicely the independence relation of CCS, which coincides with I_ℓ since there is no link causality, and hence is strongly linked to the definition of a LATS for CCS. If we try to use $\smile$-bisimilarity, using the full independence relation $\smile$ from Def. 9, that accounts for link causality, we end up with an awkward relation. This has to do with the fact that independence for a LATS for the π-calculus must respect link causality, which means, for example, that the following processes are $\smile$-bisimilar:

$$\nu n.(\overline{a}\langle n\rangle \mid n(x)) \quad \sim_{\smile} \quad \nu n.\overline{a}\langle n\rangle.n(x)$$

This is because for both processes, the two events can only execute in one order, and neither is independent of the other, hence the set of events are $\smile$-consistent. Yet these processes are not I_ℓ-bisimilar, since their pairing S is not I_ℓ-consistent. This is rather troubling when juxtapositioned with the observation that the following are not $\smile$-bisimilar.

$$\nu n.\left(\overline{a}\langle n\rangle \mid n(x).\overline{\mathrm{ok}}\langle \mathrm{ok}\rangle\right) \quad \not\sim_{\smile} \quad \nu n.\overline{a}\langle n\rangle.n(x).\overline{\mathrm{ok}}\langle \mathrm{ok}\rangle$$

Similarly to the above we have that the three events may only be fired in a given order. However, the resulting relation over events is not $\smile$-consistent, since the first and third events are independent for the left process above, but are not independent for the right process above. This seems strange that the first event of the sub-process $n(x).\overline{\mathrm{ok}}\langle \mathrm{ok}\rangle$ is somehow not location-sensitive, yet the second is. To us, this is morally broken, hence $\sim_{\smile}$ is ill-defined. On the other hand $\sim_{I_\ell}$ consistently distinguishes these two examples, where the former involves two locations while the latter involves only one location.

Indeed $\sim_{I_\ell}$ is the notion of bisimilarity that would be obtained from the notion of trace equivalence implemented in the equivalence checking tool DeepSec [9]. They call their equivalence session equivalence and define it for a fragment of the applied π-calculus only. It is clear that a notion of trace equivalence that ensures that the events in compared traces are I_ℓ-consistent is the session equivalence of DeepSec. Intuitively, this is because session equivalence forms a bijection between processes in distinct locations and matches the behaviours in each location, which is exactly what I_ℓ-consistency would demand. Interestingly, that tool employs partial order reduction to improve equivalence checking; which is evidence that POR might be lifted to other notions of equivalence defined in this paper.

Thus, for the π-calculus and its extensions, there seems to be no real connection between $\smile$ and located bisimilarity; effectively we throw away part of the LATS to obtain a located bisimilarity [27]. The above observations help explain two things. Firstly, why we chose to target equivalences related to ST-similarity and HP-similarity rather than located bisimilarities in this work. Secondly, why our definitions are more complicated than those for located bisimilarities for CCS in the literature.

6 Conclusion

Having introduced a LATS for the applied π-calculus [4], we have shown that a world of non-interleaving operational semantics opens up for value passing process calculi. Notably, by using the independence relation (Def. 9) of a LATS, we capture ST-bisimilarity (Def. 11) and HP-bisimilarity (Def. 12) that reflect

correctly link causality, which were not preserved by established located bisimilarities for the π-calculus. Both semantics have their merits: for infinite processes, ST-semantics are very close to interleaving semantics, while being naturally compatible with the independence relation of a LATS; while HP-semantics better preserves the testing of finite subcomponents, even when we consider limits and infinite process. Eq. 4 showed that HP-similarity is able to detect attacks that are detectable using interleaving similarity for finite systems, yet are not detectable even by the strictly more powerful ST-bisimilarity when we take limits. This observation is reinforced in Eq. 5 where we show that HP failure similarity picks up on attacks that would be missed by anything coarser in any dimension (ST-bimilarity, HP-similarity, or even pomset failure traces). Since HP-bisimilarity would equally pick up on the attacks, we suggest HP-bisimilarity may be a good choice for security.

Having these definitions opens up formal and practical questions. It is non-trivial to verify that these definitions are the same as what we would expect if we pass via the more denotational world of event structures, configuration structures, or ST-structures [17, 23]. It is also non-trivial to provide characterisations using tests and modal logics [20]. What is fairly clear is that the relationship between these notions, since we start with the minimal notion of presimilarity and grow from there, providing separating examples at each step. The practical questions are more pressing, in particular, whether we can make use of ST- and HP-semantics in tools for protocol verification.

Acknowledgements The definitions in this paper are introduced to support an invited talk by the second author at EXPRESS/SOS on proving privacy properties using bisimilarity. We thank the organisers Valentina Castiglioni and Claudio Antares Mezzina for this invitation.

