On the semantics of communications when verifying equivalence properties
Kushal Babel, Vincent Cheval, Steve Kremer

To cite this version:

HAL Id: hal-02446910
https://inria.hal.science/hal-02446910
Submitted on 21 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
On the semantics of communications when verifying equivalence properties

Kushal Babel\textsuperscript{a,\!*}, Vincent Cheval\textsuperscript{b,\!*\!*}, and Steve Kremer\textsuperscript{b}

\textsuperscript{a} Cornell University, US
E-mail: babel@cs.cornell.edu

\textsuperscript{b} Inria Nancy Grand-Est, Loria, France
E-mails: vincent.cheval@inria.fr, steve.kremer@inria.fr

Abstract. Symbolic models for security protocol verification were pioneered by Dolev and Yao in their seminal work. Since then, although inspired by the same ideas, many variants of the original model were developed. In particular, a common assumption is that the attacker has complete control over the network and can therefore intercept any message. This assumption has been interpreted in slightly different ways depending on the particular models: either any protocol output is directly routed to the adversary, or communications may be among any two participants, including the attacker — the scheduling between which exact parties the communication happens is left to the attacker. This difference may seem unimportant at first glance and, depending on the verification tools, either one or the other semantics is implemented. We show that, unsurprisingly, they indeed coincide for reachability properties. However, for indistinguishability properties, we prove that these two interpretations lead to incomparable semantics. We also introduce and study a new semantics, where internal communications are allowed but messages are always eavesdropped by the attacker. This new semantics yields strictly stronger equivalence relations. Moreover, we identify two subclasses of protocols for which the three semantics coincide. Finally, we implemented verification of trace equivalence for each of the three semantics in the \textit{DeepSec} tool and compare their performances on several classical examples.

Keywords: Cryptographic protocols, Symbolic models, Verification, Semantics, Equivalence properties

1. Introduction

Automated, symbolic analysis of security protocols, based on the seminal ideas of Dolev and Yao, comes in many variants. All of these models however share a few fundamental ideas:

- messages are represented as abstract terms,
- adversaries are computationally unbounded, but may manipulate messages only according to pre-defined rules (this is sometimes referred to as the perfect cryptography assumption), and
- the adversary completely controls the network.

\!* Part of the work was carried out while the first author was a student at IIT Bombay during an internship at Inria Nancy Grand-Est.
\!*\!* Corresponding author. E-mail: vincent.cheval@inria.fr.
In this paper we will revisit this last assumption. Looking more precisely at different models we observe that this assumption may actually slightly differ among the models. The fact that the adversary controls the network is supposed to represent a worst case assumption.

In some models this assumption translates to the fact that every protocol output is sent to the adversary, and every protocol input is provided by the adversary. This is the case in the original Dolev Yao model and also in the models underlying several tools, such as AVISPA [1], Scyther [2], Tamarin [3], Millen and Shmatikov’s constraint solver [4], and the model used in Paulson’s inductive approach [5]. We will refer to this choice of semantics as the private semantics, as internal communications are only allowed on private channels.

Some other models, such as those based on process algebras, e.g. work based on CSP [6], the Spi [7] and applied pi calculus [8], but also the strand space model [9], consider a slightly different communication model: any two agents may communicate. Scheduling whether communication happens among two honest participants, or a honest participant and the attacker is under the attacker’s control. We will refer to this choice of semantics as the classical semantics, as it corresponds to what is generally used in process calculi.

When considering reachability properties, these two communication models indeed coincide: intuitively, any internal communication could go through the adversary who acts as a relay and increases his knowledge by the transmitted message. However, when considering indistinguishability properties, typically modelled as process equivalences, these communication models diverge. Interestingly, when forbidding internal communication, i.e., forcing all communication to be relayed by the attacker, we may weaken the attacker’s distinguishing power. This observation may seem counter-intuitive at first. However, executing a (non-observable) internal communication may enable actions that are otherwise only available after an observable input. These actions may then provide additional capabilities for simulating the other process.

In many recent work privacy properties have been modelled using process equivalences, see for instance [10–12]. The number of tools able to verify such properties is also increasing [13–18]. For instance, the AKISS [16] and SAT-EQUIV [17] tools do not allow any direct communication on public channels, while the APTE [15] and DeepSec [18] tools allow for internal communications. One motivation for disallowing direct communication is that it allows for more efficient verification (as less actions need to be considered and the number of interleavings to be considered is smaller).

Our contributions. We have formalised three semantics in the applied pi calculus which differ by the way communication is handled:

- the classical semantics (as in the original applied pi calculus) allows both internal communication among honest participants and communication with the adversary;
- the private semantics allows internal communication only on private channels while all communication on public channels is routed through the adversary;
- the eavesdropping semantics which allows internal communication, but as a side-effect adds the transmitted message to the adversary’s knowledge.

For each of the new semantics we define may-testing and observational equivalences. We also define corresponding labelled semantics and trace equivalence and bisimulation relations (which may serve as proof techniques).

We show that, as expected, the three semantics coincide for reachability properties. For equivalence properties we show that the classical and private semantics yield incomparable equivalences,
while the eavesdropping semantics yields strictly stronger equivalence relations than both other semantics. The results are summarized in Figure 4.

An interesting question is whether these semantics coincide for specific subclasses of processes. We note that the processes that witness the differences in the semantics do not use replication, private channels, nor terms other than names, and no equational theory. Moreover, all except one of these examples only use trivial else branches (of the form else 0); the use of a non-trivial else branch can even be avoided by allowing a single free symbol.

We first study different notions of determinate processes: in the context of the applied pi calculus, Cheval et al. [19] have for instance shown that observational, testing, trace equivalence and labelled bisimulation coincide for this class of processes (for the classic semantics). We will show that this is actually the case for all semantics and show, among others that the private and eavesdropping semantics do coincide on these equivalences, and imply them for the classic semantics. We consider several specific subclasses of determinate processes when we bound the number of sessions. In particular, we show that all equivalences and semantics coincide for the class of strong action determinate processes. This class is of practical importance as this condition is checked in the AKISS and DeepSec tools to enable partial order reduction optimizations [20]. These optimizations provide spectacular speed-ups, but they were designed and shown correct only in the private semantics. Showing that all three semantics coincide for strong action determinate processes lifts the benefit of these optimizations to the other semantics. The results on subclasses of determinate processes are summarized in Figure 8.

We also identify a syntactic class of processes, that we call I/O-unambiguous. For this class we forbid communication on private channels, communication of channel names and an output may not be sequentially followed by an input on the same channel directly, or with only conditionals in between. Note however that I/O-unambiguous processes, unlike most determinate processes, do allow outputs and inputs on the same channel in parallel. We show that for this class the eavesdropping semantics (which is the most strict relation) coincides with the private one (which is the most efficient for verification).

Finally, we extended the DeepSec tool to support verification of trace equivalence for the three semantics. Verifying existing protocols in the DeepSec example repository we verified that the results, fortunately, coincided for each of the semantics. We also made slight changes to the encodings, renaming some channels, to make them I/O-unambiguous. Interestingly, using different channels, significantly increased the performance of the tool. Finally, we also observed that, as expected, the private semantics yields more efficient verification. The results of our experiments are summarized in Section 5.

A preliminary version of this work appeared in [21]. In contrast to [21], this work contains full proofs of all results, new results for several subclasses of processes, giving a detailed comparison of the different semantics and equivalences, as well as an implementation of all three semantics in the DeepSec tool, together with an experimental evaluation.

Outline. In Section 2 we define the three semantics we consider. In Section 3 we present our main results on comparing these semantics. We present subclasses for which (some) semantics coincide in Section 4 and compare the performances when verifying protocols for different semantics using DeepSec in Section 5, before concluding in Section 6.
2. Model

The applied pi calculus [8] is a variant of the pi calculus that is specialised for modelling cryptographic protocols. Participants in a protocol are modelled as processes and the communication between them is modelled by message passing on channels. In this section, we describe the syntax and semantics of the applied pi calculus as well as the two new variants that we study in this paper.

2.1. Syntax

We consider an infinite set \( N \) of names of base type and an infinite set \( C \) of names of channel type. We also consider an infinite set of variables \( X \) of base type and channel type and a signature \( F \) consisting of a finite set of function symbols. We rely on a sort system for terms. In particular, the sort base type differs from the sort channel type. Moreover, any function symbol can only be applied to and returns base type terms. We define terms as names, variables and function symbols applied to other terms. Given \( N \subseteq N, X \subseteq X \) and \( F \subseteq F \), we denote by \( T(F,X,N) \) the sets of terms built from \( X \) and \( N \) by applying function symbols from \( F \). We denote \( v(t) \) the sets of variables occurring in \( t \). We say that \( t \) is ground if \( v(t) = \emptyset \). We describe the behaviour of cryptographic primitives by the means of an equational theory \( E \) that is a relation on terms closed under substitutions of terms for variables and closed under one-to-one renaming. Given two terms \( u \) and \( v \), we write \( u =_E v \) when \( u \) and \( v \) are equal modulo the equational theory.

In the original syntax of the applied pi calculus, there is no distinction between an output (resp. input) from a protocol participant and from the environment, also called the attacker. In this paper however, we will make this distinction in order to concisely present our new variants of the semantics. Therefore, we consider two process tags \( \text{ho} \) and \( \text{at} \) that respectively represent honest and attacker actions. The syntax of plain processes and extended processes is given in Figure 1.

\[
P, Q := 0 \quad \text{plain processes} \quad A, B := P \quad \text{extended processes}
\]

\[
P \mid Q \quad A \mid B
\]

\[
\nu n. P \quad \nu x. A
\]

\[
\text{if } u = v \text{ then } P \text{ else } Q
\]

\[
\text{in}^\theta(c, x). P
\]

\[
\text{out}^\theta(c, u). P
\]

\[
eav(c, x). P
\]

where \( u \) and \( v \) are base type terms, \( n \) is a name, \( x \) is a variable and \( c \) is a name or variable of channel type, \( \theta \) is a tag, i.e. \( \theta \in \{\text{ho}, \text{at}\} \).

Fig. 1. Syntax of processes

The process \( \text{out}^\theta(c, u) \) represents the output by \( \theta \) of the message \( u \) on the channel \( c \). The process \( \text{in}^\theta(c, x) \) represents an input by \( \theta \) on the channel \( c \). The input message will instantiate the variable \( x \). The process \( \text{eav}(c, x) \) models the capability of the attacker to eavesdrop a communication on channel \( c \). The process \( !P \) represents the replication of the process \( P \), i.e. unbounded number
of copies of \( P \). The process \( P \mid Q \) represents the parallel composition of \( P \) and \( Q \). The process 
\( \nu n.P \) (resp. \( \nu x.A \)) is the restriction of the name \( n \) in \( P \) (resp. variable \( x \) in \( A \)). The process 
if \( u = v \) then \( P \) else \( Q \) is the conditional branching under the equality test \( u = v \). The process \( \omega c \)
records that a private channel \( c \) has been opened, i.e., it has been sent on a public or previously
opened channel. Finally, the substitution \( \{u/x\} \) is an active substitution that replaces the variable
\( x \) with the term \( u \) of base type.

We say that a process \( P \) (resp. extended process \( A \)) is an honest process (resp. honest extended
process) when all inputs and outputs in \( P \) (resp. \( A \)) are tagged with \( ho \) and when \( P \) (resp. \( A \)) does
not contain eavesdropping processes and \( \omega c \). We say that a process \( P \) (resp. extended process \( A \))
is an attacker process (resp. attacker extended process) when all inputs and outputs in \( P \) (resp.
\( A \)) are tagged with \( at \).

As usual, names and variables have scopes which are delimited by restrictions, inputs and
eavesdrops. We denote \( fv(A) \), \( bv(A) \), \( fn(A) \), \( bn(A) \) the sets of free variables, bound variables, free
names and bound names respectively in \( A \). Moreover, we denote by \( oc(A) \) the sets of terms \( c \)
of channel type opened in \( A \), i.e. that occurs in a process \( \omega c \). We say that an extended process
\( A \) is closed when all variables in \( A \) are either bound or defined by an active substitution in \( A \).
We define an evaluation context \( C[\_\_] \) as an extended process with a hole instead of an extended
process. As for processes, we define an attacker evaluation context as an evaluation context where
all outputs and inputs in the context are tagged with \( at \).

Note that our syntax without the eavesdropping process, opened channels and tags correspond
exactly to the syntax of the original applied pi calculus.

Lastly, we consider the notion of frame that are extended processes built from 0, parallel com-
position, name and variable restrictions and active substitution. Given a frame \( \varphi \), we consider
the domain of \( \varphi \), denoted \( dom(\varphi) \), as the set of free variables in \( \varphi \) that are defined by an active
substitution in \( \varphi \). Given an extended process \( A \), we define the frame of \( A \), denoted \( \phi(A) \), as the
process \( \varphi \) where we replace all plain processes by \( \varphi \). Finally, we write \( dom(A) \) as syntactic sugar
for \( dom(\phi(A)) \).

2.2. Operational semantics

In this section, we define the three semantics that we study in this paper, namely:

- the classical semantics from the applied pi calculus, where internal communication can occur
  on both public and private channels;
- the private semantics where internal communication can only occur on private channels;
  and
- the eavesdropping semantics where the attacker is able to eavesdrop on a public channel.

We first define the structural equivalence between extended processes, denoted \( \equiv \), as the smallest
equivalence relation on extended processes that is closed under renaming of names and variables,
closed by application of evaluation contexts, that is associative and commutative w.r.t. \( \mid \), and such
that:

\[
A \equiv A \mid 0 \quad !P \equiv !P \mid P \quad \nu n.0 \equiv 0
\]

\[
\nu i.\nu j.A \equiv \nu j.\nu i.A \quad \nu x.\{u/x\} \equiv 0 \quad \{u/x\} \mid A \equiv \{u/x\} \mid A\{u/x\}
\]

\[
A \mid \nu i.B \equiv \nu i.(A \mid B) \quad \text{when } i \notin fv(A) \cup fn(A) \quad \omega c \equiv \omega c \mid \omega c
\]

\[
\{u/x\} \equiv \{v/x\} \quad \text{when } u =_{\equiv} v
\]
The three operational semantics of extended processes are defined by the structural equivalence and by three respective internal reductions, denoted $\Rightarrow_c$, $\Rightarrow_p$, and $\Rightarrow_e$. These three reductions are the smallest relations on extended processes that are closed under application of evaluation context, structural equivalence and such that:

if $u = v$ then $P$ else $Q \Rightarrow_s P$ where $u =_E v$ and $s \in \{c, p, e\}$  
if $u = v$ then $P$ else $Q \Rightarrow_s Q$  
where $u, v$ ground, $u \neq_E v$ and $s \in \{c, p, e\}$

\[
\text{out}^\theta(c, u).P | \text{in}^\theta(c, x).Q \Rightarrow_c P | Q^{\{u/x\}}
\]

\[
\nu_c.(\text{out}^\theta(c, u).P | \text{in}^\theta(c, x).Q | R) \Rightarrow_s \nu_c.(P | Q^{\{u/x\}} | R)
\]
where $c \not\in \text{oc}(R)$ and $s \in \{p, e\}$

\[
\text{out}^\theta(c, u).P | \text{in}^\theta(c, x).Q \Rightarrow_s P | Q^{\{u/x\}}
\]
where $c \not\in \text{oc}(R)$ and $s \in \{p, e\}$

\[
\text{out}^\theta(c, d).P | \text{in}^\theta(c, x).Q \Rightarrow_s P | Q^{\{d/x\}} | \omega d
\]
where $c \not\in \text{oc}(R)$ and $s \in \{p, e\}$

\[
\text{out}^{ho}(c, u).P | \text{in}^{ho}(c, x).Q | \text{eav}(c, y).R \Rightarrow_e P | Q^{\{u/y\}} | R^{\{u/y\}}
\]
where $c \not\in \text{oc}(R)$ and $s \in \{p, e\}$

\[
\text{out}^{ho}(c, d).P | \text{in}^{ho}(c, x).Q | \text{eav}(c, y).R \Rightarrow_e P | Q^{\{d/y\}} | R^{\{d/y\}} | \omega d
\]
where $c \not\in \text{oc}(R)$ and $s \in \{p, e\}$

We emphasise that the application of the rule is closed under application of arbitrary evaluation contexts. In particular the context may restrict channels, e.g. the rule C-OPEN may be used under the context $\nu_c._c$ resulting in a private channel $c$, but with the attacker input/output being in the scope of this restriction. It follows from the definition of evaluation contexts that the resulting processes are always well defined. We denote by $\Rightarrow_s$ the reflexive, transitive closure of $\Rightarrow_s$ for $s \in \{c, p, e\}$. We note that the classical semantics $\Rightarrow_c$ is independent of the tags $\theta, \theta'$, the eavesdrop actions and the $\omega_c$ processes.

**Example 1.** Consider the process

\[
A = (\nu d.\text{out}^\theta(c, d).\text{in}^\theta(d, x).P | (\text{in}^\theta(c, y).\text{out}^\theta(y, t).Q)
\]

where $d$ is a channel name and $t$ a term of base type. Suppose $\theta = \theta' = \text{ho}$ then we have that communication is only possible in the classical semantics (using twice the COMM rule):

\[
A \Rightarrow_c \nu d.(\text{in}^\theta(d, x).P | \text{out}^\theta(d, t).Q^{\{d/y\}})
\]

while no transitions are available in the two other semantics. To enable communication in the eavesdropping semantics we need to explicitly add eavesdrop actions. Applying the rules C-OEAV
and C-Eav we have that

\[ A \parallel \text{eav}(c, z_1).\text{eav}(z_1, z_2).R \xrightarrow{\tau_e} \nu d. (\in^\theta(d, x). P \parallel \text{out}^\theta(d, t). Q^{d/y}) \parallel \text{eav}(d, z_2).R^{d/z_1} \parallel \omega d \]

\[ \xrightarrow{\tau_e} \nu d. (P^{t/x} \parallel Q^{d/y} \parallel R^{d/z_1} \parallel \omega d) \]

We note that the first transition adds the information \( \omega d \) to indicate that \( d \) is now available to the environment.

Finally, if we consider that \( at \in \{ \theta, \theta' \} \) then internal communication on a public channel is possible and, using rules C-Open and C-Env we obtain for \( s \in \{ p, e \} \) that

\[ A \xrightarrow{\tau_s} \nu d. (\in^\theta(d, x). P \parallel \text{out}^\theta(d, t). Q^{d/y}) \parallel \omega d \]

\[ \xrightarrow{\tau_s} \nu d. (P^{t/x} \parallel Q^{d/y} \parallel \omega d) \]

### 2.3. Reachability and behavioural equivalences

We are going to compare the relation between the three semantics for the two general kind of security properties, namely reachability properties encoding security properties such as secrecy, authentication, and equivalence properties encoding properties such as anonymity, unlinkability, strong secrecy, and receipt freeness. Intuitively, reachability properties encode that a process cannot reach some bad state. Equivalences define the fact that no attacker can distinguish two processes. This was originally defined by the (may)-testing equivalence [7] in the spi-calculus.

An alternate equivalence, which was considered in the applied pi calculus [8], is observational equivalence.

Reachability properties can simply be encoded by verifying the capability of a process to perform an output on a given channel. We define \( A \not\triangleright^\nu_c\theta \) to hold when \( A \Rightarrow_s C[\text{out}^\theta(c, t). P] \) for some evaluation context \( C \) that does not bind \( c \), some term \( t \) and some plain process \( P \), and \( A \not\triangleright^\nu_c\theta \) to hold when \( A \not\triangleright^\nu_c\theta \) for some \( \theta \in \{ at, ho \} \). For example the secrecy of \( s \) in the process \( \nu s.A \) can be encoded by checking whether for all attacker plain process \( I \), we have that

\[ I \mid \nu s.(A \mid \text{in}^{ho}(c, x) \text{if } x = s \text{ then out}^{ho}(bad, s)) \not\downarrow_{\text{bad}}^{\nu s} \]

where \( \text{bad} \notin \text{fn}(A) \).

Authentication properties are generally expressed as correspondence properties between events annotating processes, see e.g. [22]. A correspondence property between two events \text{begin} and \text{end}, denoted \( \text{begin} \Leftarrow \text{end} \), requires that the event \text{end} is preceded by the event \text{begin} on every trace. A possible encoding of this correspondence property consists in first replacing all instances of the events in \( A \) by outputs \( \text{out}^{ho}(ev, \text{begin}) \) and \( \text{out}^{ho}(ev, \text{end}) \) where \( ev \notin \text{fn}(A) \cup \text{bn}(A) \). This new process \( A' \) can then be put in parallel with a cell \( C\text{ell} \) that reads on the channel \( ev \) and stores any new value unless the value is \text{end} and the current stored value in the cell is not \text{begin}. In such a case, the cell will output on the channel \text{bad}. The correspondence property can therefore be encoded by checking whether for all attacker plain process \( I \), we have that \( I \mid \text{ev}v(A') \mid \text{C}\text{ell} \) \( \not\downarrow_{\text{bad}}^{\nu s} \).

We say that an attacker evaluation context \( C[\_] \) is c-closing for an extended process \( A \) if \( \text{fn}(C[A]) = \emptyset \). For \( s \in \{ p, e \} \), we say that \( C[\_] \) is \( s \)-closing for \( A \) if it is \( c \)-closing for \( A \), variables and names are bound only once in \( C[\_] \) and for all channels \( c \in \text{bn}(C[\_]) \cap \text{fn}(A) \), if the scope of \( c \) includes \( \_ \) then the scope of \( c \) also includes \( \omega c \).
We next introduce the two main notions of behavioural equivalences: may testing and observational equivalence.

**Definition 1** (May-test equivalences $\approx_{m}^{c}$, $\approx_{m}^{p}$, $\approx_{m}^{o}$). Let $s \in \{c, p, e\}$. Let $A$ and $B$ two closed honest extended processes such that $\text{dom}(A) = \text{dom}(B)$. We say that $A \approx_{m}^{s} B$ if for all attacker evaluation contexts $C[\_]$ $s$-closing for $A$ and $B$, for all channels $c$, we have that $C[A] \downarrow_{c}^{s}$ if and only if $C[B] \downarrow_{c}^{s}$.

**Definition 2** (Observational equivalences $\approx_{o}^{c}$, $\approx_{o}^{p}$, $\approx_{o}^{o}$). Let $s \in \{c, p, e\}$. Let $A$ and $B$ two closed extended processes such that $\text{dom}(A) = \text{dom}(B)$. We say that $A \approx_{o}^{s} B$ if $\approx_{o}^{s}$ is the largest equivalence relation such that:

- $A \downarrow_{c}^{s}$ implies $B \downarrow_{c}^{s}$;
- $A \Downarrow_{s} A'$ implies $B \DownRightarrow_{s} B'$ and $A' \approx_{o}^{s} B'$ for some $B'$;
- $C[A] \approx_{o}^{s} C[B]$ for all attacker evaluation contexts $C[\_]$ $s$-closing for $A$ and $B$.

For each of the semantics we have the usual relation between these two notions: observational equivalence implies testing equivalence.

**Proposition 1.** $\approx_{o}^{s} \subseteq \approx_{m}^{s}$ for $s \in \{c, e, p\}$.

**Example 2.** Consider processes $A$ and $B$ of Figure 2. Process $A$ computes a value $h^{n}(a)$ to be output on channel $c$, where $h^{n}(a)$ denotes $n$ applications of $h$ and $h^{0}(a) = a$. The value is initially $a$ and $A$ may choose to either output the current value, or update the current value by applying the free symbol $h$. $B$ may choose non-deterministically to either behave as $A$ or output the fresh name $s$. (The non-deterministic choice is encoded by a communication on the private channel $e$ which may be received by either the process behaving as $A$ or the process outputting $s$.)

We have that $A \not\approx_{o}^{s} B$. The two processes can indeed be distinguished by the context

$$\begin{align*}
C[\_] & \equiv \_ | \text{out}^{\_}(c_{a}, a) | ![\text{in}(c_{a}, x).\text{out}(c_{a}, h(x))] \\
& | \text{in}(c_{a}, y).\text{in}(c, z).\text{if} \ y = z \ \text{then} \ \text{out}(c_{i}, h(x))
\end{align*}$$

Intuitively, when $B$ outputs $s$ the attacker context $C[\_]$ can iterate the application of $h$ the same number of times as would have done process $A$. Comparing the value computed by the adversary ($h^{n}(a)$) and the honestly computed value (either $h^{n}(a)$ or $s$) the adversary distinguishes the two processes by outputting on the test channel $c_{t}$.

However, we have that $A \approx_{m}^{s} B$. Indeed, for any $s$-closing context $D[\_]$ and all public channel $ch$ we have that $D[A] \downarrow_{ch}^{s}$ if and only if $D[B] \downarrow_{ch}^{s}$. In particular for context $C[\_]$ defined above we have that both $C[A] \downarrow_{ch}^{s}$ and $C[B] \downarrow_{ch}^{s}$ for $ch \in \{c_{a}, c_{t}, c\}$. Unlike observational equivalence, may testing does not require to “mimic” the other process stepwise and we cannot force a process into a particular branch.
Example 3. Coming back to Example 1, we saw that actions in the other two semantics were available. Instead of explicitly adding eavesdrop actions, we extend the three operational semantics by a labeled operational semantics which allows processes to directly interact with (the adversarial) environment: we define the relation $\ell \equiv_c, \ell \equiv_p$ and $\ell \equiv_e$ where $\ell$ is part of the alphabet $A = \{\tau, out(c, d), eav(c, d), m(c, w), nk.out(c, k), nk.eav(c, k) \mid c, d \in Ch, k \in X \cup Ch$ and $w$ is a term of any sort}. The labeled rules are given in Figure 3.

Consider our alphabet of actions $A$ defined above. Given $w \in A^*$, $s \in \{c, p, e\}$ and an extended process $A$, we say that $A \xrightarrow{s} A^*$ when $A \xrightarrow{s} A_1 \xrightarrow{s} A_2 \xrightarrow{s} \ldots \xrightarrow{s} A_n$ for some extended processes $A_1, \ldots, A_n$ and $w = \ell_1 \cdots \ell_n$. By convention, we say that $A \xrightarrow{\varepsilon}$ for $A$ where $\varepsilon$ is the empty word. Given $tr \in (A \setminus \{\tau\})^*$, we say that $A \xrightarrow{w, s} A'$ when there exists $w \in A^*$ such that $tr$ is the word $w$ where we remove all $\tau$ actions and $A \xrightarrow{w, s} A'$.

In 2.4. Labelled semantics

The internal reduction semantics introduced in the previous section requires to reason about arbitrary contexts. Similar to the original applied pi calculus, we extend the three operational semantics by a labeled operational semantics which allows processes to directly interact with (the adversarial) environment: we define the relation $\ell \equiv_c, \ell \equiv_p$ and $\ell \equiv_e$ where $\ell$ is part of the alphabet $A = \{\tau, out(c, d), eav(c, d), m(c, w), nk.out(c, k), nk.eav(c, k) \mid c, d \in Ch, k \in X \cup Ch$ and $w$ is a term of any sort}. The labeled rules are given in Figure 3.

Consider our alphabet of actions $A$ defined above. Given $w \in A^*$, $s \in \{c, p, e\}$ and an extended process $A$, we say that $A \xrightarrow{s} A^*$ when $A \xrightarrow{s} A_1 \xrightarrow{s} A_2 \xrightarrow{s} \ldots \xrightarrow{s} A_n$ for some extended processes $A_1, \ldots, A_n$ and $w = \ell_1 \cdots \ell_n$. By convention, we say that $A \xrightarrow{\varepsilon}$ for $A$ where $\varepsilon$ is the empty word. Given $tr \in (A \setminus \{\tau\})^*$, we say that $A \xrightarrow{w, s} A'$ when there exists $w \in A^*$ such that $tr$ is the word $w$ where we remove all $\tau$ actions and $A \xrightarrow{w, s} A'$.

Example 3. Coming back to Example 1, we saw that actions in the other two semantics were available. Instead of explicitly adding eavesdrop actions,
we can apply the rules EAV-OCH and EAV-T and obtain that

\[
A \xrightarrow{\nu d.\text{eav}(c,d)} \tau_e \xrightarrow{\text{in}^\text{ho}(d,x).P \parallel \text{out}^\text{ho}(d,t).Q\{d/y\}} \mu z.\text{eav}(d,x) \xrightarrow{\tau_e} P\{t/x\} | Q\{d/y\} | \{t/z\}
\]

We can now define both reachability and different equivalence properties in terms of these labelled semantics and relate them to the internal reduction. To define reachability properties in the labelled semantics, we define \(A \parallel_c^s\) to hold when \(A \equiv A', \text{tr} = \text{tr}_1 \text{out}(c,t) \text{tr}_2 \text{and tr}_1 \text{does not bind c for some tr, tr}_1, \text{tr}_2 \in (A \setminus \{\tau\})^s\), term \(t\) and extended process \(A'\).

The following proposition states that any reachability property modelled in terms of frames, called static equivalence \([8]\).

**Proposition 2.** For all closed honest plain processes \(A\), for all \(s \in \{c, e, p\}\), \(A \parallel_c^s\) iff there exists an attacker plain process \(I^s\) such that \(I^s \parallel A \parallel_c^s\).

Next, we define equivalence relations using our labelled semantics that may serve as proof techniques for the may testing relation. First we need to define an indistinguishability relation on frames, called static equivalence \([8]\).

**Definition 3** (Static equivalence \(\sim\)). Two terms \(u\) and \(v\) are equal in the frame \(\phi\), written \((u =_E v)\phi\), if there exists \(n\) and a substitution \(\sigma\) such that \(\phi \equiv n\eta.\sigma\), \(\eta \cap (\text{fn}(u) \cup \text{fn}(v)) = \emptyset\), and \(u\sigma =_E v\sigma\).

Two closed frames \(\phi_1\) and \(\phi_2\) are statically equivalent, written \(\phi_1 \sim \phi_2\), when:

- \(\text{dom}(\phi_1) = \text{dom}(\phi_2)\), and
- for all terms \(u, v\) we have that: \((u =_E v)\phi_1\) if and only if \((u =_E v)\phi_2\).

**Example 4.** Consider the equational theory generated by the equation \(\text{dec}(\text{enc}(x,y), y) = x\). Then we have that

\[
\nu k. \{\text{enc}(a,k)/x_1\} \sim \nu k. \{\text{enc}(b,k)/x_1\} \\
\nu k. \{\text{enc}(a,k)/x_1, k/x_2\} \not\sim \nu k. \{\text{enc}(b,k)/x_1, k/x_2\} \\
\nu k, a. \{\text{enc}(a,k)/x_1, k/x_2\} \sim \nu k, b. \{\text{enc}(b,k)/x_1, k/x_2\}
\]

Intuitively, the first equivalence confirms that encryption hides the plaintext when the decryption key is unknown. The second equivalence does not hold as the test \((\text{dec}(x_1, x_2) =_E a)\) holds on the left hand side, but not on the right hand side. Finally, the third equivalence again holds as two restricted names are indistinguishable.

Now we are ready to define two classical equivalences on processes, based on the labelled semantics: trace equivalence and labelled bisimulation.

**Definition 4** (Trace equivalences \(\approx_t^c, \approx_t^p, \approx_t^e\)). Let \(s \in \{c, p, e\}\). Let \(A\) and \(B\) be two closed honest extended processes. We say that \(A \parallel_t^c B\) if for all \(A \equiv A'\) such that \(\text{bn}(\text{tr}) \cap \text{fn}(B) = \emptyset\), there exists \(B'\) such that \(B \parallel_t^c B'\) and \(\phi(A') \sim \phi(B')\). We say that \(A \approx_t^c B\) when \(A \parallel_t^c B\) and \(B \parallel_t^c A\).
Definition 5. (Labeled bisimulations $\approx^*_l$, $\approx^*_p$, $\approx^*_e$). Let $s \in \{c, p, e\}$. Let $A$ and $B$ two closed honest extended processes such that $\text{dom}(A) = \text{dom}(B)$. We say that $A \approx^*_l B$ if $\approx^*_l$ is the largest equivalence relation such that:

- $\phi(A) \sim \phi(B)$
- $A \xrightarrow{\ell, s} A'$ implies $B \xrightarrow{\ell, s} B'$ and $A' \approx^*_l B'$ for some $B'$,
- $A \xrightarrow{\ell, s} A'$ and $\text{bn}(\ell) \cap \text{fn}(B) = \emptyset$ implies $B \xrightarrow{\ell, s} B'$ and $A' \approx^*_l B'$ for some $B'$.

We again have, as usual that labelled bisimulation implies trace equivalence.

Proposition 3. $\approx^*_l \subseteq \approx^*_m \subseteq \approx^*_e$ for $s \in \{c, e, p\}$.

In [8] it is shown that $\approx^*_e = \approx^*_l$. We conjecture that for the new semantics $p$ and $e$ this same equivalence holds as well. Re-showing these results is beyond the scope of this paper, and we will mainly focus on testing/trace equivalence. As shown in [19], for the classical semantics trace equivalence implies may testing, while the converse does not hold in general. The two relations do however coincide on image-finite processes.

Definition 6. Let $A$ be a closed extended process. $A$ is image-finite for the semantics $s \in \{c, e, p\}$ if for each trace $tr$ the set of equivalence classes $\{\phi(B) \mid A \xrightarrow{\ell, s} B\}/\sim$ is finite.

Note that any replication-free process is necessarily image-finite as there are only a finite number of possible traces for any given sequence of labels $tr$. The same relations among trace equivalence and may testing shown for the classical semantics hold also for the other semantics.

Theorem 1. $\approx^*_l \subseteq \approx^*_m \subseteq \approx^*_e$ on image-finite processes for $s \in \{c, e, p\}$.

The proof of this result (for the classical semantics) is given in [19] and is easily adapted to the other semantics. To see that the implication is strict, we continue Example 2 on processes $A$ and $B$ defined in Figure 2. We already noted that $A \approx^*_m B$, but will now show that $A \not\approx^*_l B$ (for $s \in \{c, e, p\}$). All possible traces of $A$ are of the form $A \xrightarrow{\phi(x_{\text{out}}(c,x))} A'$ where $\phi(A') = \{h^n(a)/x\}$ for $n \in \mathbb{N}$. We easily see that $A \not\approx^*_l B$ as for any $n$ we have that $\{h^n(a)/x\} \not\in \{*/x\}$, by testing $x = h^n(a)$. On the other hand, given an image-finite process, we can only have a finite number of different frames for a given trace, and therefore we can bound the context size that is necessary for distinguishing the processes.

3. Comparing the different semantics

In this section we state our results on comparing these semantics. Results on equivalence comparison are summarized in Figure 4.

We first show that, as expected, all the semantics coincide for reachability properties.

Theorem 2. For all ground, closed honest extended processes $A$, for all channels $d$, we have that $A \parallel^*_d \iff A \parallel^*_e \iff A \parallel^*_d$. 
for all $s \in \{c, p, e\}$
for image finite processes $\approx^s_c = \approx^s_m$
if $s = c$ then $\approx^s_c = \approx^s_0$ (conjectured for $s \in \{p, e\}$)

Fig. 4. Overview of the results.

\[
A \triangleq \nu s_1.\nu s_2.((\text{out}^{ho}(c, s_1), \text{in}^{ho}(c, x), P_1(x)) \parallel (\text{in}^{ho}(c, y), P_2(y)))
\]
\[
B \triangleq \nu s_1.\nu s_2.((\text{out}^{ho}(c, s_1), \text{in}^{ho}(c, x), P_2(x)) \parallel (\text{in}^{ho}(c, y), P_1(y)))
\]

where
\[
P_1(x) \triangleq (\text{if } x = s_1 \text{ then } \text{out}^{ho}(d, s_2)) \parallel (\text{if } x = s_2 \text{ then } \text{out}^{ho}(e, x))
\]
\[
P_2(x) \triangleq (\text{if } x = s_1 \text{ then } \text{out}^{ho}(d, s_2))
\]

To emit on channel $e$, processes $A$ and $B$ must execute $P_2(s_1)$ followed by $P_1(s_2)$. In the classical semantics, a trace of $A$ emitting on $e$ through an internal communication between $\text{out}^{ho}(c, s_1)$ and $\text{in}^{ho}(c, y)$ forces $B$ to execute $P_1(s_1)$ thus preventing it to emit on $e$.

Fig. 5. Processes $A$ and $B$ such that $A \approx^p_c B$ and $A \not\approx_m^e B$.

The next result is, in our opinion, more surprising. As the private semantics force the adversary to observe all information, one might expect that his distinguishing power increases over the classical one. This intuition is however wrong: the classical and private trace equivalences, testing equivalence and labelled bisimulations appear to be incomparable.

**Theorem 3.** $\approx^p_c \not\subseteq \approx^c_r$ and $\approx^c_r \not\subseteq \approx^p_c$ for $r \in \{t, m, \ell\}$.

**Proof.** We show both statements separately.

$\approx^p_c \not\subseteq \approx^c_r$. We first show that there exist $A$ and $B$ such that $A \approx^p_c B$, but $A \not\approx^c_m B$. Note that, as $\approx^s_c \subseteq \approx^s_m$ for $s \in \{c, p\}$ these processes demonstrate both that $\approx^p_c \not\subseteq \approx^c_r$, $\approx^p_c \not\subseteq \approx^c_t$, and $\approx^p_m \not\subseteq \approx^c_m$.

Consider processes $A$ and $B$ defined in Figure 5. In short, the result follows from the fact that if $A$ performs an internal communication on channel $c$ followed by an output on $d$ (from $P_1$), $B$ has no choice other then performing the output on $d$ in $P_2$. In the private semantics, however, the internal communication will be split in an output followed by an input: after the output on $c$, the input $\text{in}^{ho}(c, x), P_2(x)$ following the output becomes available. More precisely, to see that $A \approx^p_c B$ we first observe that if $A \xrightarrow{\text{out}^{ho}(c, \nu z)} p A'$ then $B \xrightarrow{\text{out}^{ho}(c, \nu z)} p B'$ and $A' \equiv B'$, and vice-versa. If $A \xrightarrow{\text{in}(c,t)} p A'$ then $B \xrightarrow{\text{in}(c,t)} p B'$. As $t \not\in \{s_1, s_2\}$ we have that $P_1(t) \approx^p_c 0 \approx^p_c P_2(t)$. Finally, if
Therefore, to complete the proof, we only need to find only first possible action from $B$ may perform an internal communication in the classical semantics to mimic

\[ A \xrightarrow{c} \nu s_1.\nu s_2.(\text{out}^\text{ho}(c, s_1).\text{in}^\text{ho}(c, x).P_1(x)) \approx^p \nu s_1.\nu s_2.(\text{out}^\text{ho}(c, s_1).\text{in}^\text{ho}(c, x).P_2(x)) \]

which allows us to conclude.

As $A$ and $B$ are image-finite, we have that $A \approx^\varepsilon_m B$ if and only if $A \approx^\nu s B$. To see that $A \not\approx^\nu s B$ we observe that $A$ may perform the following transition sequence, starting with an internal communication on a public channel:

\[ A \xrightarrow{\nu z.\text{out}(d,z)} \nu s_1.\nu s_2.(\text{in}(c,z)) \xrightarrow{\text{in}(c,z)} \nu s_1.\nu s_2.(\{s_2/z\}) \]

In order to mimic the behaviour of $A, B$ must perform the same sequence of observable transitions:

\[ B \xrightarrow{\nu z.\text{out}(d,z)\text{ in}(c,z)} \nu s_1.\nu s_2.(\{s_2/z\}) \]

We conclude as $\nu s_1.\nu s_2.(\{s_2/z\}) \approx^r \nu s_1.\nu s_2.(\{\nu z'.\text{out}(c,z')\} \nu s_1.\nu s_2.(\{s_2/z\})$, but $\nu s_1.\nu s_2.(\{s_2/z\}) \not\subseteq \approx^r$. This trace inequivalence has also been shown using DeepSec.

$\not\approx^\approx^r \not\subseteq \not\approx^r$. To show that $\not\approx^r \not\subseteq \not\approx^r$ for $r \in \{\ell, m\}$ we show that there exist processes $A$ and $B$ such that $A \approx^r B$ and $A \not\approx^r B$. As in the first part of the proof, note that, as $\approx^r \subseteq \approx^\leq \subseteq \approx^\varepsilon_m$, for $s \in \{c, p\}$ these processes demonstrate that $\not\approx^r \not\subseteq \not\approx^r$, $\not\approx^r \not\subseteq \not\approx^r$ and $\not\approx^r \not\subseteq \not\approx^r$.

Consider the processes $A$ and $B$ defined in Figure 6. The proof crucially relies on the fact that $B$ may perform an internal communication in the classical semantics to mimic $A$, which becomes visible in the attacker in the private semantics. To see that $A \approx^r B$ we first observe that the only first possible action from $A$ or $B$ is an input. In particular, given a term $t$, there is a unique $B'$ such that $B \xrightarrow{\text{in}(c,t)} B'$ where $B' = \nu s.\text{out}(c, s).\text{out}(d, a) \mid \text{in}(c, y).P(y))$. However, if $A \xrightarrow{\text{in}(c,t)} A'$ then either $A' = B'$ or $A' = A''$ with $A'' \equiv \nu s.\text{in}(c, x).\text{out}(c, s).\text{out}(d, a) \mid P(t)$. Therefore, to complete the proof, we only need to find $B''$ such that $B \xrightarrow{\text{in}(c,t)} B''$ and $A'' \approx^r B''$.

Such a process can be obtained by applying an internal communication on $B'$, i.e. $B \xrightarrow{\text{in}(c,t)} \nu s.\text{out}(d, a) \mid P(s))$. Note that $t \not\in s$ since $s$ is bound, meaning that $P(t) \approx^r \text{out}(d, a)$. Moreover, $P(s) \approx^r \text{in}(c, x).\text{out}(c, s).\text{out}(d, a)$. This allows us to conclude that $\nu s.\text{out}(d, a) \mid P(s) \approx^r A''$.

Again, as $A$ and $B$ are image-finite may and trace equivalence coincide. To see that $A \not\approx^r B$ we first observe that $A$ may perform the following transition sequence:

\[ A \xrightarrow{\text{in}(c,t)} A'' \xrightarrow{t} \nu s.\text{in}(c, x).\text{out}(c, s).\text{out}(d, a) \mid \text{out}(d, a) \]

We conclude as $B \xrightarrow{\text{in}(c,t)} B'$ but $B' \not\approx^r \nu s.\text{in}(c, x).\text{out}(c, s).\text{out}(d, a) \mid \{a/z\}$. Violation of this trace equivalence has also been shown using the DeepSec tool. □
\[
A \doteq \nu s.(\text{in}^{ho}(c,x) \cdot \text{out}^{ho}(c,s) \cdot \text{out}^{ho}(d,a) \mid \text{in}^{ho}(c,y) \cdot P(y)) \\
B \doteq \nu s.(\text{in}^{ho}(c,x) \cdot (\text{out}^{ho}(c,s) \cdot \text{out}^{ho}(d,a) \mid \text{in}^{ho}(c,y) \cdot P(y)))
\]

where

\[
P(y) \doteq \text{if } y = s \text{ then } \text{in}^{ho}(c,z) \cdot \text{out}^{ho}(c,s) \cdot \text{out}^{ho}(d,a) \text{ else } \text{out}^{ho}(d,a)
\]

In the private semantics, a trace of \( A \) starting with the execution of \( \text{in}^{ho}(c,y) \) can only be matched on \( B \) by executing \( \text{in}^{ho}(c,x) \). \( B \) could then emit on channel \( c \), which is not the case for \( A \), hence yielding non equivalence. In the classic semantics, an internal communication between \( \text{out}^{ho}(c,s) \) and \( \text{in}^{ho}(c,y) \) allows to hide the fact that \( B \) can emit on \( c \).

Fig. 6. Processes \( A \) and \( B \) such that \( A \approx_e^p B \) and \( A \not\approx_{m}^p B \).

One may also note that the counter-example witnessing that equivalences in the private semantics do not imply equivalences in the classical semantics is minimal: it does not use function symbols, equational reasoning, private channels, replication nor else branches. The second part of the proof relies on the use of else branches. We can however refine this result in the case of labeled bisimulation to processes without else branches, the counter-example being the same processes \( A \) and \( B \) described in the proof but where we replace each \( \text{out}^{ho}(d,a) \) by 0. In the case of trace equivalence, we can also produce a counter-example without else branches witnessing that trace equivalences in the classical semantics do no imply trace equivalences in the private semantics but provided that we rely on a function symbol \( h \). In the appendix, we describe in more details these processes and give the proofs of them being counter-examples.

Next, we show that the eavesdropping semantics yields strictly stronger bisimilarities, trace and may testing equivalences: the eavesdropping semantics is actually strictly included in the intersection of the classic and private semantics.

**Theorem 4.** \( \approx_e^\ell \subseteq \approx_e^p \cap \approx_e^f \).

**Proof sketch.** We show the result in 3 steps: we show that (1) \( \approx_e^f \subseteq \approx_e^p \), (2) \( \approx_e^f \subseteq \approx_e^\ell \), and (3) that the implication is strict, i.e., there exist \( A, B \) such that \( A \approx_e^p B, A \approx_e^\ell B \) and \( A \not\approx_e^f B \).

(1) We first prove that \( \approx_e^f \subseteq \approx_e^p \). Suppose that \( A \approx_e^f B \). We need to show that for any \( A' \) such that \( A \xin_{tr} B \) there exists \( B' \) such that \( B \xin_{tr} B' \). It follows from the definition of the semantics that whenever \( A \xin_{tr} A' \) then we also have \( A \xin_{trE} A' \) as \( \xi_{tr} \subseteq \xi_{trE} \). As \( A \approx_e^f B \), we have that there exists \( B' \), such that \( B \xin_{trE} B' \) and \( \phi(A') \sim \phi(B') \). As \( tr \) does not contain labels of the form \( eav(c,d) \) nor \( vy.eav(c,y) \) and as no \( \text{COMM-EAV} \) are possible (\( A \) and \( B \) are honest processes) we also have that \( B \xin_{tr} B' \). Hence \( A \approx_e^p B \).

(2) We next prove that \( \approx_e^f \subseteq \approx_e^\ell \). Similar to Item 1 we suppose that \( A \approx_e^f B \) and \( A \xin_{trE} A' \).

From the semantics, we obtain that \( A \xin_{trE} A'_e \), where

- \( \phi(A'_e) \subseteq \phi(A'_e) \), i.e., \( \text{dom}(\phi(A'_e)) \subseteq \text{dom}(\phi(A'_e)) \) and the frames coincide on the common domain.
To emit on channel $e$, processes $A$ and $B$ must execute $P_2(s_1)$ by inputting twice $s_1$ followed by $P_1(s_2)$. In the classical semantics, an internal communication on $A$ between $\text{out}^\text{ho}(c,s_1)$ and $\text{in}^\text{ho}(c,y)$ forces $B$ to execute $P_1(s_1)$ but hides $s_1$, preventing a second input of $s_1$ by $A$. However, in the eavesdropping semantics, the internal communication reveals $s_1$ allowing $A$ to emit on $e$ but not $B$.

Fig. 7. Processes $A$ and $B$ such that $A \approx^\pi_1 B$, $A \approx^\pi_2 B$ but $A \not\approx^\pi_1 B$.

- $tr_e$ is constructed from $tr_e$ by replacing any $\tau$ action resulting from the COMM rule by an application of an eavesdropping rule ($\text{EAV-T}$, $\text{EAV-CH}$, or $\text{EAV-OCH}$).

The proof is done by induction on the length of $tr_e$ and the proof tree of each transition. As $A \approx^\pi_1 B$ we also have that $B \xrightarrow{tr_e} B'_e$ and $A'_e \sim B'_e$. We show by the definition of the semantics that $B \xrightarrow{tr_e} B'_e$ and $\phi(B'_e) \subseteq \phi(B'_e)$ (replacing each eavesdropping action by an internal communication). Due to the inclusions of the frames and $A'_e \sim B'_e$ we also have that $A'_e \sim B'_e$.

(3) Finally we show that the implication $\approx^\pi_1 \subseteq \approx^\pi_2 \cap \approx^\pi_1$ is strict, i.e., there exist $A$ and $B$ such that $A \approx^\pi_1 B$ (which implies $A \approx^\pi_2 B$), $A \approx^\pi_2 B$ (which implies $A \approx^\pi_1 B$) but $A \not\approx^\pi_1 B$.

Consider the processes $A$ and $B$ defined in Figure 7. This example is a variant of the one given in Figure 5. The difference is the addition of "\text{in}^\text{ho}(d,z)\), if $z = s_1$ then " in processes $P_1(x)$ and $P_2(x)$: this additional check is used to verify whether the adversary learned $s_1$ or not. The proofs that $A \approx^\pi_1 B$ and $A \approx^\pi_2 B$ follow the same lines as in Theorem 3. We just additionally observe that $\nu s_1.(\text{in}^\text{ho}(d,z), \text{if } z = s_1 \text{ then } \text{out}^\text{ho}(d,s_2)) \approx^\pi_1 \nu s_1.(\text{in}^\text{ho}(d,z), 0)$ for $s \in \{c,p\}$.

The trace witnessing that $A \not\approx^\pi_1 B$ is again similar to the one in Theorem 3, but starting with an eavesdropping transition which allows the attacker to learn $s_1$, which in turn allows him to learn $s_2$ and distinguish $P_1(s_2)$ from $P_2(s_2)$. These trace (in)equivalences have also been verified using DeepSec. \[\square\]

We note from the processes defined in Figure 7 that the implications are strict even for processes that do not communicate on private channels, do not use replication, nor else branches and terms are simply names (no function symbols nor equational theories).

**Theorem 5.** $\approx^\pi_1 \subseteq \approx^\pi_2 \cap \approx^\pi_1$.

**Proof sketch.** The proof is structured in 3 steps, as in the proof of Theorem 4.

(1) We first show that $\approx^\pi_1 \subseteq \approx^\pi_2$. Suppose $A \approx^\pi_1 B$ and let $R$ be the relation witnessing this equivalence. We will show that $R$ is also a labelled bisimulation in the private semantics. Suppose $A \equiv R B$.
as \( A \approx^\circ_B \), we have that \( \phi(A) \sim \phi(B) \).

- if \( A \leftarrow_B A' \) then, as \( A \leftarrow_B A' \), as \( A \approx_B^\circ \) there exists \( B' \) such that \( B \approx_B^\circ B' \).

As \( B \) is a honest process no Comm-EAV transition is possible, and hence \( B \approx_B^\circ B' \).

- if \( A \leftarrow_B A' \) and \( bn(\ell) \cap fn(B) = \emptyset \) then we also have that \( A \leftarrow_B A' \) as \( A \leftarrow_B A' \) and there exists \( B' \) such that \( B \approx_B^\circ B' \) and \( A' \approx_B^\circ \) \( B' \). As no Comm-EAV are possible and \( \ell \) is not of the form \( eva(c, d) \) nor \( v y.eva(c, y) \) we have that \( B \approx_B B' \).

(2) We next show that \( \approx_B^\circ \subseteq \approx_B^\circ \). We will show that \( \approx_B^\circ \) is also a labelled bisimulation in the classical semantics. The proof relies on similar arguments as in Item 2 of the proof of Theorem 4 and the facts that

\[
\nuu(A' \mid \{i/x\}) \approx_B^\circ \nuu(B' \mid \{v/x\}) \implies \nuu(A') \approx_B^\circ \nuu(B'),
\]

\[
A' \approx_B^\circ B' \implies \nu c. A' \approx_B^\circ \nu c. B'.
\]

The first property is needed when an internal communication of a term or public channel is replaced by an eavesdrop action and an input. The second property handles the case when we replace the internal communication of a private channel by an application of the EAV-OCH rule and an input.

(3) To show that the implication \( \approx_B^\circ \subseteq \approx_B^\circ \subseteq \approx_B^\circ \) is strict, we exhibit processes \( A \) and \( B \) such that \( A \approx_B^\circ B \), \( A \approx_B^\circ B \) but \( A \not\approx_B^\circ B \) (which implies \( A \not\approx_B^\circ B \)). The processes defined in Figure 7 witness this fact (cf. the discussion of these processes in the proof of Theorem 4).

Again we note that the implications are strict, even for processes containing only public channels.

**Theorem 6.** \( \approx_B^\circ \subseteq \approx_B^\circ \subseteq \approx_B^\circ \).

**Proof sketch.** The proof is structured in 3 parts, as for Theorems 5 and 4.

(1) We first prove that \( \approx_B^\circ \subseteq \approx_B^\circ \). Suppose that \( A \approx_B^\circ B \). Suppose that \( A \approx_B^\circ B \). We need to show that for all channel \( c \), for all \( C[\_] \) attacker evaluation contexts \( \leftarrow_B \) closing for \( A \) and \( B \), \( C[A] \downarrow^\circ_B \) is equivalent to \( C[B] \downarrow^\circ_B \). It follows from the definition of the private semantics that any process \( \mathsf{eva}(c, x).P \) in \( C[\_] \) has the same behaviour as the process \( 0 \). Hence, we generate a context \( C[\_] \) by replacing in \( C[\_] \) any instance of \( \mathsf{eva}(c, x).P \) by \( 0 \), and thus obtaining \( C[A] \downarrow^\circ_B \leftrightarrow C[A] \downarrow^\circ_B \leftrightarrow C[B] \downarrow^\circ_B \leftrightarrow C[B] \downarrow^\circ_B \). Notice that the definition of semantics gives us \( \leftarrow_B \subseteq \leftarrow_B \). Hence, \( C[A] \downarrow^\circ_B \) implies \( C'[A] \downarrow^\circ_B \) and \( C'[B] \downarrow^\circ_B \) implies \( C'[B] \downarrow^\circ_B \). Furthermore, since we built \( C'[\_] \) to not contain any process of the form \( \mathsf{eva}(c, x).P \), we deduce that rules C-EAV and C-OEAV can never be applied in a derivation of \( C'[A] \) or \( C'[B] \). It implies that \( C'[A] \downarrow^\circ_B \leftrightarrow C'[A] \downarrow^\circ_B \leftrightarrow C'[B] \downarrow^\circ_B \leftrightarrow C'[B] \downarrow^\circ_B \). Thanks to \( A \approx_B^\circ B \), we know that \( C'[A] \downarrow^\circ_B \leftrightarrow C'[B] \downarrow^\circ_B \) and so we conclude that \( C[A] \downarrow^\circ_B \leftrightarrow C'[B] \downarrow^\circ_B \).