References

[1] Martín Abadi, Bruno Blanchet & Cédric Fournet (2018): *The Applied Pi Calculus: Mobile Values, New Names, and Secure Communication*. *J. ACM* 65(1), pp. 1:1–1:41, doi:10.1145/3127586.

[2] Luca Aceto & Uffe Engberg (1991): *Failures semantics for a simple process language with refinement*. In Somenath Biswas & Kesav V. Nori, editors: *Foundations of Software Technology and Theoretical Computer Science*, Springer, pp. 89–108, doi:10.1007/3-540-54967-6_63.

[3] Luca Aceto & Matthew Hennessy (1994): *Adding action refinement to a finite process algebra*. *Inform. and Comput.* 115(2), pp. 179–247, doi:10.1006/inco.1994.1096.

[4] Clément Aubert, Ross Horne & Christian Johansen (2022): *Diamonds for Security: A Non-Interleaving Operational Semantics for the Applied Pi-Calculus*. In Bartek Klin, Sławomir Lasota & Anca Muscholl, editors: *33rd International Conference on Concurrency Theory*, *Leibniz International Proceedings in Informatics* 243, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 30:1–30:26, doi:10.4230/LIPIcs.CONCUR.2022.30.

[5] Michele Boreale & Davide Sangiorgi (1998): *A fully abstract semantics for causality in the π-calculus*. *Acta Inform.* 35(5), pp. 353–400, doi:10.1007/s002360050124.

[6] Gérard Boudol, Ilaria Castellani, Matthew Hennessy & Astrid Kiehn (1994): *A Theory of Processes with Localities*. *Formal Aspects Comput.* 6(2), pp. 165–200, doi:10.1007/BF01221098.

[7] Ilaria Castellani (1995): *Observing distribution in processes: static and dynamic localities*. *Int. J. Found. Comput. Sci.* 6(04), pp. 353–393, doi:10.1142/S0129054195000196.

[8] V. Cheval, R. Crubillé & S. Kremer (2022): *Symbolic Protocol Verification with Dice: Process Equivalences in the Presence of Probabilities*. In: *2022 2022 IEEE 35th Computer Security Foundations Symposium (CSF) (CSF)*, IEEE Computer Society, Los Alamitos, CA, USA, pp. 303–318, doi:10.1109/CSF54842.2022.00020.

[9] Vincent Cheval, Steve Kremer & Itsaka Rakotonirina (2019): *Exploiting Symmetries When Proving Equivalence Properties for Security Protocols*. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang & Jonathan Katz, editors: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, ACM, pp. 905–922, doi:10.1145/3319535.3354260.

[10] Silvia Crafa, Daniele Varacca & Nobuko Yoshida (2012): *Event Structure Semantics of Parallel Extrusion in the Pi-Calculus*. 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*, *LNCS* 7213, Springer, pp. 225–239, doi:10.1007/978-3-642-28729-9_15.

[11] Ioana Cristescu, Jean Krivine & Daniele Varacca (2015): *Rigid Families for CCS and the π-calculus*. In Martin Leucker, Camilo Rueda & Frank D. Valencia, editors: *Theoretical Aspects of Computing - ICTAC 2015 - 12th International Colloquium Cali, Colombia, October 29-31, 2015, Proceedings*, *LNCS* 9399, Springer, pp. 223–240, doi:10.1007/978-3-319-25150-9_14.

[12] Pierpaolo Degano, Rocco De Nicola & Ugo Montanari (1989): *Partial orderings descriptions and observations of nondeterministic concurrent processes*. In J. W. de Bakker, W. P. de Roever & G. Rozenberg, editors: *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, Springer, pp. 438–466, doi:10.1007/BFb0013030.

[13] Yuxin Deng, Matthew Hennessy, Rob van Glabbeek & Carroll Morgan (2008): *Characterising Testing Preorders for Finite Probabilistic Processes*. *Log. Methods Comput. Sci.* Volume 4, Issue 4, doi:10.2168/LMCS-4(4:4)2008.

[14] Ihor Filimonov, Ross Horne, Sjouke Mauw & Zach Smith (2019): *Breaking Unlinkability of the ICAO 9303 Standard for e-Passports Using Bisimilarity*. In Kazue Sako, Steve A. Schneider & Peter Y. A. Ryan, editors: *Computer Security - ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part I*, *LNCS* 11735, Springer, pp. 577–594, doi:10.1007/978-3-030-29959-0_28.

[15] Rob van Glabbeek (1990): *The refinement theorem for ST-bisimulation semantics*. Technical Report R 9002, Centre for Mathematics and Computer Science. Available at https://ir.cwi.nl/pub/5765.