(2) We next prove that \( \approx_B^\circ \subseteq \approx_B^\circ \). Similarly to Item 1, we consider a channel \( c \) and an attacker evaluation context \( C[\_] \) that is \( c \) closing for \( A \) and \( B \). The main difficulty of this proof is to match the application of the rule Comm in the classical semantics with the rules C-EAV and C-OEAC. However, \( C[\_] \) does not necessarily contain eavesdrop process \( \mathsf{eva}(d, x) \mid \omega c \). Moreover, as mentioned in Item 1, a process \( \mathsf{eva}(d, x).P \) has the same behaviour as \( 0 \) in the classical semantics but can have a completely different behaviour in the eavesdropping semantics if \( P \) is not \( 0 \). Thus, we remove from \( C[\_] \) the eavesdrop
processes, obtaining \( C'[\_] \). Then, we define a new context \( C''[\_] \) based on \( C'[\_] \) where
will add harmless eavesdrop process \( \text{eav}(d, y) \cdot 0 \). We first add in parallel the processes
\( \text{leav}(a, y) \mid \omega a \) for all free channels \( a \) in \( C'[\_] \), \( A \) and \( B \). Moreover, since private chan-
nels can be opened, we also replace any process \( \nu d.P, \text{in}^{\text{at}}(c, x).P \) where \( d, x \) are of channel
type with \( \nu d.(P \mid \text{leav}(d, y)) \) and \( \text{in}^{\text{at}}(c, x).(P \mid \text{leav}(x, y)) \). By induction of the derivations,
we can show that \( C[A] \downarrow^e_c \iff C''[A] \downarrow^e_c \) and \( C[B] \downarrow^e_c \iff C''[B] \downarrow^e_c \). Since \( A \approx^e_m B \), we deduce
that \( C''[A] \downarrow^e_c \iff C''[B] \downarrow^e_c \) and so \( C[A] \downarrow^e_c \iff C[B] \downarrow^e_c \).

(3) Finally we show that the implication \( \approx^e_m \subseteq \approx^p_m \cap \approx^e_m \) is strict, i.e., there exist processes
\( A \) and \( B \) such that \( A \approx^e_m B, A \approx^p_m B \) but \( A \not\approx^e_m B \). The processes defined in Figure 7
witness this fact. They already were witness of the strict inclusion \( \approx^e_m \subseteq \approx^p_m \cap \approx^e_m \) (see proof
of Theorem 4) and since \( A \) and \( B \) are image finite, we know from Theorem 1 that may and
trace equivalences between \( A \) and \( B \) coincide. \( \square \)

4. Subclasses of processes for which (some of) the semantics coincide

As illustrated in previous sections, the presence of internal communications between public
channels is the main issue when comparing the different semantics. Thus, the most natural class
of processes on which the semantics coincide are processes where no internal communication on a
public channel is possible.

**Definition 7.** Let \( P \) be a closed honest process. \( P \) is internal communication free if and only for
all \( P \xrightarrow{\tau} P' \xrightarrow{\tau} P'' \), the \( \tau \) action in \( P' \xrightarrow{\tau} P'' \) is not the application of the rule \( \text{COMM} \).
We denote by \( \text{ICF} \) the set of internal communication free processes.

**Lemma 1.** When restricted to \( \text{ICF} \), \( \approx^e_{s_1} = \approx^e_{s_2} \) for \( r \in \{ t, o, m, t \} \) and \( s_1, s_2 \in \{ c, p, e \} \).

**Proof.** Immediate from the semantics. \( \square \)

However, this class is very restrictive as it prevents any process of the form \( \text{out}^{ho}(c, u).P \mid \text{in}^{bo}(c, x).Q(x) \). Therefore, we study in the rest of this section alternate classes of processes. The
class of processes we study are mainly related to the notion of determinism: we first study the class
of determinate processes, denoted \( \mathcal{D} \), and then mainly restrict our attention to the case when the
number of sessions is bounded. This is motivated by the fact that most tools able to verify these
equivalences are restricted to a bounded number of sessions. We study three increasingly restric-
tive classes: (i) bounded determinate processes (denoted \( \mathcal{BD} \)), (ii) action-determinate processes
(denoted \( \mathcal{AD} \)), and (iii) strong action determinate processes (denoted \( \mathcal{SAD} \)). As the definition of
determinism depends on the semantics, we may add the semantics as a parameter, e.g., we write
\( \mathcal{D}(e) \) for the class of processes that are determinate in the eavesdrop semantics. Figure 8 provides
an overview of the results of this section.

Finally, we also identify a new syntactic subclass of processes, called \( I/O\)-unambiguous and
show relations among the equivalences for different semantics.
4.1. Determinate processes

4.1.1. Defining classes of determinate processes and their relations

In this section we define a subclass of determinate processes. These subclasses however depend on the semantics and therefore we also study the relations between these different subclasses. The notion of determinacy was defined in [19] for the classical semantics. It was shown that for determinate processes, observational and trace equivalence coincide. Intuitively, on determinate processes the attacker can determine, at each step of the execution, the position of the executed action in the process tree: this means that either the labels leading to the executed action differ, or, in case of two identical sequence of labels, the frames may be distinguished.

Definition 8 (determinacy). Let \( s \in \{ c, p, e \} \). Let \( \cong \) be an equivalence relation on closed honest extended processes. A closed honest extended process \( A \) is \( \cong \)-determinate if whenever \( A \overset{s}{\rightarrow} B, A \overset{\ell}{\rightarrow} B' \) and \( \phi(B) \sim \phi(B') \) then \( B \cong B' \).

We denote by \( D(s, \cong) \) the set of closed honest extended processes that are \( \cong \)-determinate.

It was shown in [19] that \( D(c, \cong_{c}^{s}) = D(c, \cong_{c}^{t}) \). We can show that this equality also holds in the private and eavesdrop semantics.

Lemma 2. For all \( s \in \{ c, p, e \} \), \( D(s, \cong_{s}^{t}) = D(s, \cong_{s}^{t}) \).

Proof. The proof of [19, Lemma 2] literally holds for all semantics. □

Thanks to the previous lemma, we may simply consider the set of \( s \)-determinate processes, denoted \( D(s) \), as the set of closed honest extended processes that are \( \cong_{s}^{t} \)-determinate or \( \cong_{s}^{t} \)-determinate coincide. It was also shown in [19] that when restricted to \( c \)-determinate processes,
Proof sketch. We sketch the proof here. A more detailed version is available in Appendix F.

We have that $\approx_P = \approx_{\ell}$. Once again, this result directly extends to $p$-determinate and $e$-determinate processes.

**Lemma 3.** When restricted to $D(s)$, $\approx_P = \approx_{s}$ for $s \in \{c, p, e\}$.

**Proof.** The proof of [19, Theorem 2] literally holds for all semantics.

The notion of determinacy depends on equivalences. Therefore one might expect the relations between determinate processes for different semantics to follow similar results as for the equivalences. However, we show that the sets of determinate processes coincide for eavesdrop and private semantics, while they are incomparable to the classic semantics.

**Lemma 4.** $D(p) = D(e)$, $D(c) \subseteq D(p)$ and $D(p) \nsubseteq D(c)$.

**Proof sketch.** We sketch the proof here. A more detailed version is available in Appendix F.

We start by showing that $D(p) \nsubseteq D(c)$. Consider the process $A$ displayed in Figure 9a. $A \in D(c)$ since $A \rightarrow c \text{ out}^{ho}(c,a)$ by the rule $\text{COMM}$ and $\text{out}^{ho}(c,a) \notin \approx_{\ell} A$. Moreover, $A \in D(p)$ since for all $\text{tr}$, there is a unique $A'$ such that $A \rightarrow_{\text{tr}} p A'$. Hence $D(p) \nsubseteq D(c)$.

We now show that $D(c) \nsubseteq D(p)$. Consider the process $B$ displayed in Figure 9b. Intuitively, the use of the private channel $s$ in $B$ encodes a non determinist choice between the two processes $P$ and $Q$. We can show that $P,Q \in D(c)$ which allows us to deduce that $B \in D(c)$. However, $B \rightarrow_{r} p P$, $B \rightarrow_{r} p Q$ and $P \notin \approx_{r} Q$ imply $B \notin D(p)$.

Let us show $D(e) \subseteq D(p)$. Consider an honest closed process $A$ such that $A \in D(e)$. Let $A \rightarrow_{r} p A_1$ and $A \rightarrow_{r} p A_2$. By definition of the semantics, $A \rightarrow_{r} p A_i$ implies $A \rightarrow_{e} A_i$, for $i = 1, 2$. Since $A \in D(e)$, we deduce $A_1 \approx_{e} A_2$. By applying Theorem 5, we obtain $A_1 \approx_{r} A_2$ which concludes the proof of $D(e) \subseteq D(p)$.

Finally, we need to show that $D(p) \subseteq D(e)$. This part of the proof is detailed in Appendix F.

**4.1.2. Relations between semantics for determinate processes**

We will now compare the equivalences when we restrict the processes to $D(s)$ for a given semantics $s$. We start with the subclass $D(p)$ (which coincides with $D(e)$).

**Theorem 7.** When restricted to $D(p)$, we have $\approx_{r_1} \subseteq \approx_{r_2} \subseteq \approx_{r_3}$ for $s_1, s_2 \in \{p, e\}$, $r_1, r_2, r_3 \in \{\ell, t\}$. 
The proof is given in Appendix F.

Next, we consider the subclass \( D(c) \). We show that even for this subclass of processes, the equivalences for the classical semantics are not included in the other ones.

**Lemma 5.** When restricted to \( D(c) \), we have \( \approx_c^e \not\subseteq \approx_c^p \) for \( s \in \{p, e\} \) and \( r \in \{\ell, t\} \).

**Proof.** In the proof of Theorem 7 we showed that the processes \( P \) and \( Q \) displayed in Figure 9b satisfy the following properties: \( P, Q \in D(c) \), \( P \approx_c^e Q \) and \( P \not\approx_c^p Q \). Note that we also have \( P \not\approx_c^e Q \). Hence \( P \) and \( Q \) allow us to prove that \( \approx_c^e \not\subseteq \approx_c^p \) for \( s \in \{p, e\} \) and \( r \in \{\ell, t\} \). \( \square \)

### 4.2. Determinacy for bounded processes

As many verification tools [14–18] consider a bounded number of sessions we study in this section, notions of determinacy when restricted to processes without replication. We also consider the notion of action-determinate which was introduced in [20] as a subclass of determinate processes that enable partial order reductions that significantly speed-up verification. Finally, we discuss the notion of strong action-determinate processes: this class was introduced in several tools, as this property can easily be checked syntactically. Interestingly, for this class of action-determinate processes, all notions of equivalences and semantics coincide.

#### 4.2.1. Bounded determinate processes

We investigate in this section whether additional relations hold between the semantics when restricted to such bounded processes, i.e., processes without replication. In particular we show that when restricted to such bounded processes, a c-determinate \( P \) cannot have internal communication. However, we also show that even when restricted to bounded processes, \( \approx_c^p \) and \( \approx_c^e \) do not coincide.

We denote by \( BD(s) \) the set of bounded processes in \( D(s) \) for \( s \in \{p, c, e\} \).

**Lemma 6.** \( BD(c) \subseteq BD(p) = BD(e) \) and \( BD(c) \subseteq ICF \).

**Proof.** Note that Lemma 4 directly gives us \( BD(p) = BD(e) \). Moreover, consider the process \( A \) displayed in Figure 9a. We already showed in Lemma 4 that \( A \in D(p) \) and \( A \not\in D(c) \). Since \( A \) does not contain a replication, we deduce that \( BD(p) \not\subseteq BD(c) \).

Let us now show that \( BD(c) \subseteq ICF \). Let \( A \in BD(c) \). Assume by contradiction that \( A \approx_c A_1 \not\approx_c \approx_c A_2 \) where the transition \( A_1 \approx_c \approx_c A_2 \) is the application of the rule COMM. Hence \( A_1 \equiv \nu\mu.(\text{out}^\mu(c, u).P | \text{in}^\mu(c, x).Q | R) \) and \( A_2 = \nu\mu.(P | Q \{a/x \} | R) \) for some \( c, u, P, Q \). Since \( A \in BD(c) \), we deduce that \( A_1 \approx_c^p A_2 \). Consider the maximal trace \( tr_m \) of \( A_1 \) (the trace \( tr_m \) exists since \( A \) is bounded), i.e. \( A_1 \not\approx_c A_2 \). Since \( A_1 \approx_c^p A_2 \), the trace \( tr_m \) is also maximal for \( A_2 \) with \( A_2 \not\approx_c A_2 \). But \( \nu\mu.(\text{out}^\mu(c, z).\text{in}(c, z).P | Q \{a/z \} | R | \{a/z \}) \) and \( \nu\mu.(P | Q \{a/z \} | R | \{a/z \}) \) are not \( \approx_c A_2 \) for some \( A_2 \) and so \( \nu\mu.(\text{out}^\mu(c, z).\text{in}(c, z).P | Q \{a/z \} | R | \{a/z \}) \) is a trace of \( A_1 \) which contradicts the maximality of \( tr_m \). Hence \( BD(c) \subseteq ICF \).

To see that the inclusion \( BD(c) \subseteq ICF \) is strict, observe that for the process

\[
P = \text{out}^\mu(c, a).\text{out}^\mu(c, a_1) | \text{out}^\mu(c, a).\text{out}^\mu(c, a_2)
\]
When restricted to Theorem 8.

coincides with any (non-trivial) replicated process violates action-determinism. 

verification tools such as verification by several orders of magnitude. Such techniques have been implemented in several 

Action-determinate.

The proof is given in Appendix G.

we have that $P \in \text{ICF}$, but $P \not\in \mathcal{BD}(c)$. 

Finally, let us prove $\mathcal{BD}(c) \subseteq \mathcal{BD}(p)$. Let $A \in \mathcal{BD}(c)$, $A \approx_p A_1$ and $A \approx_p A_2$. Note that 

$A \approx p A_i$ implies $A \approx c A_i$, for $i = 1, 2$. As $A \in \mathcal{BD}(c)$, $A_1 \approx c A_2$. As $\mathcal{BD}(c) \subseteq \text{ICF}$, $A \in \text{ICF}$ and so $A_1, A_2 \in \text{ICF}$. By Lemma 1, we obtain $A_1 \approx c A_2$ which allows us to conclude. □

In particular, as $\mathcal{BD}(c) \subset \text{ICF}$ we directly have that all semantics coincide (Lemma 1), and by 

determinacy all equivalences coincide as well (Lemma 3).

Corollary 1. When restricted to $\mathcal{BD}(c)$, we have that $\approx_{r_1} = \approx_{r_2}$ for $r_1, r_2 \in \{\ell, o, m, t\}$ and 

$s_1, s_2 \in \{c, p, e\}$.

We next investigate the relations when processes are restricted to the subclass $\mathcal{BD}(p)$ (which 

coincides with $\mathcal{BD}(e)$).

Theorem 8. When restricted to $\mathcal{BD}(p)$, we have that $\approx_p = \approx_r \subseteq \approx_r$ for $r \in \{\ell, t\}$.

The proof is given in Appendix G.

Action-determinate. As mentioned above, the class of action determinate processes is of interest 

for verification tools since it supports partial order reduction techniques [20] which speed-up verification by several orders of magnitude. Such techniques have been implemented in several 

verification tools such as APTE, AKISS and DeepSec.

Definition 9. Let $P$ be a closed honest process. We say that $P$ is action-determinate if $\text{bn}(P) \cap \text{Ch} = $\emptyset$ and for all $P \approx c P'$, $P' \not\approx c (\text{out}^{ho}(c, u_1)).Q_1 \mid \text{out}^{ho}(c, u_2).Q_2(Q_3)$ and $P' \not\approx c (\text{in}^{ho}(c, x_1)).Q_1 \mid \text{in}^{ho}(c, x_2).Q_2(Q_3)$ for all $\hat{k}, c, u_1, u_2, x_1, x_2, Q_1, Q_2, Q_3$.

We define $\mathcal{AD}$ to be the set of all action determinate processes.

Intuitively, a process is action-determinate when there are never two similar available actions, 

i.e. two inputs or two outputs on the same channel. Note in particular that any (non-trivial) 

replicated process violates action-determinism.
We first show that the class of action determinate processes is strictly included in the class of
determinate processes (for the private and, hence, eavesdropping semantics).

Lemma 7. \( AD \subseteq BD(p) \).

The proof is given in Appendix H.

We can now show that the relations among equivalences that did hold for the subclass \( BD(p) \)
(Theorem 8) do also hold for the subclass \( AD \).

Theorem 9. When restricted to \( AD \), we have that \( \approx_r^p = \approx_r^e \subseteq \approx_e^r \) for \( r \in \{ \ell, t \} \).

Note that, while the equality and inclusion is a direct corollary of Theorem 8 and Lemma 14,
the fact that the inclusion is strict needs to be shown. The proof is given in Appendix I.

Strong action determinate. In the context of automated verification, deciding whether a process
is action-determinate is still rather costly as it basically requires to verify a reachability prop-
erty. We therefore introduce a stronger and more syntactical notion of action determinate, which
is actually implemented in the verification tools \textsc{AkiSS} and \textsc{DeepSec}. Intuitively, while action
determinate processes never reach a situation where two “similar” actions are available, strong
action-determinate processes verify that such similar actions never appear in parallel, syntacti-
cally.

We first define the set of action skeletons \( S = \{ \text{out}(c), \text{in}(c) \mid c \in Ch \} \).

Definition 10 (strong action determinate). The set \( Sa(S) \) built on \( S \subseteq S \) is the smallest set of
honest processes such that \( \{u/x\}, 0 \in Sa(\emptyset) \) for all \( u, x \) and such that if \( P \in Sa(S) \) and \( Q \in Sa(S') \)
then

- \( \text{out}^{ho}(c,u).P \in Sa(\{\text{out}(c)\} \cup S) \) when \( c \in Ch \)
- \( \text{in}^{ho}(c,x).P \in Sa(\{\text{in}(c)\} \cup S) \) when \( c \in Ch \)
- \( \nu k; P \in Sa(S) \) when \( \{\text{in}(k), \text{out}(k)\} \cap S = \emptyset \)
- if \( u = v \) then \( P \) else \( Q \in Sa(S \cup S') \)
- \( Sa(P \mid Q) \in Sa(S \cup S') \) when \( S \cap S' = \emptyset \)

We define the set of strong action determinate process as \( SaD = \cup_{S \subseteq S} Sa(S) \).

As the name indicates, it is easy to see that any strong action determinate process is also an
action determinate process.

Lemma 8. \( SaD \subseteq AD \).

Proof. The implication follows directly from the definition, as any strongly action determinate
process forbids two identic skeletons in parallel. To see that the implication is strict we observe
that, for

\[
P \doteq \nu k \cdot (\text{out}^{ho}(c,k) \mid \text{in}^{ho}(c,x) . \text{if } x = k \text{ then } \text{out}^{ho}(c,u))
\]

we have that \( P \in AD \), but \( P \notin SaD \). \( \square \)
While for action determinate processes we have that $\approx^p_r \subseteq \approx^c_r$, we can show that for strong action determinate processes we actually have $\approx^c_r \subseteq \approx^p_r$.

**Theorem 10.** When restricted to $\text{SAD}$, we have $\approx^c_r \subseteq \approx^p_r$.

Proof of Theorem 10 can be found in Appendix J.

This implies the following corollary stating that for strong action determinate processes, all semantics and equivalences coincide. This is particularly interesting as the AKISS and DeepSec tools check this condition. Moreover, it means that partial-order reduction optimizations, developed and shown correct for the private semantics [20], are correctly applied by these tools, regardless of the chosen semantics.

**Corollary 2.** When restricted to $\text{SAD}$, we have that $\approx^r_{s_1} \subseteq \approx^r_{s_2}$ for $r_1, r_2 \in \{\ell, o, m, t\}$ and $s_1, s_2 \in \{c, p, e\}$.

### 4.3. I/O-unambiguous processes

Restricting processes to action-determinate processes may sometimes be too restrictive. For instance, when verifying unlinkability and anonymity properties, two outputs by different parties should not be distinguishable due to the channel name. We therefore introduce another class of processes, that we call I/O-unambiguous for which we also show that the different semantics (although not the different equivalences) do coincide.

Intuitively, an i/o-unambiguous process forbids an output and input on the same public channel to follow each other directly (or possibly with only conditionals in between). For instance, we forbid processes of the form $\text{out}^\theta(c, t).\text{in}^\theta(c, x).P$, $\text{out}^\theta(c, t).\text{in}^\theta(c, x).P | Q$ as well as $\text{out}^\theta(c, t).\text{if} \quad t_1 = t_2 \quad \text{then} \quad P \quad \text{else} \quad \text{in}^\theta(c, x).Q$. We however allow inputs and outputs on the same channel in parallel.

**Definition 11.** We define an honest extended process $A$ to be I/O-unambiguous when $\text{ioua}(A, \_ ) = \top$ where

\[
\begin{align*}
\text{ioua}(0, c) &= \top & \text{ioua}(\{u/x\}, c) &= \top & \text{ioua}(P, c) &= \text{ioua}(P, c) \\
\text{ioua}(A \mid B, c) &= \text{ioua}(A, c) \land \text{ioua}(B, c) & \text{ioua}(\nu x. A, c) &= \text{ioua}(A, c) \\
\text{ioua}(\nu n. A, c) &= \begin{cases} \\
\bot \quad & \text{if} \ n \in \text{Ch} \\
\text{ioua}(A, c) \quad & \text{otherwise} \\
\end{cases} \\
\text{ioua}(\text{if} \quad u = v \quad \text{then} \quad P \quad \text{else} \quad Q, c) &= \text{ioua}(P, c) \land \text{ioua}(Q, c) \\
\text{ioua}(\text{out}^\theta(d, u).P, c) &= \begin{cases} \\
\bot \quad & \text{if} \ d \text{ is of channel type} \\
\text{ioua}(P, d) \quad & \text{otherwise} \\
\end{cases} \\
\text{ioua}(\text{in}^\theta(d, x).P, c) &= \begin{cases} \\
\bot \quad & \text{if} \ x \text{ is of channel type or } d = c \\
\text{ioua}(P, \_ ) \quad & \text{otherwise} \\
\end{cases}
\end{align*}
\]

Note that an I/O-unambiguous process does not contain private channels and always input/output base-type terms. We also note that a simple way to enforce that processes are I/O-unambiguous is to use disjoint channel names for inputs and outputs (at least in the same parallel thread).

**Theorem 11.** When restricted to I/O-unambiguous processes, we have that $\approx^p_r = \approx^c_r$ but $\approx^c_r \subseteq \approx^p_r$ for $r \in \{\ell, t\}$.
Proof. From Theorems 5 and 4, we already know that \( \approx_{\ell} \subseteq \approx_{r} \) and \( \approx_{r} \subseteq \approx_{p} \). Hence, we only need to show that \( \approx_{p} \subseteq \approx_{r} \) and \( \approx_{r} \subseteq \approx_{\ell} \). The latter is easily shown by noticing that the processes \( A \) and \( B \) in Figure 6 are \( I/O \)-unambiguous. Thus, we focus on \( \approx_{p} \subseteq \approx_{r} \).

We start by proving that for all \( I/O \)-unambiguous processes \( A \), for all \( A \xrightarrow{tr} A' \), we have that \( A' \) is \( I/O \)-unambiguous. Note that structural equivalence preserves \( I/O \)-unambiguity, i.e., for all extended processes \( A, B \), for all channel name \( c \), \( A \equiv B \) implies \( \text{ioua}(A,c) = \text{ioua}(B,c) \). Hence, we assume w.l.o.g. that a name is bound at most once and the set of bound and free names are disjoint.

Second, we show that for all \( I/O \)-unambiguous processes \( A \), for all \( A \xrightarrow{\nu z.out(c,z).in(c,z)} B' \), we have that \( \nu z.eav(c,z) \xrightarrow{\nu z.eav(c,z)} B' \). To prove this property, denoted \( P \), let us assume w.l.o.g. that \( A \xrightarrow{\nu z.out(c,z)} B \), \( A_1 \xrightarrow{\nu z.out(c,z)} A_2 \). The transition \( A \xrightarrow{\nu z.eav(c,z)} B \) indicates that \( A \equiv \nu \bar{n}.(\text{out}_{ho}(c,u).P \mid Q) \) and \( A_1 \equiv \bar{n}.(P \mid Q \mid \{^u/z\}) \) for some \( P,Q,\bar{n},c,u \). Note that \( A \) is \( I/O \)-unambiguous, and hence \( \text{ioua}(P,c) = \top \).

As \( A \) is \( I/O \)-unambiguous implies that \( A \) does not contain private channels, we have that the rule applied in \( A_1 \rightarrow^{\nu z.eav(c,z)} A_2 \) is either the rule THEN or ELSE. Therefore, there exists \( P' \) and \( Q' \) such that \( P \rightarrow^{\nu z.eav(c,z)} P', Q \rightarrow^{\nu z.eav(c,z)} Q' \), \( A_n \equiv \nu \bar{n}.(P' \mid Q' \mid \{^u/z\}) \) and \( \text{ioua}(P',c) = \top \). Hence, we deduce that there exists \( Q_1, Q_2 \) such that \( Q' \equiv \nu \bar{n}.(\text{in}(c,x)Q_1 \mid Q_2) \) and \( A' \equiv \nu \bar{n}.\nu \bar{n}.(P' \mid Q_1 \{^u/z\} \mid Q_2) \). We conclude the proof of this property by noticing that we can first apply on \( A \) the reduction rules of \( Q \rightarrow^{\nu z.eav(c,z)} Q' \), then apply the rule C-EAV and finally apply the rules of \( P \rightarrow^{\nu z.eav(c,z)} P' \).

(1) To prove \( \approx_{p} \subseteq \approx_{r} \), we assume that \( A,B \) are two closed honest extended processes such that \( A \approx_{r} B \). For all \( A \xrightarrow{tr} A' \), it follows from the semantics that \( A \xrightarrow{tr} A' \) and \( \text{tr} \) is obtained by replacing in \( \text{tr} \) each \( \nu z.eav(c,z) \) by \( \nu z.out(c,z).in(c,z) \). Since \( A \approx_{r} B \), there exists \( B' \) such that \( B \xrightarrow{tr} B' \) and \( \phi(A') \sim \phi(B') \). Thanks to the property \( P \), we conclude that \( B \xrightarrow{tr} B' \).

(2) To prove \( \approx_{p} \subseteq \approx_{\ell} \), we assume that \( A,B \) are two closed honest extended processes such that \( A \approx_{p} B \) and \( \mathcal{R} \) be the relation witnessing this equivalence. We will show that \( \mathcal{R} \) is also a labelled bisimulation in the eavesdropping semantics. Suppose \( AB \).

- As \( A \approx_{p} B \), we have that \( \phi(A) \sim \phi(B) \).
- If \( A \xrightarrow{\nu z.eav(c,z)} A' \), then, as \( A \) is honest, \( A \xrightarrow{\nu z.eav(c,z)} A' \). As \( A \approx_{p} B \) there exists \( B' \) such that \( \nu z.eav(c,z) \xrightarrow{\nu z.eav(c,z)} B' \) and \( A' \xrightarrow{\nu z.eav(c,z)} B' \).
- If \( A \xrightarrow{\nu z.out(c,z).in(c,z)} A' \) then, as \( A \) is \( I/O \)-unambiguous, \( A \xrightarrow{\nu z.out(c,z).in(c,z)} A' \) where \( \text{tr} = \nu z.out(c,z).in(c,z) \) when \( \ell = \nu z.eav(c,z) \) else \( \text{tr} = \ell \). As \( A \approx_{p} B \), there exists \( B' \) such that \( B \xrightarrow{\nu z.eav(c,z)} B' \) and \( A' \xrightarrow{\nu z.eav(c,z)} B' \).

5. Different semantics in practice

As we have seen, in general, the three proposed semantics may yield different results. A conservative approach would consist in verifying always the eavesdropping semantics which is stronger than the two other ones, as shown before. However, this semantics seems also to be the least efficient
one to verify. Moreover, partial-order reduction techniques that provide tremendous speed-ups were only developed for the private semantics.

We have implemented the three different semantics in the DeepSec tool. This allowed us to investigate the difference in results and performance between the semantics. In our experiments we considered several examples from DeepSec’s example repository. We rely on the existing modelling and do not describe these protocols, as these details are not important for the observations we wish to make. All benchmarks were carried out on a machine with 20 Intel Xeon 3.10GHz cores and 50 Gb of memory. The implementation and the specification files are available at [23].