[16] Rob van Glabbeek (2001): *The linear time-branching time spectrum I. The semantics of concrete, sequential processes*. In J. A. Bergstra, A. Ponse & S. A. Smolka, editors: *Handbook of process algebra*, Elsevier, pp. 3–99, doi:10.1016/b978-044482830-9/50019-9.

[17] Rob van Glabbeek & Gordon D. Plotkin (2009): *Configuration structures, event structures and Petri nets*. *Theor. Comput. Sci.* 410(41), pp. 4111–4159, doi:10.1016/j.tcs.2009.06.014.

[18] Rob van Glabbeek & Frits W. Vaandrager (1997): *The Difference between Splitting in n and $n+1$*. *Inf. Comput.* 136(2), pp. 109–142, doi:10.1006/inco.1997.2634.

[19] Roberto Gorrieri & Cosimo Laneve (1995): *Split and ST Bisimulation Semantics*. *Inf. Comput.* 118(2), pp. 272–288, doi:10.1006/inco.1995.1066.

[20] Matthew Hennessy (1995): *Concurrent Testing of Processes*. *Acta Informatica* 32(6), pp. 509–543, doi:10.1007/BF01178906.

[21] Ross Horne & Sjouke Mauw (2021): *Discovering ePassport Vulnerabilities using Bisimilarity*. *Log. Meth. Comput. Sci.* 17(2), p. 24, doi:10.23638/LMCS-17(2:24)2021.

[22] Ross Horne, Sjouke Mauw & Semen Yurkov (2021): *Compositional Analysis of Protocol Equivalence in the Applied π-Calculus Using Quasi-open Bisimilarity*. In Antonio Cerone & Peter Csaba Ölveczky, editors: *Theoretical Aspects of Computing - ICTAC 2021 - 18th International Colloquium, Virtual Event, Nur-Sultan, Kazakhstan, September 8-10, 2021, Proceedings*, *LNCS* 12819, Springer, pp. 235–255, doi:10.1007/978-3-030-85315-0_14.

[23] Christian Johansen (2016): *ST-structures*. *J. Log. Algebraic Methods Program.* 85(6), pp. 1201–1233, doi:10.1016/j.jlamp.2015.10.009.

[24] Madhavan Mukund & Mogens Nielsen (1992): *CCS, Location and Asynchronous Transition Systems*. In R. K. Shyamasundar, editor: *Foundations of Software Technology and Theoretical Computer Science, 12th Conference, New Delhi, India, December 18-20, 1992, Proceedings*, *LNCS* 652, Springer, pp. 328–341, doi:10.1007/3-540-56287-7_116.

[25] Joachim Parrow, Johannes Borgström, Lars-Henrik Eriksson, Ramunas Gutkovas & Tjark Weber (2021): *Modal Logics for Nominal Transition Systems*. *Log. Meth. Comput. Sci.* 17(1), pp. 6:1–6:49, doi:10.23638/LMCS-17(1:6)2021.

[26] Alexander Rabinovich & Boris Avraamovich Trakhtenbrot (1988): *Behavior Structures and Nets*. *Fund. Inform.* 11(4), pp. 357–404, doi:10.3233/FI-1988-11404.

[27] Davide Sangiorgi (1996): *Locality and interleaving semantics in calculi for mobile processes*. *Theor. Comput. Sci.* 155(1), pp. 39–83, doi:10.1016/0304-3975(95)00020-8.

[28] Davide Sangiorgi (1996): *A Theory of Bisimulation for the pi-Calculus*. *Acta Inform.* 33(1), pp. 69–97, doi:10.1007/s002360050036.

[29] Davide Sangiorgi & David Walker (2001): *On Barbed Equivalences in pi-Calculus*. In Kim Guldstrand Larsen & Mogens Nielsen, editors: *CONCUR 2001 - Concurrency Theory, 12th International Conference, Aalborg, Denmark, August 20-25, 2001, Proceedings*, *LNCS* 2154, Springer, pp. 292–304, doi:10.1007/3-540-44685-0_20.

[30] Daniele Varacca & Nobuko Yoshida (2010): *Typed event structures and the linear pi-calculus*. *Theor. Comput. Sci.* 411(19), pp. 1949–1973, doi:10.1016/j.tcs.2010.01.024.

[31] Walter Vogler (1991): *Failures semantics based on interval semiwords is a congruence for refinement*. *Distributed Computing* 4(3), pp. 139–162, doi:10.1007/BF01798961.

[32] Walter Vogler (1996): *The Limit of Split$_n$-Language Equivalence*. *Inf. Comput.* 127(1), pp. 41–61, doi:10.1006/inco.1996.0048.