The specifications we used for these experiments include verification of

- strong secrecy in several classical authentication protocols (Denning-Sacco, Needham-Schroeder-Lowe (NSL), Wide Mouth Frog, and Yahalom-Lowe protocols);
- anonymity of the Private Authentication protocol proposed by Abadi and Fournet [24];
- anonymity and unlinkability of the passive authentication protocol implemented in the European Passport protocol [25, 26];
- unlinkability of the AKA protocol, deployed in 3G mobile telephony [27];
- vote privacy in the Helios e-voting protocol [28], and the e-voting protocol proposed by Scytl for elections in the Swiss Neuchâtel canton [29].

For all these examples we found that the results were unchanged, independent of the semantics. However, as expected, performance was generally better for the private semantics, and much better for strong action determinate processes, as this class allows for powerful partial-order reductions. The existing protocol encodings generally used a single public channel. To enforce membership in a particular subclass we had to use different channel names. Surprisingly, the use of distinct channels to enforce I/O-unambiguity, significantly enhances the tool’s performance. We could not make the voting protocols I/O unambiguous and action determinate, because the encodings use private channels. In the absence of private channels using different channel names to enhance efficiency is tempting. One must however be careful as changing channel names changes the attacker’s observation and may change the result. Typically, a single channel name models that the attacker does a priori not know which process sent a given message. Binding the channel name to the identity allows the attacker to know which host sent a message, but not necessarily which of the possibly multiple processes, e.g. sessions, on the given host. However, this modelling is typically not adequate when checking anonymity, or unlinkability, as it reveals the sender’s identity. For such properties, it may be possible to use different channel names for each session, modelling that the adversary can distinguish different sessions, but not necessarily whether the same host executed one or several sessions. This is the encoding we used for verifying anonymity and unlinkability properties to enforce strong action determinism in the AKA and Passive Authentication protocols.
<table>
<thead>
<tr>
<th>Property / Protocol</th>
<th>#roles</th>
<th>Single channel</th>
<th>1/0 unambiguous</th>
<th>SAD</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strong secrecy</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Denning-Sacco</td>
<td>6</td>
<td>3s</td>
<td>2m 7s</td>
<td>1m 58s</td>
</tr>
<tr>
<td>NSL</td>
<td>4</td>
<td>29s</td>
<td>1m</td>
<td>43s</td>
</tr>
<tr>
<td>Wide Mouth Frog</td>
<td>9</td>
<td>12m 16s</td>
<td>34m 43s</td>
<td>20m 1s</td>
</tr>
<tr>
<td>Yahalom-Lowe</td>
<td>6</td>
<td>2h 46m</td>
<td>11h 11m</td>
<td>5h 17m</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Anonymity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Passive Authentication</td>
<td>6</td>
<td>1h 59m</td>
<td>5h 6m</td>
<td>6h 49m</td>
</tr>
<tr>
<td>Private Authentication</td>
<td>4</td>
<td>9s</td>
<td>9s</td>
<td>11s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Unlinkability</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Passive Authentication</td>
<td>6</td>
<td>3h 15m</td>
<td>7h 15m</td>
<td>11h 6m</td>
</tr>
<tr>
<td>AKA</td>
<td>4</td>
<td>13m</td>
<td>26m 9s</td>
<td>17m 13s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Vote privacy</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helios</td>
<td>10</td>
<td>10m 9s</td>
<td>19m 10s</td>
<td>14m 50s</td>
</tr>
<tr>
<td>Scytl</td>
<td>3</td>
<td>2m 47s</td>
<td>5m 9s</td>
<td>5m 14s</td>
</tr>
</tbody>
</table>

6. Conclusion

In this paper we investigated two families of Dolev-Yao models, depending on how the hypothesis that the attacker controls the network is reflected. While the two semantics coincide for reachability properties, they yield incomparable notions of behavioral equivalences, which have recently been extensively used to model privacy properties. The fact that forcing all communication to be routed through the attacker may diminish his distinguishing power may at first seem counter-intuitive. We also propose a third semantics, where internal communication among honest participants is permitted but leaks the message to the attacker. This new communication semantics entails strictly stronger equivalences than the two classical ones. We also identify several subclasses of protocols for which (some) semantics coincide. Finally, we implemented the three semantics in the **DeepSec** tool. Our experiments showed that the three semantics provide the same result on the case studies in the **DeepSec** example repository. However, the private semantics is slightly more efficient, as less interleavings have to be considered. Our results illustrate that behavioral equivalences are much more subtle than reachability properties and the need to carefully choose the precise attacker model.

**Acknowledgments.** We would like to thank Catherine Meadows and Stéphanie Delaune for interesting discussions, as well as the anonymous reviewers for their comments. This work has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation program (grant agreement No 645865-SPOOC) and the ANR projects SEQUOIA ANR-14-CE28-0030-01 and TECAP ANR-17-CE39-0004-01.

**References**


\[A \ approx vs. (in^{ho}(c, x).out^{ho}(c, s) \mid in^{ho}(c, y).P(y))\]
\[B \ approx vs. (in^{ho}(c, x).(out^{ho}(c, s) \mid in^{ho}(c, y).P(y)))\]

where

\[P(y) \ approx \text{ if } y = s \text{ then } in^{ho}(c, z).out^{ho}(c, s)\]

Fig. 11. A and B (without else branches) such that \(A \ approx B\) and \(A \ not \ approx B\)


**Appendix A. Refining Theorem 3**

We here give a more refined version of Theorem 3. In particular we show that the private and classical semantics are incomparable for trace equivalence and labelled bisimulation, even when restricted to processes that do not use else branches.

**Theorem 12.** When restricted to processes without else branches, we have that \(\approx_p \ not \ subseteq \approx_c\) and \(\approx_p \ not \ subseteq \approx_c\) for \(r \in \{\ell, t\}\).

**Proof.** The fact that \(\approx_p \ not \ subseteq \approx_c\) for \(r \in \{\ell, t\}\) has already been shown in the proof of Theorem 3 as the processes \(A, B\) witnessing the result did not have else branches.

To show that \(\approx_p \ not \ subseteq \approx_c\) we show that there exist processes \(A, B\) without else branches such that \(A \ approx_p B\) and \(A \ not \ approx_c B\). Such processes are defined in Figure 11. To see that \(A \ approx_p B\) we first observe that the only first possible action from \(A\) or \(B\) is an input. In particular, given a term \(t\), there is a unique \(B'\) such that \(B \ underarrow{in^{(c,t)}} B'\) where \(B' = vs.(out^{ho}(c, s) \mid in^{ho}(c, y).P(y))\). On the other hand, if \(A \ underarrow{in^{(c,M)}} A'\) then either \(A' = B'\) or \(A' = A''\) where \(A'' \ approx vs.(in^{ho}(c, x).out^{ho}(c, s) \mid P(t))\). Therefore, to complete the proof, we only need to find \(B''\) such that \(B \ underarrow{in^{(c,t)}} B''\) and \(A'' \ approx_p B''\). Such process can be obtain by applying an internal communication on \(B'\), i.e. \(B \ underarrow{in^{(c,t)}} c\)

\(B' \ underarrow{vs.P(s)}\). Note that \(t \ not \ approx 0\) since \(s\) is bound, meaning that \(P(t) \ approx 0\). Moreover, \(P(s) \ approx \ approx P(s) \ approx 0\). This allows us to conclude that \(vs.P(s) \ approx A''\).
\[ A_i \doteq \nu_{s_1} \nu_{s_2}.(\text{out}^\text{ho}(c, h(s_1)) | \text{out}^\text{ho}(c, h(s_2)) | \\
\text{in}^\text{ho}(d, x). (\text{if } x = h(s_1) \text{ then } Q_1 \text{ if } x = h(s_2) \text{ then } P_2) \]

where \( Q_1 \doteq P_1 \), \( Q_2 \doteq P_2 \) and

\[ P_1 \doteq \text{out}^\text{ho}(e, a) \]
\[ P_2 \doteq \text{out}^\text{ho}(f, a). \text{out}^\text{ho}(e, a) | \text{in}^\text{ho}(f, x) \]

Fig. 12. \( A_1 \) and \( A_2 \) such that \( A_1 \approx^c_\ell A_2 \), but \( A_1 \not\approx^p_\ell A_2 \).

To see that \( A \not\approx^p_\ell B \) we first observe that when \( A \xrightarrow{\text{in}(c,t)} \rho A'' \), \( B \) can only mimic \( A \) by pre-
forming the transition \( B \xrightarrow{\text{in}(c,t)} B' \). We conclude as \( B' \xrightarrow{\text{out}(c,z)} \rho \nu_{a}.(\text{in}^\text{ho}(c, y). P(y) | \{^*/z\}) \) and
\[ A'' \xrightarrow{\text{out}(c,z)} \rho P \]

We next show that there also exist \( A_1 \) and \( A_2 \) such that \( A_1 \approx^c_\ell A_2 \), but \( A_1 \not\approx^p_\ell A_2 \).

We define such processes in Figure 12. Using the DeepSec tool we have shown that indeed
\( A_1 \approx^c_\ell A_2 \) and \( A_1 \not\approx^p_\ell A_2 \). The main argument why the result holds is that \( P_1 \) is trace included in
\( P_2 \) in the classical semantics (as the output on channel \( f \) can be made silent through an internal
communication) while this is not the case in the private semantics. \( \square \)

**Appendix B. Proof of Proposition 2**

**Definition 12.** We say that a plain process \( P \) (resp. extended process \( P \)) is name-cleaned if \( P \) is
of the form \( P_1 | \ldots | P_m \) and every \( P_i \) is not of the form \( \nu k. B' \) with \( k \) a name or variable of any
type.

**Lemma 9.** Let \( A \) be an extended process. There exist a sequence of names and variables \( \hat{k} \) and a
name-cleaned extended process \( A' \) such that \( A \equiv \nu k. A' \).

**Proof.** Direct from the definition of structural equivalence. \( \square \)

**Proposition 2.** For all closed honest plain processes \( A \), for all \( s \in \{c, e, p\} \), \( A \Downarrow^s_c \text{ iff there exists an attacker plain process } I^s \text{ such that } I^s | A \Downarrow^s_c \).

**Proof.** We will prove that \( A \Downarrow^s_c \) implies there exists an attacker plain process \( I^s \) such that \( C^s[A] \Downarrow^s_c \)
for \( s \in \{c, e, p\} \) by constructing \( I^s \).

Let us first focus on \( s = c \). Since \( A \Downarrow^s_c \), we know that there exist \( A' \), \( t \) and \( \text{tr} \in (A \setminus \{\tau\})^* \) such
that \( A \Downarrow^s_c A', c \notin \text{bn}(\text{tr}) \) and \( \text{out}(c, t) \in \text{tr} \). Note that we can assume w.l.o.g. that no name in \( \text{tr} \)
is bound twice and bound names in \( \text{tr} \) are distinct from free names that occurs in \( \text{tr} \) and \( \text{tr} \).

Let \( \{a_1, \ldots, a_k\} \) be all the channel names that occur in \( \text{tr} \) (bound or free). To each \( a_1, \ldots, a_k \),
we associate a variable of channel type \( x_{a_1}, \ldots, x_{a_k} \). Given a subset \( S \subseteq \{a_1, \ldots, a_k\} \), we denote
by \( \sigma(S) \) the substitution \( \{x_a \rightarrow a \mid a \in S\} \). We define \( I^s \) such that \( I^s = Q_c(\text{tr}, \sigma(\text{fc}(\text{tr}))) \) where
\( Q_c(\text{tr}, \sigma) \) is defined by induction on \( \text{tr} \) as follows:
that and two sequences of names $A$ for all definitions of $C$-Open property on the traces to account the presence of opened channels that are generated by the rule $\ell_1 \ldots \ell_N$. We can show by induction that for all $n \leq N$, there exist a plain process $Q_n$ and two sequences of names $\tilde{y}_n, \tilde{r}_n$ such that:

1. $I^c A \rightarrow^* \nu y_n.\nu \tilde{r}_n(A_n \mid Q_n)$
2. $\tilde{r}_n$ is the sequence of bounded channel names in $\ell_1 \ldots \ell_{n-1}$
3. $\tilde{y}_n \subseteq dom(\phi(A_n))$
4. $tr_n$ is the sequence $\ell_1 \ldots \ell_N$ where the $\tau$ action are removed
5. $Q_n = Q_c(tr_n, \sigma(f c(tr_n)))$

To conclude this proof, recall that $out(c, t) \in tr$ and $c \notin bn(tr)$ so there exists $n \leq N$ such that $\ell_n = out(c, t)$ or $\ell_n = \nu t.out(c, t)$. But since $A_{n-1} \xrightarrow{\ell_n} A_n$ and $A_{n-1} \equiv \nu k_{n-1}.B_{n-1}$ with $B_{n-1}$ being name-cleaned, we deduce that there exist $P, R$ such that $B_{n-1} = out^{bo}(c, t).P \mid R$ and $c \notin k_{n-1}$. Therefore, $I^c \mid A \rightarrow^* \nu y_{n-1}.\nu \tilde{r}_{n-1}.(\nu k_{n-1}.(out^{bo}(c, t).P \mid R \mid Q_{n-1})$. Note that $\tilde{y}_{n-1} \subseteq dom(\phi(B_{n-1}))$ hence $c \notin \tilde{y}_{n-1}$. Moreover, we assumed that $c \notin bn(tr)$ hence $c \notin \tilde{r}_{n-1}$ by definition of $\tilde{r}_{n-1}$. It allows us to conclude $I^c \mid A \not\parallel^c_{\psi}^{bo}$.

The proof for the other two semantics is very similar. First, the construction of the context changes to adapt the changes in the labeled semantics. Second, we prove a slightly different property on the traces to account the presence of opened channels that are generated by the rule $C$-OPEN. The rest stay the same (up to renaming of $c$ into $p$ and $e$ respectively).

Concerning the semantics private, we define $I^p \equiv I^c$ and we can prove the following sequences for all $n \leq N$, there exist two extended processes $Q_n, R_n$ and two sequences of names $\tilde{y}_n, \tilde{r}_n$ such that:

1. $C_p[A] \rightarrow^* \nu y_n.\nu \tilde{r}_n.(A_n \mid Q_n \mid R_n)$
2. $\tilde{r}_n$ is the sequence of bounded channel names in $\ell_1 \ldots \ell_{n-1}$
3. $R_n \equiv \omega c_1 \mid \ldots \mid \omega c_m$ for some $c_1, \ldots, c_m$ such that $\tilde{r}_n \subseteq \{c_1, \ldots, c_m\}$
4. $\tilde{y}_n \subseteq dom(\phi(B_n))$
5. $tr_n$ is the sequence $\ell_1 \ldots \ell_N$ where the $\tau$ action are removed
Notice that the presence of \( R_n \) is the only difference between the property in the classical and private semantics. This is the consequence of the application of the rule C-OPEN that introduces opened channels \( \omega c_1 \) and that we apply when the trace contains labeled transitions \( \text{out}(c,d) \) or \( \nu d . \text{out}(c,d) \).

For the eavesdropping semantics, we can prove the same property as for private semantics (up to renaming of \( p \) into \( e \)) but we need to modify the context as follows. We define \( C_e[\_] \) such that

\[
I^s[\_] \triangleright Q_e(\nu x . eav, \sigma) = Q_e(\nu x . eav, \sigma) \quad \text{if } y \text{ is of base type then}
\]

\[
Q_e(\nu x . eav, \sigma) = \text{eav}(x, \sigma, y).Q_e(\text{tr}', \sigma)
\]

where \( y \) is fresh variable of base type.

Let us now focus on the other implications, that are: if there exists an attacker plain process \( I^s \) such that \( I^s \models A \downarrow_c^{s . \text{ho}} \) then \( A \|_c^s \) for \( s \in \{c, e, p\} \). By Lemma 9, we can assume w.l.o.g. that \( I^s = \nu k . D \) for some name-cleaned plain process \( D \) and some sequence of names and variables of any type \( k \). We now prove that for all \( I^s \models A \rightarrow^*_s B \), there exist an attacker evaluation context \( C'[\_] \vdash k'.(D' \ | \_) \) with \( D' \) name-cleaned, an honest extended process \( A' \) and \( \text{tr} \) such that \( C'[\_] \) is s-closing for \( A, B \equiv C'[A'] \) and \( A \upmodels^s A' \). We first focus on \( s = c \). Note that it is not necessary to prove the property name-cleaned since it is implied by Lemma 9.

We prove this result by induction on the number of reduction rules in \( I^c \models A \rightarrow^*_c B \).
Base case: By structural equivalence, there exists $\bar{k}'$ and $D'$ such that $I | A \equiv \nu k'.(D' \mid A)$. Moreover, since $fv(A) = \emptyset$, $k'.(D' \mid \_)$ is closing for $A$ and so the base case holds.

Inductive step $C^\nu[A] \Rightarrow \_ : B': \_ \Rightarrow B$: By our inductive hypothesis, we know that there exist $C^\nu[\_] = \nu k'.(D' \mid \_)$, and honest extended process $A'$ and $tr$ such that $C^\nu[\_]$ is $c$-closing for $A'$, $B' \equiv C^\nu[A']$ and $A \triangleright\_ \equiv A'$. Note that due to the structural equivalence, we can assume w.l.o.g. that $A' = \nu r'.P$ where $P$ is name-cleaned. Moreover, since $B' \Rightarrow \_ B$ and $B' \equiv C^\nu[A']$, we deduce that $C^\nu[A'] \Rightarrow \_ B$. Let us do a case analysis on the rule applied.

Case 1, internal reduction on $A'$, i.e. there exists $A''$ such that $A' \triangleright_{\nu c} A''$ and $B \equiv C^\nu[A'']$. In such a case, we have that $C^\nu[A'] \triangleright_{\nu c} C^\nu[A'']$. Moreover, since $A \triangleright_{\nu c} A''$ and so the result holds.

Case 2, internal reduction on $C'$, i.e. there exists $D''$ such that $D' \triangleright_{\nu c} D''$ and $B \equiv \nu k'.(D'' \mid A')$. By the structural equivalence, we know that there exist $\bar{k}'$ and $D''$ such that $D''$ is name-cleaned and $\nu k'.(D'' \mid A') \equiv \nu k''.(D'' \mid A')$. Therefore, we can define $C''[\_] = \nu k''.(D'' \mid \_)$ and obtain that $C''[\_] \Rightarrow \_ C''[\_].$ Since $A \triangleright_{\nu c} A''$, the result holds.

Case 3, rule Comm between $C'$ (input) and $A'$ (output), i.e. $D' = in^x(c,x).D_1 \mid D_2$, $A' = \nu r.(out^h(c,u).P_1 \mid P_2)$ and $B \equiv \nu k'.\nu r.(D_1\{u/x\} \mid D_2 \mid P_1 \mid P_2)$ (We assume w.l.o.g. that the names and variables in $\bar{r}$ are not in $D'$). Note that in such a case, $c \notin \bar{r}$. We do a case analysis on $u$.

- Case 3.a, $u \in Ch \cap \bar{r}$: Let us re-denote $\nu \bar{r}$ as $\nu \bar{r}.uu$. Thus $A' \triangleright_{\nu r.out^h(c,u)} \nu \bar{r}.uu.(P_1 \mid P_2)$. Hence, since the names and variables in $\bar{r}$ are not in $D'$, we obtain that $B \equiv \nu k'.uu.(D_1\{u/x\} \mid D_2 \mid \nu \bar{r}.uu.(P_1 \mid P_2))$, the result holds.
- Case 3.b, $u \in Ch$ but $u \notin \bar{r}$: In such a case, $A' \triangleright_{\nu r.out^h(c,u)} \nu \bar{r}.(P_1 \mid P_2)$. Hence, since the names and variables in $\bar{r}$ are not in $D'$, we obtain that $B \equiv \nu k'.(D_1\{u/x\} \mid D_2 \mid \nu \bar{r}.(P_1 \mid P_2))$. By denoting $C''[\_] = \nu k'.(D_1\{u/x\} \mid D_2 \mid \_)$, the result holds.
- Case 3.c, $u \notin Ch$: In such a case, $A' \triangleright_{\nu r.out^h(c,u)} \nu \bar{r}.(P_1 \mid P_2 \mid \{u/y\})$ with $y \notin fv(A') \cup uu$. Note we can take $y$ such that $y \notin fv(C'[A']) \cup bn(C'[A'])$. Note that $B \equiv \nu k'.\nu r.(D_1\{u/x\} \mid D_2 \mid P_1 \mid P_2)$, by definition of the structural equivalence and since we took $y \notin fv(C'[A']) \cup bn(C'[A'])$, we deduce that $B \equiv \nu k'.\nu r.(D_1\{u/x\} \mid D_2 \mid P_1 \mid P_2 \mid \{u/y\})$. Lastly, since the names and variables in $\bar{r}$ are not in $D'$, we deduce that $B \equiv \nu k'.\nu r.(D_1\{u/x\} \mid D_2 \mid \nu \bar{r}.(P_1 \mid P_2 \mid \{u/y\})$. By denoting $C''[\_] = \nu k'.\nu r.(D_1\{u/x\} \mid D_2 \mid \_)$, the result holds.

Case 4, rule Comm between $A'$ (input) and $C'$ (output), i.e. $D' = out^x(c,x).D_1 \mid D_2$, $A' = \nu r.(in^h(c,u).P_1 \mid P_2)$ and $B \equiv \nu k'.\nu r.(D_1 \mid D_2 \mid P_1\{u/x\} \mid P_2)$ (We assume w.l.o.g. that the names and variables in $\bar{r}$ are not in $D'$). Note that in such a case, $c \notin \bar{r}$. Moreover, we know that the names and variables in $\bar{r}$ are not in $D'$, meaning that the names and variables in $\bar{r}$ does not occur in $u$. Hence, $A' \triangleright_{\nu r.in^h(c,u)} \nu \bar{r}.(P_1\{u/x\} \mid P_2)$. Once again due to the fact the names and variables in $\bar{r}$ are not in $D'$, we obtain that $B'' \equiv \nu k'.(D_1 \mid D_2 \mid \nu \bar{r}.(P_1\{u/x\} \mid P_2))$. By denoting $C''[\_] = \nu k'.(D_1 \mid D_2 \mid \_)$, the result holds.
We have concluded the proof of the property: for all $C^c[A] \rightarrow^*_c B$, there exist an evaluation attacker context $C'[\_] = \nu k'.(D' \mid \_)$ with $D'$ name-cleaned, an honest extended process $A'$ and $tr$ such that $C'[\_]$ is c-closing for $A'$, $B \equiv C^c[A]$ and $A \not\equiv^c_c A'$. It remains to prove this result for $s = e$ and $s = p$. Let us focus first on the case $s = p$. The proof is in fact similar to the case $s = c$. Notice that the case of the rule C-Env correspond to either Case 2, 3.c or 4 when $u$ is of base type. Hence it remains the case of the rules C-PRIV and C-OPEN.

Case 5, rule C-OPEN between $C'$ (input) and $A'$ (output), i.e. $D' = \text{in}^0(c, x).D_1 \mid D_2$, $A' = \nu r.(\text{out}^{ho}(c, d).P_1 \mid P_2)$ and $B \equiv \nu k'.\nu r.(D_1 \{d/x\} \mid D_2 \mid P_1 \mid P_2 \mid \omega d)$. Let us do a case analysis on whether $(5.a) \! d \in \! \bar{r}$ or $(5.b) \! d \not\in \! \bar{r}$. Note that Case $(5.a)$ is in fact almost identical to Case $(3.a)$ and that the result holds with $C''[\_] = \nu k'.\nu d.(D_1 \{d/x\} \mid D_2 \mid \omega d \mid \_)$ and $A'' = \nu r'.(P_1 \mid P_2)$ with $\nu r = \nu r'.\nu d$. Furthermore, note that Case $(5.b)$ is also very similar to Case $(3.b)$ and that the result holds with $C''[\_] = \nu k'.(D_1 \{d/x\} \mid D_2 \mid \omega d \_)$ and $A'' = \nu r.(P_1 \mid P_2)$. Notice that in both case $C''[\_]$ is indeed p-closing for $A''$.

Case 6, rule C-OPEN between $A'$ (input) and $C'$ (output), i.e. $D' = \text{out}^0(c, d).D_1 \mid D_2$, $A' = \nu r.(\text{in}^{ho}(c, x).P_1 \mid P_2)$ and $B \equiv \nu k'.\nu r.(D_1 \mid D_2 \mid P_1 \{d/x\} \mid P_2 \mid \omega d)$. This case if very similar to Case 4 when $u$ is of channel type and the result holds with $C''[\_] = \nu k'.(D_1 \mid D_2 \mid \omega d \_)$ and $A'' = \nu r.(P_1 \{d/x\} \mid P_2)$.

Case 7, rule C-PRIV with a communication on a channel $c$. Notice that this rule is in fact partially covered by the beginning of the proof. Indeed, Case 1 and 2 cover the cases where $c$ is not in $k'$. Therefore, we only need to focus on the case where the private channel is in $k'$, i.e. $\nu k' = \nu k''.\nu c$ for some $\tilde{k}$. We know that $C''[\_]$ is p-closing for $A'$. Hence since $c$ is a channel bound in $C''[\_]$ whose scope includes $\_$, we deduce that if $c \in fr(A)$ then $\omega c$ is also in the scope of $c$. But according to the definition of the rule, we know that $\omega c$ is not in the scope of $\nu c$. Moreover, if the output or input is done by $A'$ then it would implies that $c \in fr(A)$. Thus, this allows us to deduce that this both output and input are tagged with $at$, meaning that there exists $D''$ such that $\nu c.(D' \mid A') \not\rightarrow^*_p \nu c.(D'' \mid A')$ and $B \equiv \nu k''.\nu c.(D'' \mid A')$. In such a case, by denoting $C''[\_] = \nu k''.\nu c.(D'' \mid \_)$ and $A'' = A'$, the result directly holds.

We have concluded the proof of the property for $s = p$ hence it remains the case $s = e$. Once, again several cases are already covered since $\not\rightarrow^*_p C \not\rightarrow^*_e$. Hence we only need to focus on the cases of the rules C-EAV and C-OEAV:

Case 8, rule C-EAV, i.e. $A' = \nu r.(\text{out}^{ho}(c, u).P_1 \mid \text{in}^{ho}(c, x).P_2 \mid P_3)$, $D'' = \text{eav}(c, y).Q_1 \mid Q_2$, $B \equiv \nu k'.\nu r.(Q_1 \{u/y\} \mid Q_2 \mid P_1 \mid P_2 \{u/x\} \mid P_3)$ and $u$ is of base type (We assume w.l.o.g. that the names and variables in $\bar{r}$ are not in $D'$). Note that in such a case $c \not\in \bar{r}$. Moreover, note this is the only possible combination of input and output since $C'$ is an attacker evaluation context and $A'$ is an honest extended process. Let us consider a variable $z$ such that $z \not\in \text{fv}(C'[A']) \cup \text{bn}(C'[A'])$.

Hence $A' \not\rightarrow^*_{\nu z,\text{eav}(cz)} \nu r.(P_1 \mid P_2 \{u/x\} \mid P_3 \mid \{u/z\})$. But since $z \not\in \text{fv}(C'[A']) \cup \text{bn}(C'[A'])$, we deduce that $B \equiv \nu k'.\nu z.\nu r.(Q_1 \{z/y\} \mid Q_2 \mid P_1 \mid P_2 \{u/x\} \mid P_3 \mid \{u/z\})$. Hence, by denoting $C''[\_] = \nu k'.\nu z.(Q_1 \{z/y\} \mid Q_2 \mid \_)$ and $A'' = \nu r.(P_1 \mid P_2 \{u/x\} \mid P_3 \mid \{u/z\})$, the result holds.
Case 9. rule C-OEAV, i.e. \( A' = \nu r_\ast ((\text{out} \circ \text{ho})(c, d) \cdot P_1 \mid \text{in} \circ \text{ho}(c, x) \cdot P_2 \mid P_3) \), \( B' = \nu k_\ast \cdot \nu r_\ast (Q_1\{d/y\} \mid Q_2 \mid P_1 \mid P_2\{d/x\} \mid P_3 \mid \omega d) \) and \( d \) is of channel type (We assume w.l.o.g. that the names and variables in \( \tilde{r} \) are not in \( D' \)). We have to do a case analysis on \( d \):

- Case \( d \in \tilde{r} \): Let us denote \( \nu r_\ast = \nu r' \cdot \nu d_\ast \). In such a case \( A' \xrightarrow{\text{vd.eav}(c,d)} \nu r' \cdot (P_1 \mid P_2\{u/x\} \mid P_3) \).
  
  But we know that the names and variables in \( \tilde{r} \) are not in \( D' \) hence \( B'' = \nu k' \cdot \nu d_\ast (Q_1\{d/y\} \mid Q_2 \mid \nu r' \cdot (P_1 \mid P_2\{d/x\} \mid P_3)) \). Therefore, by denoting \( C''[\_] = \nu k' \cdot \nu d_\ast (Q_1\{u/y\} \mid Q_2 \mid \omega d \mid \_) \) and \( A'' = \nu r_\ast (P_1 \mid P_2\{d/x\} \mid P_3) \), the result holds.

- Case \( d \not\in \tilde{r} \): In such a case \( A' \xrightarrow{\text{eav}(c,u)} \nu r_\ast (P_1 \mid P_2\{u/x\} \mid P_3) \) and so the result holds by denoting \( C''[\_] = \nu k' \cdot (Q_1\{u/y\} \mid Q_2 \mid \omega d \mid \_) \) and \( A'' = \nu r_\ast (P_1 \mid P_2\{u/x\} \mid P_3) \).

Note that in both case, \( C''[\_] \) is indeed \( e \)-closing for \( A'' \).

We have proved that for all \( s \in \{c, p, e\} \), for all \( C^* [A] \rightarrow^s B \), there exist an attacker evaluation context \( C''[\_] = \nu k' \cdot (D' \mid \_) \) with \( D' \) name-cleaned, an honest extended process \( A' \) and \( \text{tr} \) such that \( C''[\_] \) is \( s \)-closing for \( A' \), \( B \equiv C'[A'] \) and \( A \triangleq^s A' \). This property allows us to conclude the main proof. Indeed, consider \( s \in \{c, e, p\} \) and \( C''[\_] \) an attacker evaluation context such that \( C^* [A] \not\equiv^s \text{ho} \). By definition, we deduce that \( C^* [A] \rightarrow^s C' \) for some evaluation context \( C \) that does not bind \( c \), some \( t \) and some plain process \( P \). By our property, we deduce that there exists an attacker evaluation context \( C' \) with \( D' \) name-cleaned, an honest extended process \( A' \) and \( \text{tr} \) such that \( C' \equiv C'[A'] \) and \( A \triangleright^s A' \). More specifically, since \( C''[\_] \) is an attacker evaluation context, \( C' \triangleright^s A' \) and \( C \) does not bind \( c \), we deduce that \( A' \equiv \nu r_\ast (\text{out} \circ \text{ho})(c, t) \cdot P' \mid Q' \) for some \( t', P', Q', \tilde{r} \) such that \( c \not\in \tilde{r} \). Therefore, if \( t' \in Ch \) but \( t' \not\in \tilde{r} \) then \( A' \xrightarrow{\text{out}(c,t')} A'' \) for some \( A'' \) meaning that \( A \xrightarrow{\text{tr}(c,t')} A'' \); else \( A' \xrightarrow{\text{tr}(c,z)} A'' \) for some \( A'' \) and some \( z \) fresh (\( z \) being either a base type variable or a channel), meaning that \( A \xrightarrow{\text{tr}(c,z)} A'' \). In both cases, we obtain that \( A \triangleright^s A'' \), \( \text{out}(c, t) \in \text{tr}' \) and \( c \not\in \text{tr}(c, t) \) for some \( \text{tr}' \), \( A'' \) and \( t \). It allows us to conclude that \( A \triangleright^s \). □

Appendix C. Proof of Theorem 1

We start by restating the a proposition from [19] that was used to prove that trace equivalence implies may equivalence in the classical semantics. In order to prove the proposition for the semantics private and eavesdrop, we will first write exactly the proof of from [19] for the classical semantics and then highlight what changes are required to obtain the proofs for the private and eavesdropping semantics.

Proposition 4. Let \( s \in \{c, p, e\} \). Let \( A \) and \( B \) be two honest closed extended process with \( \text{dom}(A) = \text{dom}(B) \), and \( C[\_] = \nu r_\ast (D \mid \_) \) be an attacker evaluation context \( s \)-closing for \( A \). If \( C[A] \rightarrow^s A'' \) for some process \( A'' \), then there exists a closed extended process \( A' \), an attacker evaluation context \( C' = \nu r_\ast (D' \mid \_) \) \( s \)-closing for \( A' \), and a trace \( \text{tr} \in (A \setminus \{\tau\})^* \) such that \( A'' \equiv C'[A'] \), \( A \triangleright^s A' \), and for all closed extended process \( B' \), we have:

\[
C'[\_] \text{ is } s \text{-closing for } B \text{ and } B \triangleright^s B' \text{ and } \phi(B') \sim \phi(A')
\]

implies that

\[
C' \text{ is } s \text{-closing for } B' \text{ and } C[B] \rightarrow^s C'[B']
\]
Proof. We first focus on the case $s = \cdot$. Let $A$ and $B$ be two extended processes with $dom(A) = dom(B)$ and $C[..] = \nu n.(D | ..)$ be an evaluation context $\cdot$-closing for $A$. Let $A''$ be such that $C[A] \rightarrow_{\cdot}^* A''$. We prove the result by induction on the length $\ell$ of the derivation.

Base case $\ell = 0$: In such a case, we have that $A'' \equiv C[A]$. Let $A' = A$, $C' = C$ and $tr = \cdot$, we have that $A'' \equiv C'[A']$, and $A \Rightarrow_{\cdot} A'$. Let $B'$ be a closed extended process such that $B \Rightarrow_{\cdot} A'$ and $\phi(B') \sim \phi(A')$ for some $B'$. Clearly, we have that $C[B] \rightarrow_{\cdot}^* C'[B']$ and $C'[..]$ is $\cdot$-closing for $B'$ since $C' = C$ and $B \rightarrow_{\cdot} B'$.

Inductive case $\ell > 0$: In such a case, we have that there exists a closed extended process $A_1$ such that $C[A] \rightarrow_{\cdot}^* A_1$ with a derivation whose length is smaller than $\ell$, and $A_1 \rightarrow_{\cdot} A''$. Thus, we can apply our induction hypothesis allowing us to deduce that there exist an extended process $A'_1$, an evaluation context $C'_1[..] = \nu n'.(D'_1 | ..)$ $\cdot$-closing for $A'_1$, and a trace $tr_1 \in (A \setminus \{\cdot\})^*$ such that $A_1 \equiv C'_1[A'_1]$, $A \Rightarrow_{\cdot} A'_1$, and for all closed extended processes $B'_1$, we have that:

\begin{equation}
C[..] \text{ is } \cdot\text{-closing for } B \text{ and } B \Rightarrow_{\cdot} B'_1 \text{ and } \phi(B'_1) \sim \phi(A'_1)
\end{equation}

implies that

\begin{equation}
C'_1[..] \text{ is } \cdot\text{-closing for } B'_1 \text{ and } C[B] \rightarrow_{\cdot}^* C'_1[B'_1].
\end{equation}

Since $A_1 \equiv C'_1[A'_1]$ and $A_1 \rightarrow_{\cdot} A''$, we have that $C'_1[A'_1] \rightarrow_{\cdot} A''$, (internal reduction is closed under structural equivalence). W.l.o.g., we can assume that $D'_1$ is name-cleaned, the bound names and variables in $C'_1[A'_1]$ are bound once and distinct from the free names. We do a case analysis on the rule involved in this reduction.

Case 1: internal reduction in $A'_1$, i.e. there exists $A'$ such that $A'_1 \rightarrow_{\cdot} A'$ and $A'' \equiv C'_1[A']$. In such a case, we have that $C'_1[A'_1] \rightarrow_{\cdot} C'_1[A']$. Let $C'[..] = C'_1[..]$ and $tr = tr_1$. We have that $A'' \equiv C'[A'] = C'[A']$ and $A \Rightarrow_{\cdot} A'_1 \rightarrow_{\cdot} A'$, i.e. $A \Rightarrow_{\cdot} A'$. Lastly, let $B'$ be a closed extended process such that $B \Rightarrow_{\cdot} A'$ and $\phi(B') \sim \phi(A')$. We have that $B \Rightarrow_{\cdot} B'$ and $\phi(B') \sim \phi(A'_1) \equiv \phi(A')$, and thus relying on our induction hypothesis, we obtain that $C[..] \text{ is } \cdot\text{-closing for } B'$ and $C[B] \rightarrow_{\cdot}^* C'_1[B']$. Since $C'_1[..] = C[..]$, we conclude.

Case 2.a: rule Then in $D'_1$, i.e. $D'_1 = \text{ if } u = v \text{ then } P_1 \text{ else } P_2 \mid P_3$ and $A'' \equiv \nu n'.(P_1 | P_3 | A'_1)$. In such a case, we have $C'_1[A'_1] \rightarrow_{\cdot} \nu n'.(P_1 | P_3 | A'_1)$. Let $A' = A'_1$, $C'[..] = \nu n'.(P_1 | P_3 | ..)$ and $tr = tr_1$. We have that $A'' \equiv C'[A']$ and $A \Rightarrow_{\cdot} A'$. Lastly, let $B'$ be a closed extended process such that $B \Rightarrow_{\cdot} B'$ and $\phi(B') \sim \phi(A')$. By renaming, we can assume that the bound names of $B'$ are distinct from the free names of $C'_1$. Moreover, we know that $C'_1$ is $\cdot$-closing for $A'_1$ meaning that $v(u, v) \subseteq dom(\phi(A'_1))$. Furthermore, since the free names are distinct from bound names, we obtain that $fn(u, v) \cap bn(A'_1) = \emptyset$. But $\phi(A'_1) = \phi(A') \sim \phi(B')$ and $(u =_{\cdot} v) \phi(A')$ hence we obtain $(u =_{E} v)\phi(B')$ meaning that $C'_1[B'] \rightarrow_{\cdot} \nu n'.(P_1 | P_3 | B') = C'[B']$. By our inductive hypothesis, we also have that $C'_1$ is $\cdot$-closing for $B'$ and $C[B] \rightarrow_{\cdot}^* C'[B']$. Hence, we conclude that $C[B] \rightarrow_{\cdot}^* C'[B']$ and $C'[..]$ is $\cdot$-closing for $B'$.

Case 2.b: rule Else in $D'_1$, i.e. $D'_1 = \text{ if } u = v \text{ then } P_1 \text{ else } P_2 \mid P_3$ and $A'' \equiv \nu n'.(P_2 | P_3 | A'_1)$. Similar to case 2.a.
Case 3: rule Comm in $D'_1$, i.e. $D'_1 = \text{out}^!(c,u).P_1 \parallel \text{in}^!(c,x).P_2 \parallel P_3$ and $A'' \equiv \nu \nu^! \cdot (P_1 \parallel P_2 \parallel x \parallel P_3 \parallel A'_1)$. In such a case, we have $C'[A'_1] \rightarrow_c \nu \nu^! \cdot (P_1 \parallel P_2 \parallel x \parallel P_3 \parallel A'_1)$. Let $A' = A'_1$, $C'[\_] = \nu \nu^! \cdot (P_1 \parallel P_2 \parallel x \parallel P_3 \parallel \_)$ and $tr = tr_1$. We have that $A'' \equiv C'[A']$ and $A \frac{\_}{\_} A'$. Lastly, let $B'$ be a closed extended process such that $B \frac{\_}{\_} \_ B'$ and $\phi(B') \sim \phi(A')$. Since $\phi(A') = \phi(A'_1)$ then by our inductive hypothesis, we obtain $C'_1$ is $c$-closing for $B'$ and $C[B] \rightarrow_c C'[B']$. But $C'_1[B'] \rightarrow_c \nu \nu^! \cdot (P_1 \parallel P_2 \parallel x \parallel P_3 \parallel B' = C'[B']$ and so the result holds.

Case 4: rule Comm between $D'_1$ (output) and $A'_1$ (input), i.e. $D'_1 = \text{out}^!(c,M).P_1 \parallel P_2$, $A'_1 = \nu \nu^! \cdot (\text{in}^!(c,x).Q_1 \parallel Q_2)$ and $A'' \equiv \nu \nu^! \cdot \nu \nu^! \cdot (P_1 \parallel P_2 \parallel Q_1 \parallel Q_2)$ (recall that we assume that bound names and variables are distinct from free names and variables and are only bound once). Note that in such a case, $c \notin \tilde{r}$. Hence $A'_1 \frac{\text{in}(c,M)}{\nu \nu^! \cdot (Q_1 \parallel Q_2)}$. Moreover, since $\tilde{r}$ is not in $P_1, P_2$, we have $A'' \equiv \nu \nu^! \cdot (P_1 \parallel P_2 \parallel \nu \nu^! \cdot (Q_1 \parallel Q_2)$. Let $A' = \nu \nu^! \cdot (Q_1 \parallel Q_2)$, $C'[\_] = \nu \nu^! \cdot (P_1 \parallel P_2 \parallel \_)$ and $tr = tr_1.\text{in}(c,M)$. We have that $A'' \equiv C'[A']$ and $A \frac{\_}{\_} A'$. Lastly let $B'$ be a closed extended process such that $B \frac{\_}{\_} \_ B'$ and $\phi(B') \sim \phi(A')$. We have that there exists $B'_1$ such that $B \frac{\_}{\_} \_ B'_1 \frac{\text{in}(c,M)}{\nu \nu^! \cdot (Q_1 \parallel Q_2)} \_ B'$. By renaming, we can assume that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once. Since $\phi(B') \sim \phi(A')$, we have also that $\phi(B'_1) \sim \phi(A'_1)$. Thus, we can apply our induction hypothesis on $B'_1$. This allows us to deduce that $C[B] \rightarrow_c C'[B'_1]$ and $C'_1$ is $c$-closing for $B'_1$. In order to conclude, it remains to show that $C'_1[B']$ is $c$-closing for $B'_2$ and $B'_2 \rightarrow_c \_ B'$ implies $C'[B'_2] \rightarrow_c C'[B']$ and $C'$ is $c$-closing for $B'_2$.

We have seen that $B'_1 \frac{\text{in}(c,M)}{\nu \nu^! \cdot (Q_1 \parallel Q_2)}$. Hence, we know that $B'_1 = \nu \nu^! \cdot (\text{in}^!(c,x).Q_1 \parallel Q_2)$ for some $\tilde{r}'$, $Q'_1, Q'_2$ and $B'_2 \equiv \nu \nu^! \cdot (Q'_1 \parallel Q'_2)$. But since we assumed that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once, we obtain that $C'_1[B'_1] \equiv \nu \nu^! \cdot \nu \nu^! \cdot \text{out}^!(c,M).P_1 \parallel P_2 \parallel \text{in}^!(c,x).Q'_1 \parallel Q'_2)$. Hence $C'_1[B'_1] \rightarrow_c \nu \nu^! \cdot \nu \nu^! \cdot (P_1 \parallel P_2 \parallel Q'_1 \parallel Q'_2) \equiv \nu \nu^! \cdot \nu \nu^! \cdot (Q'_1 \parallel Q'_2) \equiv C'[B'_2]$. Notice that $C'[\_]$ is $c$-closing for $B'_2$ since $\nu \nu^! \cdot (C'_1[B'_1]) = \emptyset$.

Case 5: rule Comm between $C'_1$ (input) and $A'_1$ (output), i.e. $D'_1 = \text{in}^!(c,x).P_1 \parallel P_2$, $A'_1 = \nu \nu^! \cdot (\text{out}^!(c,M).Q_1 \parallel Q_2)$ and $A'' \equiv \nu \nu^! \cdot \nu \nu^! \cdot (P_1 \parallel P_2 \parallel Q_1 \parallel Q_2)$ (recall that we assume that bound names and variables are distinct from free names and variables and are only bound once). Note that in such a case, $c \notin \tilde{r}$. We do a case analysis on $M$.

- Case 5.a, $M \in \text{Ch} \cap \tilde{r}$: Let us denote $\nu \nu^! = \nu \nu^! \cdot M$. Thus $A'_1 \frac{\nu \nu^! \cdot M \cdot \text{out}(c,M)}{\nu \nu^! \cdot (Q_1 \parallel Q_2)}$. Hence, since the names and variables in $\tilde{r}$ are not in $D'_1$, we obtain that $A'' \equiv \nu \nu^! \cdot \nu \nu^! \cdot M \cdot (P_1 \parallel P_2 \parallel \nu \nu^! \cdot (Q_1 \parallel Q_2))$. Let $A' = \nu \nu^! \cdot (Q_1 \parallel Q_2)$, $C'[\_] = \nu \nu^! \cdot \nu \nu^! \cdot M \cdot (P_1 \parallel P_2 \parallel \_)$ and $tr = tr_1.\nu \nu^! \cdot \text{out}(c,M)$. We have that $A'' \equiv C'[A']$ and $A \frac{\_}{\_} A'$. Lastly, let $B'$ be a closed extended process such that $B \frac{\_}{\_} \_ B'$ and $\phi(B') \sim \phi(A')$. We have that there exists $B'_1$ such that $B \frac{\_}{\_} \_ B'_1 \frac{\nu \nu^! \cdot M \cdot \text{out}(c,M)}{\nu \nu^! \cdot (Q_1 \parallel Q_2)} \_ B'$. By renaming, we can assume that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once. Since $\phi(B') \sim \phi(A')$, we have also that $\phi(B'_1) \sim \phi(A'_1)$. Thus, we can apply our induction hypothesis on $B'_1$. This allows us to deduce that $C[B] \rightarrow_c C'[B'_1]$ and $C'[\_]$ is $c$-closing for $B'_1$. In order to conclude, it remains to show that $C'_1[B'_1] \rightarrow_c C'[B'_2]$ (since $\nu \nu^! \cdot (C'_1[B'_1])$ and since $B'_2 \rightarrow_c \_ B'$ implies $C'[B'_2] \rightarrow_c C'[B']$).
We have seen that $B'_1 \xrightarrow{\nu_{M,\text{out}(c,M)}} B'_2$. Hence, $B'_1 = \nu m.\nu M.(\text{out}^0(c,M).Q'_1 | Q'_2)$ for some $m$, $Q'_1$, $Q'_2$ and $B'_2 \equiv \nu m.\nu M.(P_1 | P_2 | \text{out}^0(c,M).Q'_1 | Q'_2)$. But since we assumed that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once, we obtain that $C'[B'_1] \equiv \nu n'_1.\nu m.\nu M.(\text{in}^n(c,x).P_1 | P_2 | \text{out}^0(c,M).Q'_1 | Q'_2)$. Hence $C'[B'_1] \rightarrow_c \nu n'_1.\nu m.\nu M.(P_1^{\{M/x\}} | P_2 | Q'_1 | Q'_2) \equiv C'[\nu m.(Q'_1 | Q'_2)] \equiv C'[B'_2].$

- Case 5.b, $M \notin \check{r}$: Thus $A'_1 \xrightarrow{\text{out}(c,M)} \nu r_.(Q_1 | Q_2)$. Hence, since the names and variables in $\check{r}$ are not in $D'_1$, we obtain that $A'' \equiv \nu n'_1.(P_1^{\{M/x\}} | P_2 | \nu r_.(Q_1 | Q_2))$. Let $A' = \nu r_.(Q_1 | Q_2), C'[\_] = \nu n'_1.(P_1^{\{M/x\}} | P_2 | \_)$ and $tr = tr_1.\text{out}(c,M)$. We have that $A'' \equiv C'[A']$ and $A \xrightarrow{\text{tr}_c} A'$. Lastly let $B'$ be a closed extended process such that $B \xrightarrow{\text{tr}_c} B'$ and $\phi(B') \sim \phi(A')$. We have that there exists $B'_1$ such that $B \xrightarrow{\text{tr}_c} B'_1 \xrightarrow{\text{out}(c,M)} B'_2 \rightarrow^* B'$. By renaming, we can assume that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once. Since $\phi(B') \sim \phi(A')$, we have also that $\phi(B'_1) \sim \phi(A'_1)$. Thus, we can apply our induction hypothesis on $B'_1$. This allows us to deduce that $C[B] \rightarrow^*_c C'[B'_1]$ and $C'[\_]$ is $c$-closing for $B'_1$. In order to conclude, it remains to show that $C'[B'_1] \rightarrow_c C'[B']$ (since $fv(C'_1[B'_1])$ and since $B'_2 \rightarrow^*_c B'$ implies $C'[B'_2] \rightarrow^*_c C'[B']$).

We have seen that $B'_1 \xrightarrow{\text{out}(c,M)} B'_2$. Hence, $B'_1 = \nu m.\nu M.(Q'_1 | Q'_2)$ for some $m$, $Q'_1$, $Q'_2$ such that $M \notin \check{m}$ and $B'_2 \equiv \nu m.\nu M.(P_1 | P_2 | \text{out}^0(c,M).Q'_1 | Q'_2)$. But since we assumed that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once, we obtain that $C'[B'_1] \equiv \nu n'_1.\nu m.\nu M.(\text{in}^n(c,x).P_1 | P_2 | \text{out}^0(c,M).Q'_1 | Q'_2)$. Moreover, since $y$ is fresh, we obtain that $\nu n'_1.\nu y.\nu M.(\text{in}^n(c,x).P_1 | P_2 | \text{out}^0(c,y).Q'_1 | Q'_2 | \{y/g\})$. Hence $C'[B'_1] \rightarrow_c \nu n'_1.\nu y.\nu M.(P_1^{\{y/x\}} | P_2 | Q'_1 | Q'_2 | \{y/g\}) \equiv C'[\nu m.(Q'_1 | Q'_2)] \equiv C'[B'_2].$

This conclude the proof of the proposition for $s = c$. Therefore, it remains to take care of the cases $s = p$ and $s = e$. Let us focus first on the case $s = p$. The proof is in fact very similar to the classical semantics. Considering that the differences between the classical semantics and the
private semantics are on the internal communication, we only need the rules that are not already covered in the classical proof. Notice that the rule C-ENV correspond to either Case 3 or Case 4 when M is of base type or Case 5.c. Moreover, the rules THEN and ELSE are already covered either Case 1 or 2.a or 2.b. Hence it remains the case of the rules C-PRIV and C-OPEN.

Case 6, rule C-OPEN between C′i (input) and A′i (output), i.e. \( D'_i = \text{in}^0(c,x).P_1 | P_2, A'_i = \nu\tilde{r} \cdot (\text{out}^0(c,d).Q_1 | Q_2) \) and \( A'' \equiv \nu\tilde{r} \cdot (P_1^{d/x} | P_2 | Q_1 | Q_2 | \omega d) \). Lets us do a case analysis on whether (6.a) \( d \in \tilde{r} \) or (6.b) \( d \notin \tilde{r} \). Note that Case (6.a) is in fact almost identical to Case (5.a) and that the result holds with \( C'[\_] = \nu\tilde{r}_1.\nu\tilde{r} \cdot (P_1^{d/x} | P_2 | \omega d | \_), A' = \nu\tilde{r} \cdot (Q_1 | Q_2) \) and \( tr = tr.\nu d \) with \( \nu\tilde{r} = \nu\tilde{r}.\nu d \). Furthermore, note that Case (6.b) is also very similar to Case (5.b) and that the result holds with \( C'[\_] = \nu\tilde{r}_1.\nu\tilde{r} \cdot (P_1^{d/x} | P_2 | \omega d | \_) \) and \( A'' = \nu\tilde{r} \cdot (Q_1 | Q_2) \). Notice that in both cases \( C'[\_] \) is p-closing for \( A' \) and \( B' \) since \( \omega d \) was added to \( C'[\_] \).

Case 7, rule C-OPEN between A′i (input) and C′i (output), i.e. \( D'_i = \text{out}^0(c,d).P_1 | P_2, A'_i = \nu\tilde{r} \cdot (\text{in}^0(c,x).Q_1 | Q_2) \) and \( A'' \equiv \nu\tilde{r}_1.\nu\tilde{r} \cdot (P_1 | P_2 | Q_1^{d/x} | Q_2 | \omega d) \). This case if very similar to Case 4 when M is of channel type and the result holds with \( C'[\_] = \nu\tilde{r}_1.\nu\tilde{r} \cdot (P_1 | P_2 | \omega d | \_) \) and \( A' = \nu\tilde{r} \cdot (Q_1^{d/x} | Q_2) \). Notice that in both cases \( C'[\_] \) is p-closing for \( A' \) and \( B' \) since \( \omega d \) was added to \( C'[\_] \).

Case 8, rule C-PRIV with a communication on a channel c. Notice that this rule is in fact partially covered by the beginning of the proof. Indeed, Case 1 and 3 cover the cases where c is not in \( \tilde{n}_1 \).

We have concluded the proof for the case \( s = p \) hence it remains the case \( s = e \). Once, again several cases are already covered since \( \tilde{\Delta}_p \subset \tilde{\Delta}_e \). Hence we only need to focus on the cases of the rules C-EAV and C-OEAV:

Case 8, rule C-EAV, i.e. \( A'_1 = \nu\tilde{r} \cdot \text{out}^0(c,u).Q_1 | \text{in}^0(c,x).Q_2 | Q_3, D'_1 = \text{eav}(c,y).P_1 | P_2, A'' \equiv \nu\tilde{r}_1.\nu\tilde{r} \cdot (P_1^{u/y} | P_2 | Q_1 | Q_2^{u/x} | Q_3) \) and \( u \) is of base type (We assume w.l.o.g. that the names and variables in \( \tilde{r} \) not in \( D'_1 \)). Note that in such a case \( c \notin \tilde{r} \). Moreover, note this is the only possible combination of input and output since \( C'_1 \) is an attacker evaluation context and \( A'_1 \) is an honest extended process. Let us consider a fresh variable z. Hence \( A'_1 \xrightarrow{\nu z,\text{eav}(c,z)} \nu\tilde{r}.(Q_1 | Q_2^{u/x} | Q_3 | \{^{u/z}\}) \). But since z is fresh, we deduce that \( A'' \equiv \nu\tilde{r}_1.\nu\tilde{r} \cdot (P_1^{u/y} | P_2 | Q_1 | Q_2^{u/x} | Q_3 | \{^{u/z}\}) \). Let \( C'[\_] = \nu\tilde{r}_1.\nu z \cdot (P_1^{u/y} | P_2 | \_), A' = \nu\tilde{r} \cdot (Q_1 | Q_2^{u/x} | Q_3 | \{^{u/z}\}) \).
$Q_3 \mid \{^n/z\}$ and $\text{tr} = \text{tr}_1.\nu z.\text{eav}(c, z)$. We have $A'' \equiv C'[A']$ and $A \equiv A'$. Let $B'$ be a closed extended process such that $B \xrightarrow{\text{r}_e} A' \phi(B) \xrightarrow{\alpha} \phi(A')$. We have that there exists $B'_1$ such that $B \xrightarrow{\text{r}_e} B'_1 \xrightarrow{\nu z.\text{eav}(c, z)} e \xrightarrow{\nu} B'$. By renaming, we can assume that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once. Since $\phi(B) \sim \phi(A')$, we deduce that $\text{dom}(B'_1) = \text{dom}(A'_1)$ and $\phi(B'_1) \sim \phi(A'_1)$. Thus, we can apply our induction hypothesis on $B'_1$.

This allows us to deduce that $C'[B'_1] \xrightarrow{\nu} C'[B'_1]_e$ and $C'[\_]$ is $\text{e}$-closing for $B'_1$. In order to conclude, it remains to show that $C'[B'_1] \xrightarrow{\nu} C'[B'_2]$ and $C'[\_]$ is $\text{p}$-closing for $B'_2$ (since $C'[\_]$ is $\text{e}$-closing for $B'_2$ and $B'_2 \xrightarrow{\nu} B'$ implies $C'[B'_2] \xrightarrow{\nu} C'[B']$ and $C'[\_]$ is $\text{p}$-closing for $B'$).

We have seen that $B'_1 \xrightarrow{\nu z.\text{eav}(c, z)} e B'_2$. Hence, $B'_1 = \nu \mu n.(\text{out}^b(c, N), Q_1) \mid \text{in}(c, x) Q_2 \mid Q_3$ for some $\mu n$. $N$ is of base type, $Q_1$, $Q_2$, $Q_3$ and $B'_2 \equiv \nu \mu n.(Q_1' \mid Q_2' \{N/x\} \mid Q_3' \mid \{N/y\})$. Since we assume that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once, we obtain that $C'[B'_1] \equiv \nu \mu n.\nu \mu n.(\text{eav}(c, y), P_1) \mid P_2 \mid \text{out}^b(c, N), Q_1' \mid \text{in}(c, x) Q_2' \mid Q_3'$. Moreover, since $z$ is fresh, we obtain $\nu \mu n.\nu \mu n.\nu \mu n.\nu \mu n.\nu \mu n.(\text{eav}(c, y), P_1) \mid P_2 \mid \text{out}^b(c, z), Q_1' \mid \text{in}(c, x) Q_2' \mid Q_3' \mid \{N/y\}) \equiv C'[\nu \mu n.(Q_1' \mid Q_2' \mid \{N/y\}) \mid Q_3' \mid \{N/y\}) \equiv C'[B'_2]$. Note that since the rule is focused on base type terms, we directly have that $C'[\_]$ is $\text{e}$-closing for $B'_2$.

Case 9, rule C-OEAV, i.e. $A'_1 = \nu r.(\text{out}^b(c, d), Q_1) \mid \text{in}^b(c, x) Q_2 \mid Q_3$, $D'_1 = \text{eav}(c, y), P_1 \mid P_2$, $A'' \equiv \nu \mu n.\nu \mu n.(P_1 \{d/y\}) \mid P_2 \mid Q_1 \mid Q_2 \{d/x\} \mid Q_3 \mid \omega d$ and $d$ is of channel type (We assume w.l.o.g. that the names and variables in $\tilde{r}$ are not in $D'$). We have to do a case analysis on $d$:

- Case $d \notin \tilde{r}$: Let us denote $\nu r = \nu r'.\nu d$. In such a case $A'_1 \xrightarrow{\text{vd.eav}(c, d)} \nu r'.(Q_1 \mid Q_2 \{d/x\} \mid Q_3)$. But we know that the names and variables in $\tilde{r}$ are not in $D'_1$ hence $A'' \equiv \nu \mu n.\nu d.\nu r'.(Q_1 \mid Q_2 \{d/x\} \mid Q_3)$.

We have seen that $B'_1 \xrightarrow{\text{vd.eav}(c, d)} B'_2$. Hence, $B'_1 = \nu \mu n.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.(Q_1 \mid Q_2 \{d/x\} \mid Q_3)$ for some $\mu n$. $Q_1$, $Q_2$, $Q_3$ and $B'_2 \equiv \nu \mu n.(Q_1' \mid Q_2' \{d/x\} \mid Q_3)$. But since we assume that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once, we obtain that $C'[B'_1] \equiv \nu \mu n.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.(Q_1' \mid Q_2' \{d/x\} \mid Q_3') \mid \omega d).$ Hence $C'[B'_1] \xrightarrow{\nu r} C'[B'_1]$ and $C'[\_]$ is $\text{e}$-closing for $B'_1$. In order to conclude, it remains to show that $C'[B'_1] \xrightarrow{\nu} C'[B'_2]$ and $C'[\_]$ is $\text{e}$-closing for $B'_2$ (since $C'[\_]$ is $\text{e}$-closing for $B'_2$ and $B'_2 \xrightarrow{\nu} B'$ implies $C'[B'_2] \xrightarrow{\nu} C'[B']$ and $C'[\_]$ is $\text{e}$-closing for $B'$).

We have seen that $B'_1 \xrightarrow{\text{vd.eav}(c, d)} B'_2$. Hence, $B'_1 = \nu \mu n.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.(Q_1 \mid Q_2 \{d/x\} \mid Q_3)$ for some $\mu n$. $Q_1$, $Q_2$, $Q_3$ and $B'_2 \equiv \nu \mu n.(Q_1' \mid Q_2' \{d/x\} \mid Q_3)$.

Let $C'[\_] = \nu k'.\nu r'.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.(Q_1 \mid Q_2 \{d/x\} \mid Q_3 \mid \omega d).$ Hence $C'[B'_1] \equiv \nu \mu n.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.(Q_1' \mid Q_2' \{d/x\} \mid Q_3' \mid \omega d)$.

Note that $d$ is possible a new free channel of $B'_2$. However, since we have $\omega d$ in $C'$, we ensure that $C'$ is $\text{e}$-closing for $B'_2$.

- Case $d \notin \tilde{r}$: In such a case $A'_1 \xrightarrow{\text{eav}(c, d)} \nu r.(Q_1 \mid Q_2 \{d/x\} \mid Q_3)$. But we know that the names and variables in $\tilde{r}$ are not in $D'_1$ hence $A'' \equiv \nu \mu n.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.(Q_1 \mid Q_2 \{d/x\} \mid Q_3)$. Let $C'[\_] = \nu k'.\nu r'.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.(Q_1 \mid Q_2 \{d/x\} \mid Q_3)$ and $\text{tr} = \text{tr}_1.\text{eav}(c, d)$. But we know that the names and variables in $\tilde{r}$ are not in $D'_1$ hence $A'' \equiv \nu \mu n.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.(Q_1 \mid Q_2 \{d/x\} \mid Q_3)$. Let $C'[\_] = \nu k'.\nu r'.\nu d.\nu r'.\nu d.\nu r'.\nu d.\nu r'.(Q_1 \mid Q_2 \{d/x\} \mid Q_3)$.
We have $A'' \equiv C'[A']$ and $A \Downarrow_c A'$. Let $B'$ be a closed extended process such that $B \Downarrow_c B'$ and $\phi(B') \sim \phi(A')$. We have that there exists $B'_1$ such that $B \Downarrow_{\nu m}(c,d) e_{\text{eav}} B'_1 \Downarrow_{\nu m}(c,d) e_{\text{eav}} B'_2 \Downarrow_B B'$. By renaming, we can assume that the bound names of $B'_1$ are distinct from the names of $C'_1$ and are bound only once. Since $\phi(B') \sim \phi(A')$, we deduce that $\text{dom}(B'_1) = \text{dom}(A'_1)$ and $\phi(B'_1) \sim \phi(A'_1)$. Thus, we can apply our induction hypothesis on $B'_1$. This allows us to deduce that $C'[B] \Downarrow_{\nu \tilde{m}} C'[B'_1]$ and $C'[\_] \Downarrow_{\nu \tilde{m}} B'$ implies $C'[B'_2] \Downarrow_{\nu \tilde{m}} C'[B']$ and $C'[\_] \Downarrow_{\nu \tilde{m}} B'$.

We have seen that $B'_1 \Downarrow_{\nu \tilde{m}} B'_2$. Hence, $B'_1 = \nu \tilde{m}.(\text{out}_h(c,d).Q'_1 \mid \text{in}(c,x)Q'_2 \mid Q'_3)$ with $d \notin \tilde{m}$ for some $\tilde{m}, Q'_1, Q'_2, Q'_3$ and $B'_2 = \nu \tilde{m}.(Q'_1 \mid Q'_2[^d/x] \mid Q'_3)$ with $x$ and $y$ distinct from the names of $C'_1$ and are bound only once.

Let $C'_1[B'_1] \Downarrow_{\nu \tilde{m}} \nu \tilde{m}.(\text{eav}(c,y).P_1 \mid P_2 \mid \text{out}_h(c,d).Q_1 \mid \text{in}(c,x)Q_2 \mid Q_3)$. Hence $C'_1[B'_1] \Downarrow_{\nu \tilde{m}} \nu \tilde{m}.(P_1[^d/y] \mid P_2 \mid Q_1 \mid Q_2[^d/x] \mid Q_3 \mid \text{out}_h(c,y)) \Downarrow_{\nu \tilde{m}} C'[B'_2]$.

Note that $d$ is a name that occurs in $\text{fn}(A) \cup \text{fn}(B)$ and not in $\text{bn}(A) \cup \text{bn}(B)$, and $u$ be a fresh name. For all $s \in \{c,p,e\}$, we have $A^{[u'/u]} \Downarrow_{\nu \tilde{m}} B^{[u'/u]}$.

**Proof.** By induction on the derivation. \(\Box\)

The previous lemma indicates that the trace equivalence are preserved by replacement of free names.

As for the previous proposition, the proof of Theorem 1 is taken from [19] for the classical semantics and we adapt it for the private and eavesdropping semantics.

**Theorem 1.** $\approx^*_{\nu} \subseteq \approx^*_{\nu m}$ and $\approx^*_{\nu} \equiv \approx^*_{\nu m}$ on image-finite processes for $s \in \{c,e,p\}$.

**Proof.** We first prove that for all $s \in \{c,p,e\}$, $\approx^*_{\nu} \subseteq \approx^*_{\nu m}$. Since we already proved in the body of the paper that there exist two closed extended processes such that $A \approx_{\nu m} B$ but $A \not\approx_{\nu} B$, we would thus obtain that $\approx^*_{\nu} \not\subseteq \approx^*_{\nu m}$.

Let $A, B$ be two closed extended processes such that $A \approx^*_{\nu} B$. Let $C[\_]$ be an evaluation context $s$-closing for $A$ and $B$, and $c$ be a channel name. We assume w.l.o.g. that $C[\_] = \nu \tilde{m}.(D_1 \mid \nu \tilde{m}.(\_ \Downarrow D_2))$ for some extended processes $D, D'$ and for some sequences of names and variables $\tilde{m}$, and $\tilde{n}$. We assume w.l.o.g. that $\tilde{m} \cap (\text{bn}(A) \cup \text{bn}(A)) = \emptyset$ and $\tilde{m} \cap (\text{bn}(B) \cup \text{bn}(B)) = \emptyset$.

Let $A_2 = A^{[\tilde{m}'/\tilde{m}]}$ and $B_2 = B^{[\tilde{m}'/\tilde{m}]}$ where $m'$ is a sequence of fresh names and variables. Thanks to Lemma 10, we have that $A_2 \approx^*_{\nu} B_2$. Hence, by structural equivalence, there exists $C_2[\_] = \nu \tilde{m}.(D \mid \_)$ such that $C[A] \equiv C_2[A_2]$ and $C[B] \equiv C_2[B_2]$.

Assume now that $C[A] \Downarrow^*_{\nu}$. This means that there exist a evaluation context $C_1$ that does not bind $c$, a term $M$, and a plain process $P, \theta \in \{\text{at,ho}\}$ such that $C[A] \equiv C_2[A_2] \Downarrow^*_{\nu} C_1[\text{out}_h(c,M).P]$. Applying Proposition 4 on $A_2, B_2$ and $C_2[\_], we know that there exist a closed extended process $A_2$, an evaluation context $C_2[\_] = \nu \tilde{m}.(E \mid \_) s$-closing for $A_2$ and $\text{tr} \in \{A \setminus \{\tau\}\}^*$.
such that $C_1[\mathrm{out^θ}(c,M).P] \equiv C_2[A'_2]$ and $A_2 \not\vdash_\ast A'_2$, and for all closed extended process $B'_2$ such that $B_2 \not\vdash_\ast B_2'$ and $\phi(B'_2) \sim \phi(A'_2)$, we have that $C_2[B_2] \rightarrow_\ast C_2[B'_2]$. Moreover, we assume w.l.o.g. that $bn((\mathsf{tr}) \cap fn((B_2)) = \emptyset$.

Since $C'_2 = \nu \tilde{r}.(E|\_)$, we can deduce from $C_1[\mathrm{out^θ}(c,M).P] \equiv C'_2[A'_2]$ that the output $\mathrm{out^θ}(c,M)$ comes from the process $E$ when $\theta = \text{at}$ or from $A'_2$ when $\theta = \text{ho}$. We distinguish these two cases:

- **Case $\theta = \text{at}$:** Since, we have that $A_2 \not\approx^s B_2$, we know that that there exists $B'_2$ such that $B_2 \not\vdash_\ast B_2'$ and $\phi(A'_2) \sim \phi(B'_2)$. Therefore, we have that $C_2[B_2] \rightarrow^s C'_2[B'_2] \equiv \nu \tilde{r}.(E|B'_2)$. But by hypothesis, we know that the output $\mathrm{out^θ}(c,M)$ comes from $E$ and $c \not\in \tilde{r}$. Hence we have that $C_2[B_2] \not\vdash_\ast$, and since $C[B] \equiv C_2[B_2]$, we conclude that $C[B] \not\vdash_\ast$.

- **Case $\theta = \text{ho}$:** Thus, we have that $A'_2 \equiv \nu \tilde{v}.(\mathrm{out^θ}(c,M).P | A_3)$ with $c \not\in \tilde{v}, \tilde{r}$. Thus, we have that $A'_2 \not\approx^s B_2$, we have that there exists $B'_2$ such that $B_2 \not\vdash^s B_2'$ and $\phi(A') \sim \phi(B'_2)$. Since internal reduction rules do not modify the frame (modulo structural equivalence), we can deduce w.l.o.g. that there exists $B'$ such that $B_2 \not\vdash^s B'$ and $\phi(A') \sim \phi(B')$. Therefore, we have that there exists a term $N$, an evaluation context $C_3$ and a process $Q$ such that $B' \equiv C_3[\mathrm{out^θ}(c,N).Q]$ and $c$ is not bind by $C_3$. Furthermore, we have that $\phi(A'_2) \sim \phi(B')$ which means that $C_i[B_2] \rightarrow^s C'_2[B'_2]$, and thus $C_i[B_2] \not\rightarrow^s C'_2[\mathrm{out^θ}(c,N).Q]$. Hence, we have that $C_i[B_2] \not\vdash_\ast$, and since $C[B] \equiv C_2[B_2]$, we conclude that $C[B] \not\vdash_\ast$.

This concludes the proof of $\approx^s \subseteq \approx^s_m$. It remains to prove that on imagine-finite processes, $\approx^s = \approx^s_m$ for all $s \in \{ c, e, p \}$. We first focus on $s = c$.

Assume that $A \not\approx^s B$. We assume w.l.o.g. that $A \not\approx^s c, B$. In such a case, there exists a witness for the non equivalence. This means that there exists $A', \mathsf{tr}$ such that $bn((\mathsf{tr}) \cap fn((B)) = \emptyset$, and for all $B'$, $B \not\vdash^c c B'$ implies $\phi(A') \not\sim \phi(B')$. Moreover, we assume that no name in $\mathsf{tr}$ is bound twice (i.e. $\nu \alpha$ can not occur twice in $\mathsf{tr}$) and bound names in $\mathsf{tr}$ are distinct from free names that occur in $A$, $B$, and $\mathsf{tr}$.

We build an evaluation context $C_4[\_\_\_\_\_\_\_\_\_]$ according to the trace $\mathsf{tr}$ and also the tests that witness the fact that static equivalence does not hold. Let $S_{\mathsf{tr}} = \{ \phi(B') | B \not\vdash^c c B' \}$. Since $B$ is image-finite, we know that $S_{\mathsf{tr}}/\approx$ is finite. Let $\{ \phi_1, \ldots, \phi_m \} = S/\approx$. Note that $m$ can be equal to 0 if there is no $B'$ such that $B \not\vdash^c c B'$.

We know that $\{1, \ldots, m\} = T^+ \uplus T^-$ with:

- for each $i \in T^+$, there exist two terms $M_i$ and $N_i$ such that $v(M_i) \cup v(N_i) \subseteq \mathrm{dom}(\phi(A'))$, $(M_i \not\vdash^E N_i) \phi(A')$, and $(M_i \not\vdash^E N_i) \phi_i$; and
- for each $i \in T^-$, there exist two terms $M_i$ and $N_i$ such that $v(M_i) \cup v(N_i) \subseteq \mathrm{dom}(\phi(A'))$, $(M_i \not\vdash^E N_i) \phi(A')$, and $(M_i =^E N_i) \phi_i$.

Let $\text{bad}$ be a fresh channel name that does not occur in $A$ and $B$. Let $P_1, \ldots, P_m, P_{m+1}$ be the plain processes defined as follows:

- $P_{m+1} = \mathrm{out}^{\text{at}}(\text{bad}, \text{bad}).0$
• for $1 \leq i \leq m$, we define $P_i$ as follows:

$$P_i \triangleq \begin{cases} M_i = N_i \text{ then } P_{i+1} \text{ else } 0 \text{ when } i \in T^+ \\
M_i = N_i \text{ then } 0 \text{ else } P_{i+1} \text{ when } i \in T^- 
\end{cases}$$

Let $\{a_1, \ldots, a_k\}$ be channel names that occur free in $A$, $B$, and $tr$. Let $X_{ch}^0 = \{x_{a_1}, \ldots, x_{a_k}\}$ be a set of variables of channel type, and $\sigma = \{x_{a_1} \mapsto a_1, \ldots, x_{a_k} \mapsto a_k\}$. Moreover, for all channel names $\{d_1, \ldots, d_m\}$ that are bound in $tr$, we also associate fresh variables $x_{d_1}, \ldots, x_{d_m}$.

We define $C_c[\_]$ such that $C_c[\_] = \nu\tilde{\sigma}.(Q_c(tr, X_{ch}^0) \mid \_)$ where $\tilde{\sigma} = dom(\phi(A))$ and $Q_c(tr, X_{ch})$ is defined by recurrence on $tr$ as follows:

• if $tr = \epsilon$ then $Q_c(tr, X_{ch}) = P_1$;

• if $tr = in(a, M).tr'$ then $Q_c(tr, X_{ch}) = out^x(x_0, M).Q_c(tr', X_{ch})$;

• if $tr = \nu z.\_a(z).tr'$ and $z$ is of base type then $Q_c(tr, X_{ch}) = in^x(x_0, x).Q_c(tr', X_{ch})$;

• if $tr = out(a, c).tr'$ then $Q_c(tr, X_{ch}) = in^x(x_0, y).if \ y = x_0 then Q_c(tr', X_{ch}) else 0$ where $y$ is fresh variable of channel type; and

• if $tr = \nu c.out(a, c)$ and $c$ is of channel type then $Q_c(tr, X_{ch}) = in^x(x_0, x_c).if \ x_c \in X_{ch}\sigma \text{ then } 0 \text{ else } Q_c(tr', X_{ch})$ where $X_{ch} = X_{ch} \cup \{x_c\}$.

We use the conditional if $u \in \{u_1, \ldots, u_k\}$ then 0 else $P$ as a shortcut for

if $u = u_1$ then 0 else (if $u = u_2$ then 0 else $\ldots$ (if $u = u_k$ then 0 else $P$) $\ldots$).

We can see that $C_c[A] \not\preceq \text{bad}$ since $A = A' \not\lessgtr A' \land \phi(A')$ satisfies by definition all the tests that are tested in $P_1, \ldots, P_m$. However, by construction of $C_c[\_]$, we have that $C_c[B] \not\preceq \text{bad}$.

This conclude the proof for the case $s = c$. The proof for $s = p$ and $e$ are very similar. We only need to slightly modify the context $C_c[\_]$. In fact since the possible labels in the private semantics are the same as in the original semantics, we have $C_p[\_] = C_c[\_]$. However, for the eavesdropping semantics, we define $C_e[\_]$ such that $C_e[\_] = \nu\tilde{\tau}.(Q_e(tr, X_{ch}^0) \mid \_)$ where $\tilde{\tau} = dom(\phi(A))$ and $Q_e(tr, X_{ch})$ is defined by recurrence on $tr$ as follows:

• if $tr = \epsilon$ then $Q_e(tr, X_{ch}) = P_1$;

• if $tr = in(a, M).tr'$ then $Q_e(tr, X_{ch}) = out^x(x_0, M).Q_e(tr', X_{ch})$;

• if $tr = \nu z.\_a(z).tr'$ and $z$ is of base type then $Q_e(tr, X_{ch}) = in^x(x_0, x).Q_e(tr', X_{ch})$;

• if $tr = out(a, c).tr'$ then $Q_e(tr, X_{ch}) = in^x(x_0, y).if \ y = x_0 then Q_e(tr', X_{ch}) else 0$ where $y$ is fresh variable of channel type; and

• if $tr = \nu c.out(a, c)$ and $c$ is of channel type then $Q_e(tr, X_{ch}) = in^x(x_0, x_c).if \ x_c \in X_{ch}\sigma \text{ then } 0 \text{ else } Q_e(tr', X_{ch})$ where $X_{ch} = X_{ch} \cup \{x_c\}$.

$\square$
Appendix D. Proof of Theorem 6

In this section, we want to prove that $\approx_{m}^{e} \subseteq \approx_{m}^{a} \cap \approx_{m}^{e}$. In order to show that $\approx_{m}^{e} \subseteq \approx_{m}^{e}$, we need to build a transformation of context that would allow us to go from the classic semantics to eavesdropping semantics, and vice versa.

Notice that in the definition of structural equivalence $!A \parallel !A$ is not equivalence to $!A$ even though they have the same behavior. In fact, for reachability, may equivalence, trace equivalence, observational equivalence and labeled bisimilarity, using the structural equivalence coincides with using the struutural equivalence augmented with the equality $!A \parallel !A \equiv !A$. As such in this section, we will consider the structural equivalence augmented with the equality $!A \parallel !A \equiv !A$.

Definition 13. Let $P$ be an extended attacker process. We define $\overline{P}$ inductively as follows:

1. $0$ when $P = 0$
2. $\overline{P_1} \mid \overline{P_2}$ when $P = P_1 \mid P_2$
3. $P$ when $P = \{u/x\}$
4. $\omega c$ when $P = \omega c$
5. $\nu n.(\overline{P} \parallel \text{leav}(n, y))$ when $P = \nu n.P'$, $n$ is of channel type and $y, z$ are variables of base and channel type respectively.
6. $\nu k.\overline{P}$ when $P = \nu n.P'$, $n$ is of base type
7. if $u = v$ then $\overline{P_1}$ else $P_2$ when $P = $ if $u = v$ then $P_1$ else $P_2$
8. $\text{eav}(c, x).\overline{P}$ when $P = \text{eav}(c, x).P'$
9. $\text{out}^{t}(c, u).\overline{P}$ when $P = \text{out}^{t}(c, u).P$
10. $\text{in}^{t}(c, x).\overline{P}$ when $P = \text{in}^{t}(c, x).P'$ and $x$ is of base type
11. $\text{in}^{t}(c, x).(\overline{P} \parallel \text{leav}(x, y))$ when $P = \text{in}^{t}(c, x).P', y, z$ are variables of base and channel type respectively.

Let $\mathcal{T}_{ch}$ be the terms of channel type, i.e. names and variables of channel type. Let $C[\_] = \nu n.(D \mid \_)$ be an attacker evaluation context and $S$ a set of channel names. We define $\overline{C[\_]}$ as follows:

$$\nu n.((D \mid \_ \mid \prod_{a \in \check{\mathcal{T}}_{ch}} \text{leav}(a, y) \parallel \text{leav}(a, z) \mid \omega a) \mid \prod_{a \in \check{S}} \text{leav}(a, y) \parallel \text{leav}(a, z) \mid \omega a)$$

where $y$ and $z$ are variables of base and channel type respectively.

In order to facilitate the readability of the proof, for a set $S$ of names and variables, we will denote $\mathbb{P}(S) = \prod_{a \in S \cap \mathcal{T}_{ch}} \text{leav}(a, y) \parallel \text{leav}(a, z)$ and $\mathbb{P}_{o}(S) = \prod_{a \in S \cap \mathcal{T}_{ch}} \text{leav}(a, y) \parallel \text{leav}(a, z) \mid \omega a$. Moreover, we will consider that $\mathbb{P}(S) \parallel \mathbb{P}(S) \equiv \mathbb{P}(S)$.

Hence, $\overline{C[\_]}$ can now be written as $\nu n.((D \mid \_ \mid \mathbb{P}_{o}(\check{n})) \mid \mathbb{P}_{o}(S))$.

Note that from the definition, we have that for all $A$ closed honest extended process, if $C[\_] = \nu n.(D \mid \_)$ is c-closing for $A$ then $\overline{C[\_]}$ is e-closing for $A$ for all $S$.

Lemma 11. Let $A$ be an extended process and $\nu n$ a sequence of names and variables. We have $\nu n.A \equiv \nu n.(\overline{A} \parallel \mathbb{P}(\check{n}))$.

Proof. Direct from the definition. □
Lemma 12. Let $A$ be an closed honest extended process. Let $C[\_] = \nu n. (\nu m. D | \_)$ be an attacker evaluation context $c$-closing for $A$ such that $D$ is named-cleaned and eavesdrop-free. Let $S$ be a set of channel names such that $fc(C[A]) \subseteq S$.

(1) For all $C[A] \rightarrow_r A_0$, there exist $A'$ closed honest extended process, $C'[\_] = \nu n'. (\nu m'. D' | \_)$ an attacker evaluation context $c$-closing for $A'$ such that $D'$ is name-cleaned and eavesdrop-free, $C'[A'] = A_0$ and $\overline{C}[A] \rightarrow_r \overline{C}[A']$.

(2) For all $\overline{C}[A] \rightarrow_r A_0$, there exist $A'$ closed honest extended process, $C'[\_] = \nu n'. (\nu m'. D' | \_)$ an attacker evaluation context $c$-closing for $A'$ such that $D'$ is name-cleaned and eavesdrop-free, $\overline{C}[A] \equiv A_0$ and $C[A] \rightarrow_c C'[A']$.

Proof. We first start by proving the first property. Notice that by structural equivalence, we can always assume that the bound names and variables in $C[A]$ are only bound once and are distinct from the free names in $S$. Indeed, for all $C'[\_], A'$, if $C[A] \equiv C'[A']$ only by renaming of bound names and variables then we obtain that $\overline{C}[A] \equiv \overline{C}[A']$.

We do a case analysis on the internal rule applied.

Case 1.a, rule THEN on $D$, i.e. $D = u = v$ then $D_1$ else $D_2 | D_3$ and $A_0 \equiv \nu n. (D_1 | D_3 | A)$: In such a case we have $D \rightarrow e \frac{D_1}{D_3}$ and so $\nu n. (D | P(n)) \rightarrow e \nu n. (D_1 | D_3 | P(n))$. By Lemma 11, we obtain that $\nu n. D \rightarrow \nu n. (D_1 | D_3 | \_)$ and $A' = A$. Since $\overline{C}[\_] = \nu n. (D_1 | D_3 | \_ | P_o(n)) | P_o(S)$, we obtain that $A_0 \equiv C'[A']$ and $\overline{C}[A] \rightarrow_r \overline{C}[A']$.

Case 1.b, rule ELSE on $D$: Similar to Case 1.a.

Case 2.a, rule THEN on $A$, i.e. $A \equiv \nu r. (if u = v then P_1 else P_2 | P_3)$ and $A_0 \equiv C[\nu r. (P_1 | P_3)]$: In such a case, let us denote $C'[\_] = C[\_]$ and $A' = \nu r. (P_1 | P_3)$. Therefore, $C'[A'] = C[\_] \equiv A_0$. Note that $A \rightarrow_r A'$. Hence $C[A] \rightarrow_r C'[A']$ and $\overline{C}[A] \rightarrow_r \overline{C}[A']$. Thus the result holds.

Case 2.b, rule ELSE on $A$: Similar to Case 2.a.

Case 3, rule Comm on $A$, i.e. $A \equiv \nu r. (\text{out}(c,u).P_1 | \text{in}(c,x).P_2 | P_3)$ and $A_0 \equiv C[\nu r. (P_1 | P_2^{\langle u / x \rangle} | P_3)]$: Note that even though $A \rightarrow_r \nu r. (P_1 | P_2^{\langle u / x \rangle} | P_3)$, we don’t necessarily have that $A \rightarrow_r \nu r. (P_1 | P_2^{\langle u / x \rangle} | P_3)$.

We have to make a case analysis on $u$ and $c$:

• Case 3.a, $c \in r$: In such a case, we know from $A$ being an honest processes that $c \notin oc(P_3)$. Thus we can apply rule C-PRIV to obtain that $A \rightarrow_r \nu r. (P_1 | P_2^{\langle u / x \rangle} | P_3)$. Hence, by denoting $C'[\_] = C[\_]$ and $A' = \nu r. (P_1 | P_2^{\langle u / x \rangle} | P_3)$, we obtain that $C'[A'] = C[\_] \equiv A_0$. $A \rightarrow_r A'$ and so $C[A] \rightarrow_r C[A']$ and $\overline{C}[A] \rightarrow_r \overline{C}[A']$. Therefore, the result holds.

• Case 3.b, $c \notin r$ and $u$ of base type: In such a case, $\text{out}(c,u).P_1 | \text{in}(c,x).P_2 | P_3$ or $\text{eav}(c,y) \rightarrow_r P_1 | P_2^{\langle u / x \rangle} | P_3$ by rule C-EAV. Let us denote $A' = \nu r. (P_1 | P_2^{\langle u / x \rangle} | P_3)$. Since $c \notin r$, we obtain that $A \rightarrow_r \nu r. (P_1 | P_2^{\langle u / x \rangle} | P_3)$ and so $A \| eav(c,y) \rightarrow_r A' \| eav(c,y)$. By noticing that $c$ is either in $\bar{n}$ or in $fc(C[A])$ and in $S$, the structural equivalence gives us that $\overline{C}[A] \rightarrow_r \overline{C}[A']$. Hence the result holds with $C'[\_] = C[\_]$. 


Let us do a case analysis on $\nu$. Case 5, rule $\textsc{Comm}$ on $D$, i.e. $D = \text{out}^c(c,u).D_1 | \text{in}^c(x).D_2 | D_3$ and $A_0 \equiv \nu n.(\nu m.(D_1 | D_2^{u/x} | D_3) | A)$. Let us do a case analysis on $u$:

- Case 3.c, $c \notin \hat{r}$ and $u$ of channel type: This case is very similar to Case 3.b. Indeed, $\text{out}^c(c,u).P_1 | \text{in}^c(c,x).P_2 | P_3 | \text{eav}(c,z) \Rightarrow P_1 | P_2^{x/z} | P_3 | \omega c$ by the rule $\text{C-OEav}$. Let us denote $A' = \nu r.(P_1 | P_2^{x/z} | P_3)$. Since $c \notin \hat{r}$, we obtain that $A \Rightarrow A'$ by $\omega c$ and so $A \Rightarrow \text{eav}(c,z) | \omega c \Rightarrow A'$ by $\text{eav}(c,z) | \omega c$. By noticing that $c$ is either in $\hat{n}$ or in $fc(C[A])$ and so in $S$, the structural equivalence gives us that $\overline{C}_S[A] \Rightarrow \overline{C}_S[A']$. The result holds with $C'[\_] = C[\_]$.

Case 4, rule $\textsc{Comm}$ on $D$, i.e. $D = \text{out}^c(c,u).D_1 | \text{in}^c(x).D_2 | D_3$ and $A_0 \equiv \nu n.(\nu m.(D_1 | D_2^{u/x} | D_3) | A)$. Let us do a case analysis on $u$:

- Case 4.a, $u$ is of base type: In such a case, we have $\text{out}^c(c,u).D_1 | \text{in}^c(c,x).D_2 | D_3$ by the rule $\text{C-Env}$. Hence, $\nu n.(\text{out}^c(c,u).D_1 | \text{in}^c(c,x).D_2 | D_3 | \text{P}(\hat{m})) \Rightarrow e \nu n.(D_1 | D_2^{u/x} | D_3 | \text{P}(\hat{m}))$. Let us denote $D' = (D_1 | D_2^{u/x} | D_3)$. By Lemma 11, we obtain that $\nu m.\text{D} \Rightarrow e \nu m.\text{D'}$. Hence, we deduce $\nu n.(\nu m.\text{D} | A | \text{P}_o(\hat{n})) | \text{P}_o(S) \Rightarrow e \nu n.(\nu m.\text{D'} | A | \text{P}_o(\hat{n})) | \text{P}_o(S)$. Let us denote $C'[\_] = \nu n.(\nu m.\text{D'} | A) | \text{P}_o(\hat{n})$. We have $A_0 \equiv C'[A]$ and $\overline{C}_S[A] \Rightarrow e C'[A']$. Hence the result holds.

- Case 4.b, $u$ is of channel type and $u \notin \hat{n} \cup \hat{n}$: In such a case, $u \in f v(C[A]) \subseteq S$ and we have $\text{out}^c(c,u).D_1 | \text{in}^c(c,x).D_2 | \text{P}(x)) | D_3$ by the rule $\text{C-OPEN}$. Since $u \notin \hat{m}$, we obtain $\nu n.(\text{out}^c(c,u).D_1 | \text{in}^c(c,x).D_2 | \text{P}(x)) | D_3 | \text{P}(\hat{m})) \Rightarrow e \nu n.(\nu m.\text{D} | D_2^{u/x} | D_3 | \text{P}(\hat{m})) | \text{P}_o(u)$. Let us denote $D' = (D_1 | D_2^{u/x} | D_3)$. By Lemma 11, we obtain that $\nu m.\text{D} \Rightarrow e \nu m.\text{D'}$. Moreover, since $u \notin \hat{n}$ then $\nu n.(\nu m.\text{D} | A | \text{P}_o(\hat{n})) \Rightarrow e \nu n.(\nu m.\text{D'} | A | \text{P}_o(\hat{n})) | \text{P}_o(u)$. Lastly, since $u \in S$ and $\text{P}_o(u) | \text{P}_o(\hat{n}) \Rightarrow e \nu n.(\nu m.\text{D'} | A | \text{P}_o(\hat{n})) | \text{P}_o(S)$ and so the result holds with $A' = A$ and $C'[\_] = \nu n.(\nu m.\text{D'} | A)$.

Case 4.c, $u$ is of channel type and $u \in \hat{n}$: This case is similar to Case 4.b. Since $u \notin \hat{m}$, we can apply the same reasoning and obtain $\nu m.\text{D} \Rightarrow e \nu m.\text{D'} | \text{P}_o(u)$ where $D' = (D_1 | D_2^{u/x} | D_3)$. Since $u \in \hat{n}$ and $\text{P}_o(u) \Rightarrow e \nu n.(\nu m.\text{D} | A | \text{P}_o(\hat{n}))$, we deduce that $\nu n.(\nu m.\text{D} | A | \text{P}_o(\hat{n})) \Rightarrow e \nu n.(\nu m.\text{D'} | A | \text{P}_o(\hat{n}))$. Therefore, we obtain that $\nu n.(\nu m.\text{D} | A | \text{P}_o(\hat{n})) | \text{P}_o(S) \Rightarrow e \nu n.(\nu m.\text{D'} | A | \text{P}_o(\hat{n})) | \text{P}_o(S)$ and so the result holds with $A' = A$ and $C'[\_] = \nu n.(\nu m.\text{D'} | \nu n.(\nu m.\text{D'}))$.

Case 4.d, $u$ is of channel type and $u \in \hat{m}$: First of all, note that since $u \in \hat{m}$, $\nu n.D \equiv \nu u.\nu n.D'$ for some $m'$ such that $u \notin \hat{m'}$. Note that since $u$ is bound, $u \notin f v(A) \cup fn(A)$. Hence, by applying the same reasoning as in Case 4.b, we obtain that $\nu n.(\nu m. D \Rightarrow e \nu m. D') | \text{P}_o(u)$ where $D' = (D_1 | D_2^{u/x} | D_3)$. Since $\text{P}(u) | \text{P}_o(u) \equiv \text{P}(u) | \text{P}(u) \equiv \text{P}_o(u)$, we deduce that $\nu n.(\nu m. D | A | \text{P}_o(\hat{n})) \Rightarrow e \nu n.(\nu m. D' | A | \text{P}_o(\hat{n}))$. Therefore, we denote $\nu n.(\nu n.\nu m. D | A | \text{P}_o(\hat{n})) | \text{P}_o(S) = \nu n.(\nu m. D' | A | \text{P}_o(\hat{n})) | \text{P}_o(S)$ and so the result holds with $A' = A$ and $C'[\_] = \nu n.(\nu m. D' | \nu n.(\nu m. D'))$.

Case 5, rule $\textsc{Comm}$ between $A$ (input) and $D$ (output), i.e. $D = \text{out}^c(c,u).D_1 | D_2, A \equiv \nu r.(\text{in}^c(c,x).P_1 | P_2)$ and $A_0 \equiv \nu n.\nu m.\nu r.(D_1 | D_2 | P_1^{x/z} | P_2)$. Note that $c \notin \hat{m} \cup \hat{r}$. Let us do a case analysis on $u$:
Case 5.a, u is of base type: In such a case, let us split \( \tilde{r} \) and \( \tilde{m} \) in \( \tilde{r}_b, \tilde{r}_c \) and \( \tilde{m}_b, \tilde{m}_c \) respectively, such that \( \tilde{r}_c \) and \( \tilde{m}_c \) are of channel type, and \( \tilde{r}_b \) and \( \tilde{m}_b \) are of base type. Since \( u \) is of base type, we deduce that \( A_0 \equiv \nu \tilde{m}.\nu \tilde{b}.(\nu \tilde{m}.(D_1 | D_2) | P_1) \). Note that \( \text{out}^\text{at}(c, u).D_1 | D_2 | \text{in}^\text{ho}(c, x).P_1 \rightarrow \emptyset.D_1 | D_2 | P_1 \rightarrow P_2 \) by the rule C-ENV. Hence \( \text{out}^\text{at}(c, u).D_1 | D_2 | \text{in}^\text{ho}(c, x).P_1 \rightarrow \emptyset.D_1 | D_2 | P_1 | P_2 \rightarrow P \). Therefore, \( \nu \tilde{m}.\nu \tilde{r}.(\text{out}^\text{at}(c, u).D_1 | D_2 | \text{in}^\text{ho}(c, x).P_1 | P_2 \rightarrow \emptyset.D_1 | D_2 | P_1 | P_2 \rightarrow P \). But \( \nu \tilde{m}.\nu \tilde{r}.(\text{out}^\text{at}(c, u).D_1 | D_2 | \text{in}^\text{ho}(c, x).P_1 | P_2 \rightarrow \emptyset.D_1 | D_2 | P_1 | P_2 \rightarrow P \). According to Lemma 11 and since we assume that bound names and variables are bound once and distinct from free names and variables. Moreover, \( \nu \tilde{m}.\nu \tilde{r}.(\emptyset.D_1 | D_2 | P_1 | P_2 \rightarrow P \). Therefore, let us denote \( \tilde{n}' = \tilde{n}.\tilde{m}_b.\tilde{r}_b, D' = D_1 | D_2 \) and \( A' = \nu \tilde{r}_c.(P_1^{u/x} | P_2) \). Notice that \( \tilde{m}_b \) and \( \tilde{r}_b \) being of base type implies that \( \emptyset.D_1 | D_2 | P_1^{u/x} | P_2 \rightarrow P \). Hence, the result holds with \( C'[\_] = \nu \tilde{m}'.(\nu \tilde{m}.D'|\_). \)

Case 5.b, u is of channel type and \( u \notin \tilde{m} \cup \tilde{n} \): Notice that in such a case \( A_0 \equiv \nu \tilde{m}.(\nu \tilde{m}.(D_1 | D_2) | \emptyset.D_1 | D_2 | P_1^{u/x} | P_2) \). The rest of the proof follows a similar reasoning as in Case 4.b and the result will hold with \( C'[\_] = \nu \tilde{m}.(\nu \tilde{m}.D'|\_).D' = D_1 | D_2 \) and \( A' = \nu \tilde{r}_c.(P_1^{u/x} | P_2) \).

Case 5.c, u is of channel type and \( u \notin \tilde{n} \): Notice that in such a case \( A_0 \equiv \nu \tilde{m}.(\nu \tilde{m}.(D_1 | D_2) | \emptyset.D_1 | D_2 | P_1^{u/x} | P_2) \). The rest of the proof follows a similar reasoning as in Case 4.c and the result will hold with \( C'[\_] = \nu \tilde{m}.(\nu \tilde{m}.D'|\_).D' = D_1 | D_2 \) and \( A' = \nu \tilde{r}_c.(P_1^{u/x} | P_2) \).

Case 5.d, u is of channel type and \( u \notin \tilde{n} \): Note that since \( u \notin \tilde{n} \), \( \nu \tilde{m}.D \equiv \nu \tilde{u}.\nu \tilde{m}.D \) for some \( \tilde{m}' \) such that \( u \notin \tilde{m}' \). Hence, \( A_0 \equiv \nu \tilde{m}.\nu \tilde{u}.(\nu \tilde{m}'.(D_1 | D_2) | \emptyset.D_1 | D_2 | P_1^{u/x} | P_2) \). The rest of the proof follows a similar reasoning as in Case 4.d and the result will hold with \( C'[\_] = \nu \tilde{m}'.(\nu \tilde{m}'.D'|\_).D' = D_1 | D_2 \) and \( A' = \nu \tilde{r}.'(P_1^{u/x} | P_2) \).

Case 6, rule COMM between A (output) and D (input), i.e. \( D = \text{in}^\text{at}(c, x).D_1 | D_2, A \equiv \nu \tilde{r}.(\text{out}^\text{ho}(c, u).P_1 | P_2) \) and \( A_0 \equiv \nu \tilde{m}.\nu \tilde{r}.(\text{in}^\text{ho}(c, x).P_1 | P_2 | P_1 | P_2) \): Note that \( c \notin \tilde{m} \cup \tilde{r} \).

Let us do a case analysis on \( u \):

Case 6.a, u is of base type: In such a case, let us split \( \tilde{r} \) and \( \tilde{m} \) in \( \tilde{r}_b, \tilde{r}_c \) and \( \tilde{m}_b, \tilde{m}_c \) respectively, such that \( \tilde{r}_c \) and \( \tilde{m}_c \) are of channel type, and \( \tilde{r}_b \) and \( \tilde{m}_b \) are of base type. The rest of the proof follows a similar reasoning as in Case 5.a and the result holds with \( C'[\_] = \nu \tilde{m}'.(\nu \tilde{m}.D'|\_). \)

Case 6.b, u is of channel type and \( u \notin \tilde{m} \cup \tilde{n} \): Notice that in such a case \( A_0 \equiv \nu \tilde{m}.(\nu \tilde{m}.(D_1 | D_2) | \emptyset.D_1 | D_2 | P_1^{u/x} | P_2) \). The rest of the proof follows a similar reasoning as in Case 4.b and the result will hold with \( C'[\_] = \nu \tilde{m}.(\nu \tilde{m}.D'|\_).D' = D_1 | D_2 \) and \( A' = \nu \tilde{r}_c.(P_1^{u/x} | P_2) \).

Case 6.c, u is of channel type and \( u \notin \tilde{n} \): Notice that in such a case \( A_0 \equiv \nu \tilde{m}.(\nu \tilde{m}.(D_1 | D_2) | \emptyset.D_1 | D_2 | P_1^{u/x} | P_2) \). The rest of the proof follows a similar reasoning as in Case 4.c and the result will hold with \( C'[\_] = \nu \tilde{m}'.(\nu \tilde{m}'.D'|\_).D' = D_1 | D_2 \) and \( A' = \nu \tilde{r}'.(P_1^{u/x} | P_2) \).

Case 6.d, u is of channel type and \( u \notin \tilde{n} \): Note that since \( u \notin \tilde{n} \), \( \nu \tilde{m}.D \equiv \nu \tilde{u}.\nu \tilde{m}.D \) for some \( \tilde{m}' \) such that \( u \notin \tilde{m}' \). Hence, \( A_0 \equiv \nu \tilde{m}.\nu \tilde{u}.(\nu \tilde{m}'.(D_1 | D_2) | \emptyset.D_1 | D_2 | P_1^{u/x} | P_2) \). The rest of the proof follows a similar reasoning as in Case 4.d and the result will hold with \( C'[\_] = \nu \tilde{m}'.(\nu \tilde{m}'.D'|\_).D' = D_1 | D_2 \) and \( A' = \nu \tilde{r}'.(P_1 | P_2) \).
This conclude the proof of the first property. The second property is in fact easy to prove: All rules in the eavesdropping semantics other than THEN and ELSE will be mapped by the rule COMM in the classical semantics. One can notice that since we know that A and C do not contain eavesdropped processes and since the transformation from A to $\overline{A}$ and $C[\ldots]$ to $\overline{C}_S[\ldots]$ only adds processes of the form $\text{eav}(c,y).0$, the communication rules all becomes instances of the rule COMM. For instance, an application of rule C-EAV would result into the following

$$\text{out}_{ho}(c,u).P | \text{in}_{ho}(c,x).Q | \text{eav}(c,y).0 \xrightarrow{\tau_e} P | Q\{u/x\}$$

which is typically the rule COMM when we remove the transformation and so the process $\text{eav}(c,y).0$. Lastly, since any instance of $\omega d$ has no impact on the classical semantics, every rules thus corresponds to the rule COMM once the transformation removed. □

**Corollary 3.** Let A be an closed honest extended process. Let $C[\ldots] = \nu_\pi.(\nu_\pi.D | \ldots)$ be an attacker evaluation context $c$-closing for A such that D is named-cleaned and eavesdrop-free. Let $S$ be a set of channel names such that $fc(C[A]) \subseteq S$. For all channel c, $C[A] \downarrow^c_S$ iff $\overline{C}_S[A] \downarrow^c_S$.

**Theorem 6.** $\approx^e_m \subseteq \approx^p_m \cap \approx^c_m$.

**Proof.** Consider two closed honest extended process A and B. We assume $A \approx^e_m B$. We first show that $A \approx^e_m B$.

Let $C[\ldots]$ be an attacker evaluation context $c$-closing for A and B. Notice that in the classical semantics, a process $\text{eav}(c,x).P$ as the same behaviour as the process 0. Hence, there exists $C^1[\ldots]$ an attacker evaluation context eavesdrop-free and $c$-closing for A and B such that for all c, $C[A] \downarrow^c_S \Rightarrow C^1[\ldots] \downarrow^c_S$ and $C[B] \downarrow^c_S \Rightarrow C^1[\ldots] \downarrow^c_S$ (1). Moreover, relying on the structural equivalence, we deduce that there exists $C^2 = \nu_\pi.(\nu_\pi.D | \nu_\pi.(\ldots | E))$ attacker evaluation context eavesdrop-free and $c$-closing for A and B such that D is named-cleaned, $C^1[A] \equiv C^2[A]$ and $C^1[B] \equiv C^2[B]$. Lastly, by renaming $\pi$ through the structural equivalence, we deduce that there exist $A', B'$ two closed honest extended process and $C^3[\ldots] = \nu_\pi.(\nu_\pi'.(D' | \ldots))$ attacker evaluation context eavesdrop-free and $c$-closing for A and B such that D is named-cleaned, $C^2[A] \equiv C^3[A']$ and $C^2[B] \equiv C^3[B']$. Therefore, we have $C[A] \equiv C^3[A']$ and $C[B] \equiv C^3[B']$. Lastly, let us denote $S = fc(C^3[A']) \cup fc(C^3[B'])$, relying on Lemma 11 and Definition 13, one can note that there exists $C^4$ attacker evaluation context $e$-closing for A and B such that $\overline{C^3}_S[A'] \equiv C^4[A]$ and $\overline{C^3}_S[B'] \equiv C^4[B]$. We can conclude the proof as follows: Let $S = fc(C[A]) \cup fc(C[B])$. For all channel c,
Let us now prove that $A \approx^e_p B$. Let $C[\_\_]$ be an attacker evaluation context $p$-closing for $A$ and $B$. As for the classical semantics, notice that in the private semantics, a process $eav(c, x).P$ as the same behaviour as the process $0$. Hence, there exists $C^1[\_\_]$ an attacker evaluation context eavesdrop-free and $p$-closing for $A$ and $B$ such that for all $c$, $C[A] \downarrow^e_p \leftrightarrow C^1[A] \downarrow^e_p$ and $C[B] \downarrow^e_c \leftrightarrow C^1[B] \downarrow^e_c$. Moreover, notice that $\rightarrow_p \subseteq \rightarrow_c$. Hence, for all $c$, $C^1[A] \downarrow^e_c$ implies $C^1[A] \downarrow^e_c$ and $C^1[B] \downarrow^e_e$ implies $C^1[B] \downarrow^e_e$. Furthermore, since $C^1[\_\_]$ is eavesdrop-free and $A, B$ are both honest, we deduce that rules C-EAV and C-OEAV can never be applied in a derivation of $C^1[A]$ or $C^1[B]$. Hence, we obtain that for all $c$, $C^1[A] \downarrow^e_p \leftrightarrow C^1[A] \downarrow^e_p$ and $C^1[B] \downarrow^e_p \leftrightarrow C^1[B] \downarrow^e_p$. Lastly, $A \approx^e_m B$ implies that for all channel $c$, $C^1[A] \downarrow^e_p \leftrightarrow C^1[B] \downarrow^e_p$. We can conclude the proof by combining all these statements as follows: for all channel $c$,

$$C[A] \downarrow^e_p \leftrightarrow C^1[A] \downarrow^e_p \leftrightarrow C^1[A] \downarrow^e_c \leftrightarrow C^1[B] \downarrow^e_c \leftrightarrow C^1[B] \downarrow^e_p \leftrightarrow C[B] \downarrow^e_p$$

We have concluded the proof of $\approx^e_m \subseteq \approx^e_p \cap \approx^e_c$. Therefore, it remains to show that this inclusion is not strict. In Figure 7, we have provided two processes $A$ and $B$ such that $A \approx^e_m B, A \approx^e_p B$ but $A \not\approx^e_c B$. Notice that these processes do not contain replication and so are imagine-finite. Thus, by Theorem 1, $A \not\approx^e_c B$ implies $A \not\approx^e_m B$. Moreover, by Proposition 3, $A \approx^e_m B$ and $A \approx^e_p B$ implies $A \approx^e_c B, A \approx^e_p B$. Once again by Theorem 1, we deduce that $A \approx^e_m B, A \approx^e_p B$. Hence, we conclude that $\approx^e_m \subseteq \approx^e_p \cap \approx^e_c$. □

Appendix E. Proof of Theorem 2

**Theorem 2.** For all ground, closed honest extended processes $A$, for all channels $d$, we have that $A \parallel^s_d \iff A \parallel^d_d \iff A \parallel^s_d$.

**Proof.** We will prove that the following three implications: (1) $A \parallel^s_d \Rightarrow A \parallel^d_d$, (2) $A \parallel^d_d \Rightarrow A \parallel^s_d$ and (3) $A \parallel^s_d \Rightarrow A \parallel^d_d$.

Given a trace $\tau$, let us denote $S(\tau) = \{c \mid \text{tr}_1\text{out}(c, t)\text{tr}_2 = \tau \text{ and } \text{tr}_1 \text{ does not bind } c\}$.

**Implication 1.** $A \parallel^s_d \Rightarrow A \parallel^d_d$: Since $A$ is honest, the only rules that differs are the rules COMM and C-PRIV. Furthermore, since $A$ is honest we also know that $c \notin oc(A)$.

We show that for all $A \parallel^s_d A'$, there exist $\nu n. A'' \equiv A', \nu n. A''$ and a frame $\phi$ such that $S(\tau) \subseteq S(\tau')$ and $\nu n. (A'' \mid \phi)$ such that $\nu n. (A'' \mid \phi)$ such that. We prove this result by induction on the length of the derivation $A \parallel_{\ell_1, \ldots, \ell_m} \tau c A'$ with $\tau$ being $\ell_1 \ldots \ell_m$ without the $\tau$ actions.

**Base case $m = 0$:** Hence $\tau = \varepsilon$ and so the result directly holds with $\phi = 0$.

**Inductive step $m > 0$:** In such a case, by our inductive hypothesis, there exists $\nu r. B \equiv A_{m-1}$ and a frame $\phi$ such that $S(\tau') \subseteq S(\tau')$ and $A \parallel^s_d \nu r. (B \mid \phi)$. W.l.o.g. we can assume that bound names and variables in $\nu r. (B \mid \phi)$ are bound once and distinct from free names and variables. We can also assume that $B$ is name-cleaned. We do case analysis on the rule applied in $A_{m-1} \parallel^s_d A_m$.

- Case 1, any rule but the rule COMM: In such a case, by definition of the semantics, the result directly holds.
Thus by our property, we obtain that there exist $A$, $\tau_r$, and distinct from free names and variables. Moreover we can assume that $\tau_r = \nu^r.(P_1{u/z} | P_2 | P_3 | \phi)$, Hence the result holds.

We conclude by noticing that if $A \models^p_\sigma$ then there exist $A_c$, $tr_c$ such that $A \equiv^e_c A_c$ and $d \in S(tr_c)$.

In such a case, by our inductive hypothesis, there exists $\tau_r$, $A'$ such that $A \equiv^e_p A'$ and $d \in S(tr)$. Since $\ell_p \subset \ell_e$, $A \equiv^e_p A'$ and so $A \models^d_\sigma$.

Implication 3, $A \models^d_\theta \Rightarrow A \models^e_\sigma$: Since $A$ is honest, the only rules that differ are the rules $\text{COMM}$, $\text{C-PRIV}$, $\text{EAV-OCH}$, $\text{EAV-CH}$, $\text{EAV-T}$.

We show that for all $A \equiv^e_p A'$, there exist $tr'$ such that $A \equiv^e_p A'$ and $S(tr) \subseteq S(tr')$. We prove this result by induction on the length of the derivation $A \equiv^{\ell_1, \ldots, \ell_m}_e A'$ with $tr$ being $\ell_1\ldots \ell_m$ without the $\tau$ actions.

Base case $m = 0$: Hence $tr = \varepsilon$ and so the result directly holds with $tr' = \varepsilon$.

Inductive step $m > 0$: In such a case, by our inductive hypothesis, there exists $tr''$ such that $A \equiv^e_p A_{m-1}$. W.l.o.g. we can assume that bound names and variables in $A_{m-1}$ are bound once and distinct from free names and variables. Moreover we can assume that $A_{m-1} \equiv \nu \tilde{n}.B$ with $B$ name-cleaned. We do a case analysis on the rule applied in $A_{m-1} \equiv^{\ell_1, \ldots, \ell_m}_e A_m$.

- Case 1, rule $\text{C-PRIV}$: In such a case, $B = \text{out}^{ho}(c,u).P \mid \text{in}^{ho}(c,x).Q \mid R$, $c \in \tilde{n}$ and $A_m \equiv \nu \tilde{n}.(P \mid Q{u/x} \mid R)$. Notice that $B \equiv^{\ell_1}_e \nu \tilde{n}.(P \mid Q{u/x} \mid R)$ by rule $\text{COMM}$ hence the result holds with $tr' = tr''$.

- Case 2, rule $\text{EAV-OCH}$: In such a case, $B = \text{out}^{ho}(c,u).P \mid \text{in}^{ho}(c,x).Q \mid R$, $\ell = \nu u.eav(c,u)$, $u$ is of channel type, $u \in \tilde{n}$ and $A_m \equiv \nu \tilde{n}'.(P \mid Q{u/x} \mid R)$ with $\tilde{n} = \tilde{n}'.u$. By applying rule $\text{OPEN-CH}$ followed by rule IN, we obtain that $A_{m-1} \equiv^{\nu u.out(c,u).in(c,u)}_e A_m$. Hence the result holds with $tr' = tr''.\nu u.out(c,u).in(c,u)$.

- Case 3, rule $\text{EAV-CH}$: In such a case, $B = \text{out}^{ho}(c,u).P \mid \text{in}^{ho}(c,x).Q \mid R$, $\ell = eav(c,u)$, $u$ is of channel type, $u \not\in \tilde{n}$ and $A_m \equiv \nu \tilde{n}.(P \mid Q{u/x} \mid R)$. By applying rule $\text{OUT-CH}$ followed by rule IN, we obtain that $A_{m-1} \equiv^{\text{out}(c,u).in(c,u)}_e A_m$. Hence the result holds with $tr' = tr''.\text{out}(c,u).in(c,u)$.
Lemma 13. When restricted to $\mathcal{D}(p)$, we have $\approx^p_e = \approx^p_c \subseteq \approx^p_r$ for $r \in \{\ell, t\}$.

Proof. By Lemma 3, we only need to consider the case $r = \ell$.

Before proving the main result, we show the following preliminary result: For all honest processes $A, B$, if $B \in \mathcal{D}(p)$, $A \approx^p_B B$, $A \equiv \nu \eta.(\text{out}^{ho}(c, u).P \ | \ \text{in}^{ho}(c, x).Q \ | \ R, \ \epsilon = \nu z.eav(c, z))$, $u$ is of base type and $A_m \equiv \nu \eta.(P \ | \ Q^{\{u/z\}} \ | \ R)$. By applying rule Out-T followed by rule In, we obtain that $A_{m-1} \overset{\nu z.\text{out}(c, z).\text{in}(c, z)}{\Rightarrow} A_m$. Hence the result holds with $tr' = tr''\nu z.\text{out}(c, z).\text{in}(c, z)$.

• Case 5, any other rule : In such a case, by definition of the semantics, the result directly holds.

We conclude by noticing that if $A \parallel^r s$ then there exist $A', tr_e$ such that $A \overset{\nu z_e}{\Rightarrow} A'$ and $d \in S(tr_e)$. Thus by our property, we obtain that there exist $tr_e$ such that $A \overset{\nu z_e}{\Rightarrow} A'$ and $S(tr_e) \subseteq S(tr_e)$ and so $d \in S(tr_e)$ which implies $A \parallel^r s$. $\square$

Appendix F. Proof of Theorem 7

Let us now focus on the main result. We start by proving $\approx^p_e = \approx^p_c$. By Theorem 5 we have that $\approx^p_r \subseteq \approx^p_e$. It remains to show that $\approx^p_e \subseteq \approx^p_r$.

Let $A, B \in \mathcal{D}(p)$ such that $A \approx^p_e B$. We show that $\approx^p_e$ is also a labelled bisimulation in the eavdrop semantics.

• Since $A \approx^p_e B$, we have that $\phi(A) \sim \phi(B)$.

• If $A \overset{\ell}{\Rightarrow} e A'$ then, as $B$ is a honest process, no C-EAV or C-OEAV transition is possible. Thus $A \overset{\ell}{\Rightarrow} p A'$. As $A \approx^p_e B$, there exists $B \overset{\ell}{\Rightarrow} p B'$ such that $A' \approx^p_e B'$. Since $\overset{\ell}{\Rightarrow} p \subseteq \overset{\ell}{\Rightarrow} e$, we have $B \overset{\ell}{\Rightarrow} e B'$.

• If $A \overset{\ell}{\Rightarrow} e A'$ with $\ell = \nu x.eav(c, x)$ or $\ell = \text{in}(c, M)$ then $A \overset{\ell}{\Rightarrow} e A'$. As $A \approx^p_e B$, there exists $B \overset{\ell}{\Rightarrow} e B'$ such that $A' \approx^p_e B'$. Once again since $\overset{\ell}{\Rightarrow} e \subseteq \overset{\ell}{\Rightarrow} p$, we have $B \overset{\ell}{\Rightarrow} p B'$. 

• Case 4, rule EAV-T: In such a case, $B = \text{out}^{ho}(c, u).P \ | \ \text{in}^{ho}(c, x).Q \ | \ R, \ \ell = \nu z.eav(c, z)$, $u$ is of base type and $A_m \equiv \nu \eta.(P \ | \ Q^{\{u/z\}} \ | \ R)$. By applying rule Out-T followed by rule In, we obtain that $A_{m-1} \overset{\nu z.\text{out}(c, z).\text{in}(c, z)}{\Rightarrow} A_m$. Hence the result holds with $tr' = tr''\nu z.\text{out}(c, z).\text{in}(c, z)$.
• if $A \nu z.eav(c,z)$ then $A \equiv \nu n.(\text{out}^\text{ho}(c,u).P \mid \text{in}^\text{ho}(c,x).Q \mid R)$ and $A' = \nu n.(P \mid Q\{v/z\} \mid R \mid \{v/z\})$. Note that $A \nu z.out(c,z) \nu z.p \circ A'$. As $A \approx_p B$, there exists $B \nu z.out(c,z).in(c,z) \approx_p B'$ and $A' \approx_\ell B'$. Thanks to our preliminary result, we know that there exists $B \approx_{\nu z} B_2$ with $B_2 = \nu n.(\text{out}^\text{ho}(c,v),P' \mid \text{in}^\text{ho}(c,x).Q' \mid R')$ and $\nu n.\{(v/z) \cup \phi(R')\} \sim \nu n.\{(v/z) \cup \phi(R')\}$ where $z$ is fresh. Thus, $B \approx_{\nu z} B_2 \nu z.eav(c,z) B_3$, $\phi(B_3) \sim \phi(A')$ and $B \nu z.out(c,z).in(c,z) \approx_{\nu z} B_3$ for some $B_3$. Recall that $A' \approx_p B'$ meaning that $\phi(A') \sim \phi(B')$ and so $\phi(B') \sim \phi(B_3)$. As $B \in D(p), \phi(B') \sim \phi(B_3)$, $B \nu z.eav(c,z) B_3$ and $B \nu z.out(c,z).in(c,z) \approx_{\nu z} B'$ imply $B_3 \approx_\ell B'$. As $A' \approx_p B', A' \approx_\ell B_3$. Hence, we showed that $B \nu z.eav(c,z) B_3$ and $A' \approx_\ell B_3$ which allows us to conclude the proof of $\approx_\ell \subseteq \approx_\ell$.

Let us now prove that $\approx_\ell \subseteq \approx_\ell$. We showed above that $\approx_\ell = \approx_\ell$. Moreover, we have that $\approx_\ell \subseteq \approx_\ell$ by Theorem 5, which allows us to conclude.

We conclude our proof by showing that the inclusion $\approx_\ell \subseteq \approx_\ell$ is strict. Consider the processes $P$ and $Q$ displayed in Figure 9b. First notice that $P \not\approx_\ell Q$ since $Q \approx_\ell B$ for some $Q'$ but the process $P$ cannot output on $d$ directly. We can also easily show that $P, Q \in D(p)$ and $P \approx_\ell Q$. Indeed, first the frame of any transition consists only of multiple outputs of the public name $a$. Therefore the static equivalence always holds. Second, any action on $P$, i.e. output on $c$ or $d$ and input on $c$ can always be matched on $Q$ by unfolding a replication. Similarly, any output or input on $c$ from $Q$ can be matched by $P$ by unfolding a replication. Finally, any output on $d$ from $Q$ can be matched by unfolding the replication and applying an internal communication on $c$. Therefore $P \approx_\ell Q$. The proof of $Q \in D(p)$ is similar. To prove that $P \in D(p)$, one must notice that in $P \approx_\ell P'$, the number of output transitions on $d$ available on $P'$ is uniquely defined by $\text{tr}$. Thus, when considering $P \approx_\ell P_1$ and $P \approx_\ell P_2$, an output on $d$ is possible on $P_1$ if and only if an output on $d$ is possible on $P_2$. Hence $P_1 \approx_\ell P_2$ and so $P \in D(p)$. □

**Lemma 4.** $D(p) = D(e), D(c) \not\subseteq D(p)$ and $\not\subseteq D(c)$.

**Proof.** We start by showing that $D(p) \not\subseteq D(c)$. Consider the process $A$ displayed in Figure 9a. $A \in D(c)$ since $A \approx_\ell \circ \text{out}^\text{ho}(c,a)$ by the rule COMM and $\text{out}^\text{ho}(c,a) \not\approx_\ell A$. Moreover, $A \in D(p)$ since for all $\text{tr}$, there is a unique $A'$ such that $A \approx_\ell A'$. Hence $D(p) \not\subseteq D(c)$.

We now show that $D(c) \not\subseteq D(p)$. Consider the process $B$ displayed in Figure 9b. Intuitively, the use of the private channel $s$ in $B$ encodes a non deterministic choice between the two processes $P$ and $Q$. We already showed in the proof of Lemma 13 that $P \approx_\ell Q$ and $P \approx_\ell Q$. With a similar proof, we can also show that $P, Q \in D(c)$ which allows us to deduce that $B \in D(c)$. However, $B \approx_\ell P, B \approx_\ell Q$ and $P \not\approx_\ell Q$ imply $B \not\in D(p)$.

Let us show $D(e) \subseteq D(p)$. Consider an honest closed process $A$ such that $A \in D(e)$. Let $A \approx_\ell A_1$ and $A \approx_\ell A_2$. By definition of the semantics, $A \approx_\ell A_1$ implies $A \approx_{\nu z} A_i$, for $i = 1, 2$. Since $A \in D(e)$, we deduce $A_1 \approx_\ell A_2$. By applying Theorem 5, we obtain $A_1 \approx_\ell A_2$ which concludes the proof of $D(e) \subseteq D(p)$.

Finally, let us show that $D(p) \subseteq D(e)$. Let $A \in D(p), A \approx_\ell A_1$ and $A \approx_\ell A_2$. We can define $\text{tr}'$ from $\text{tr}$ by replacing all instances of $\nu z.eav(c,z)$ by $\nu z.out(c,z).in(c,z)$ to obtain $A \approx_\ell A_1$ and
A \overset{r_1}{\Rightarrow}_p A_2. As A \in \mathcal{D}(p), A_1 \approx^p_p A_2. Moreover, by definition of \mathcal{D}(p), A \in \mathcal{D}(p) also implies that A_1, A_2 \in \mathcal{D}(p). By Lemma 13, A_1 \approx^p_p A_2 implies A_1 \approx^p_p A_2 which allows us to conclude. □

The following theorem now follows directly from Lemmas 4 and 13

**Theorem 7.** When restricted to \( \mathcal{D}(p) \), we have \( \approx^p_{r_1} = \approx^p_{r_2} \subseteq \approx^p_\ell \) for \( s_1, s_2 \in \{p, e\} \), \( r_1, r_2, r_3 \in \{\ell, t\} \).

**Appendix G. Proof of Theorem 8**

**Theorem 8.** When restricted to \( \mathcal{BD}(p) \), we have that \( \approx^p_p = \approx^p_\ell \subseteq \approx^p_\ell \) for \( r \in \{\ell, t\} \).

**Proof.** Thanks to Lemma 13, we already know that \( \approx^p_\ell = \approx^p_\ell \) and \( \approx^p_p \subseteq \approx^p_\ell \). We will show in Lemma 15 the stronger result that the implication is strict for the class \( \mathcal{AD} \) of action determinate processes which is a subset of \( \mathcal{BD}(p) \) (Lemma 14).

**Appendix H. Proof of Lemma 14**

**Lemma 14.** \( \mathcal{AD} \subseteq \mathcal{BD}(p) \).

**Proof.** Let \( P \in \mathcal{AD} \). We will show that \( P \in \mathcal{D}(p) \) which allows us to conclude as processes in \( \mathcal{AD} \) do not use replication. Let us define the following transition rule \( \frac{\star}{\star} \) such that \( P \overset{\star}{\Rightarrow} P' \) iff \( P \overset{r_1}{\Rightarrow}_p P' \) with only application of the rules THEN or ELSE, and these two rules cannot be applied on \( P' \). Since \( bn(P) \cap Ch = \emptyset \), we deduce that the rule C-PRIV cannot be applied (all channels are public). Hence the only possible \( \tau \) transitions are the applications of the rules THEN and ELSE.

Notice that \( P \overset{\star}{\Rightarrow} P' \) implies \( P \approx^p_p P' \). Moreover for all \( P \overset{\star}{\Rightarrow} P' \), we can always swap the internal transitions in the trace so that they are always applied before visible actions when possible.

Hence we can always obtain a trace of the form \( P \overset{\star}{\Rightarrow} P_1 \overset{\ell_1}{\Rightarrow}_p P_1' \overset{\star}{\Rightarrow} P_2 \overset{\ell_2}{\Rightarrow}_p \ldots \overset{\star}{\Rightarrow} P_n \overset{\ell_n}{\Rightarrow}_p P_n' \) such that \( P_n' \approx^p_p P' \) and \( tr = \ell_1 \ldots \ell_n \).

To prove that \( P \in \mathcal{D}(p) \), we show the following property: for all \( P \overset{\star}{\Rightarrow} P_1 \overset{\ell_1}{\Rightarrow}_p P_1' \overset{\star}{\Rightarrow} P_2 \overset{\ell_2}{\Rightarrow}_p \ldots \overset{\star}{\Rightarrow} P_n \overset{\ell_n}{\Rightarrow}_p P_n' \), for all \( P \overset{\star}{\Rightarrow} Q_1 \overset{\ell_1}{\Rightarrow}_p Q_1' \overset{\star}{\Rightarrow} Q_2 \overset{\ell_2}{\Rightarrow}_p \ldots \overset{\star}{\Rightarrow} Q_n \overset{\ell_n}{\Rightarrow}_p Q_n' \), we have \( Q_i = P_i \) and \( Q_i' \equiv P_i' \) for all \( i \in \{1, \ldots, n\} \). This property can be proved by induction on \( n \). The base case \( n = 0 \) being trivial, we focus on the inductive case \( n > 0 \). In such a case, by applying our inductive hypothesis, we deduce that \( Q_{n-1}' \equiv P_{n-1}' \). Note that \( Q_{n-1} \overset{\star}{\Rightarrow} Q_n, P_{n-1}' \overset{\star}{\Rightarrow} P_n \) and \( Q_{n-1}' \equiv P_{n-1}' \) implies that \( P_{n-1}' \overset{\star}{\Rightarrow} Q_n \). As mentioned in the previous paragraph, it entails \( P_n \equiv Q_n \). Thus, it remains to show that \( P'_{n-1} \overset{\star}{\Rightarrow} Q_{n-1} \). If \( \ell_n = in(c, t) \) then in such a case, \( P'_{n-1} \overset{\nu k}(in(c, x), R \mid U) \) for some \( R \) and \( U \). By definition of an action determinate process, we know that \( U \) does not have an input on \( c \) at top-level, i.e. \( U \not\overset{\star}{\Rightarrow} in(c, y) \). Hence, there is only one possible transition \( \ell_n \) on \( P_n \) meaning that \( P'_{n-1} \equiv Q_{n-1}' \). A similar reasoning holds for the case \( \ell_n = \nu z.out(c, z) \).

To see that the inclusion is strict, simply observe that for the process \( P \overset{\star}{\Rightarrow} \text{out}^{\nu z}(c, a) \mid \text{out}^{\nu z}(c, a) \), we have \( P \in \mathcal{D}(p) \), but \( P \not\in \mathcal{AD} \). □
Appendix I. Proof of Theorem 9

Lemma 15. There exist $P, Q \in \mathcal{AD}$ such that $P \approx_{\mathcal{E}} Q$ but $P \not\approx_{\mathcal{P}} Q$.

Proof. Consider the processes $P$ and $Q$ displayed in Figure 10. Notice that an input transition with channel $d$ in the private semantics is possible on $P$ but not on $Q$. Therefore, we directly deduce that $P \not\approx_{\mathcal{P}} Q$. Let us now prove that $P \approx_{\mathcal{E}} Q$. We do an analysis on the transitions on $P$ and $Q$.

- $P \xrightarrow{\tau} P'$: Note that from $P$, the only possible $\tau$ action is an internal communication on the channel $c$. More specifically,

$$P' = \nu k_1, \ldots, k_7. (R_1(k_1) \mid \text{in}^{ho}(d, x_2) \text{if } x_2 = k_2 \text{ then out}^{ho}(c, k_3))$$

We can also apply an internal communication on $Q$ to obtain the same process, i.e., $Q \xrightarrow{\tau} Q'$. Note that we trivially have $P' \approx_{\mathcal{E}} P'$.

- $Q \xrightarrow{\tau} Q'$: Once again the only possible $\tau$ action is an internal communication on the channel $c$. In fact, we have $Q' = P'$ where $P'$ was defined in the previous case with $P \xrightarrow{\tau} P'$.

- $P \xrightarrow{\nu x.\text{out(c,x)}} P'$: In such a case, $P' = \nu k_1, \ldots, k_5. (\text{in}^{ho}(c, x_1). R_1(x_1) \mid \text{in}^{ho}(d, x_2) \text{if } x_2 = k_2 \text{ then out}^{ho}(c, k_3) \mid \{k_1 / z\})$. Moreover, notice that $Q \xrightarrow{\nu z.\text{out(c,z)}} P'$ too. Hence the result holds.

- $P \xrightarrow{\text{in(c,t)}} P'$: Since $k_1, \ldots, k_5$ are all bound and there is no frame in $P$, we know that $t \neq k_i$ for all $i = 1 \ldots 5$. Thus, $P' = \nu k_1, \ldots, k_5. (R_1(t) \mid \text{out}^{ho}(c, k_1) \mid \text{in}^{ho}(d, x_2) \text{if } x_2 = k_2 \text{ then out}^{ho}(c, k_3))$. We can make two observations on the process $P'$: The first one being that the only possible transition on $R_1(t)$ is the execution of the conditional leading to the nil process. The second one being that in $k_2$ does not appear anymore in $P'$ other than in the test $x_2 = k_2$. Therefore, $k_2$ cannot be deducible from $P'$ and so we deduce that $P' \approx_{\mathcal{E}} P'$.

For sake of readability, we denote by $\tau(c)$ a $\tau$ action corresponding to an internal communication on a channel $c$. Moreover, we denote by $\tfrac{\tau(c)}{\tau(d)}$ a $\tau$ action corresponding to a conditional (i.e. rules THEN or ELSE) and we denote by $\tfrac{\tau(c)}{\tau(d)}$ the transition $\text{if } \tau(c) \text{ then } \tau(d)$ $\text{else } \tau(c)$. Finally, we denote by $k = k_1, \ldots, k_5$. Let us execute $Q$ as follows:

$$Q \xrightarrow{\tau(c)} \nu \tilde{k}.(R_1(k_1) \mid \text{in}^{ho}(d, x_2) \text{if } x_2 = k_2 \text{ then out}^{ho}(c, k_3)) \xrightarrow{\tau(d)} \nu \tilde{k}.(\text{in}^{ho}(c, x_3) \text{if } x_3 = k_3 \text{ then } R_3 \text{ else } \text{in}^{ho}(d, x) \mid \text{out}^{ho}(c, k_3)) \xrightarrow{\text{in(c,t)}} \nu \tilde{k}.(\text{in}^{ho}(d, x) \mid \text{out}^{ho}(c, k_3))$$

Let us denote $Q' = \tilde{k}.(\text{in}^{ho}(d, x) \mid \text{out}^{ho}(c, k_3))$. We trivially have that $Q' \approx_{\mathcal{E}} \nu k_1. (\text{out}^{ho}(c, k_1) \mid \text{in}^{ho}(d, x_2))$ and so $P' \approx_{\mathcal{E}} Q'$.

- $Q \xrightarrow{\text{in(c,t)}} Q'$: Once again, we know that $t \neq k_i$ for all $i = 1 \ldots 5$. Thus $Q' = \nu \tilde{k}.(R_1(t) \mid \text{out}^{ho}(c, k_1) \mid \text{in}^{ho}(d, x_2) \text{if } x_2 = k_2 \text{ then out}^{ho}(c, k_3))$. With $t \neq k_1$ and the fact that $k_2$ is no longer deducible in $Q'$, we obtain that $Q' \approx_{\mathcal{E}} \nu k_1. \text{out}^{ho}(c, k_1) \mid \text{in}^{ho}(d, x_2)$. 

Let us execute $P$ as follows:

$$
P \xrightarrow{\tau(c)}_{c} \nu \tilde{k}.(R_{1}(k_{1}) | \text{in}^{ho}(d, x_{2}).\text{if } x_{2} = k_{2} \text{ then out}^{ho}(c, k_{3}))
$$

$$
\xrightarrow{\tau(d)}_{c} \nu \tilde{k}.(\text{in}^{ho}(c, x_{3}).\text{if } x_{3} = k_{3} \text{ then } R_{3} \text{ else in}^{ho}(d, x) | \text{out}^{ho}(c, k_{3}))
$$

$$
\xrightarrow{\tau(c)}_{c} \nu \tilde{k}.R_{3}
$$

$$
\xrightarrow{\text{in}(c,t)}_{c} \nu \tilde{k}.(\text{out}^{ho}(c, k_{1}).\text{in}^{ho}(d, x_{3}).R_{5}(x_{3}))
$$

Let us denote $P' = \nu \tilde{k}.(\text{out}^{ho}(c, k_{4}).\text{in}^{ho}(d, x_{5}).R_{5}(x_{5}))$. Note that $k_{5}$ is not deducible in $P'$ meaning that $P' \approx_{\tilde{e}}^{\xi} \nu k_{4}.\text{out}^{ho}(c, k_{4}).\text{in}^{ho}(d, x_{5})$. Since we already showed that $Q' \approx_{\tilde{e}}^{\xi} \nu k_{1}.\text{out}^{ho}(c, k_{1}).\text{in}^{ho}(d, x_{2})$, we conclude that $P' \approx_{\tilde{e}}^{\xi} Q'$.

- $P' \xrightarrow{\text{in}(d,t)}_{c} P'$: In such a case, we have $P' = \nu \tilde{k}.(\text{in}^{ho}(c, x_{1}).R_{3}(x_{1}) | \text{out}^{ho}(c, k_{1}) | \text{if } t = k_{2} \text{ then out}^{ho}(c, k_{3}))$. Note that $t \neq k_{2}$ and that so $k_{3}$ is not deducible in $P'$. Thus, we obtain the following:

$$
P' \approx_{\tilde{e}}^{\xi} \nu \tilde{k}.(\text{in}^{ho}(c, x_{1}).\text{if } x_{1} = k_{1} \text{ then out}^{ho}(d, k_{2}).\text{in}^{ho}(c, x_{3}).\text{in}^{ho}(d, x) | \text{out}^{ho}(c, k_{1}))
$$

Let us execute $Q$ as follows:

$$
Q \xrightarrow{\tau(c)}_{c} \nu \tilde{k}.(R_{1}(k_{1}) | \text{in}^{ho}(d, x_{2}).\text{if } x_{2} = k_{2} \text{ then out}^{ho}(c, k_{3}))
$$

$$
\xrightarrow{\tau(d)}_{c} \nu \tilde{k}.(\text{in}^{ho}(c, x_{3}).\text{if } x_{3} = k_{3} \text{ then } R_{3} \text{ else in}^{ho}(d, x) | \text{out}^{ho}(c, k_{3}))
$$

$$
\xrightarrow{\tau(c)}_{c} \nu \tilde{k}.R_{3}
$$

$$
\xrightarrow{\tau(d)}_{c} \nu \tilde{k}.(\text{in}^{ho}(d, x_{5}).R_{5}(x_{5}) | \text{out}^{ho}(d, k_{5}))
$$

$$
\xrightarrow{\text{if in}(c,t)}_{c} \nu \tilde{k}.(\text{in}^{ho}(c, x_{6}).\text{if } x_{6} = k_{6} \text{ then out}^{ho}(d, k_{7}).\text{in}^{ho}(c, x_{3}).\text{in}^{ho}(d, x) | \text{out}^{ho}(c, k_{5}))
$$

As we already proved $P' \approx_{\tilde{e}}^{\xi} \nu \tilde{k}.(\text{in}^{ho}(c, x_{1}).\text{if } x_{1} = k_{1} \text{ then out}^{ho}(d, k_{2}).\text{in}^{ho}(c, x_{3}).\text{in}^{ho}(d, x) | \text{out}^{ho}(c, k_{1}))$, we conclude that $P' \approx_{\tilde{e}}^{\xi} Q'$.

We conclude that $P \approx_{\tilde{e}}^{\xi} Q$.

It remains to prove that $P, Q \in \mathcal{D}(p)$. We show in fact that $P$ and $Q$ are both action-determinate and we conclude by applying Lemma 14.

We do a case analysis on the trace of $P$.

Case 1: $P \xrightarrow{\text{in}(c,t)}_{c} P_{1}$. In such a case,

$$
P_{1} = \nu \tilde{k}.(R_{1}(t) | \text{out}^{ho}(c, k_{1}) | \text{in}^{ho}(d, x_{2}).\text{if } x_{2} = k_{2} \text{ then out}^{ho}(c, k_{3}))
$$

Since $t \neq k_{1}$, $R_{1}(t)$ can only be reduced into the nil process. Moreover, $k_{2}$ is not deducible in $P_{1}$, meaning that $\text{out}^{ho}(c, k_{3})$ can never be executed. Thus, $P_{1}$ is action-determinate.
Case 2: \( P \xrightarrow{in(d,t)} c P_1 \). In such a case,
\[
    P_1 = νk.(\text{in}^h_o(c,x_1).R_1(x_1) \mid \text{out}^h_o(c,k_1) \mid \text{if } t = k_2 \text{ then } \text{out}^h_o(c,k_3))
\]
Since \( t \neq k_2 \), \( \text{out}^h_o(c,k_3) \) can never be executed meaning that \( k_3 \) is not deducible in \( P_1 \). As such, we deduce that proving that \( P_1 \) is action-determinate is equivalent to proving that the following process is action-determinate:
\[
    νk.(\text{in}^h_o(c,x_1) \mid \text{if } x_1 = k_1 \text{ then } \text{out}^h_o(d,k_2) \cdot \text{in}^h_o(c,x_3) \cdot \text{in}^h_o(d,x) \mid \text{out}^h_o(c,k_1))
\]
Since there is no common actions between the two parallel processes in the previous process, we conclude that it is action-determinate.

Case 3: \( P \xrightarrow{νz.out(z)} c P_1 \). In such a case,
\[
    P_1 = νk.(\text{in}^h_o(c,x_1).R_1(x_1) \mid \text{in}^h_o(d,x_2) \mid \text{if } x_2 = k_2 \text{ then } \text{out}^h_o(c,k_3) \mid \{k_1/z\})
\]
We have to consider two possible actions next, that are either an input on \( c \) or an input on \( d \). If we consider \( P_1 \xrightarrow{in(d,t)} c P_2 \) then once again, \( t \neq k_2 \) meaning that \( \text{out}^h_o(c,k_3) \) can never be executed. Thus proving \( P_2 \) is action-determinate is equivalent to proving that the following process is action-determinate:
\[
    νk.(\text{in}^h_o(c,x_1) \mid \text{if } x_1 = k_1 \text{ then } \text{out}^h_o(d,k_2) \cdot \text{in}^h_o(c,x_3) \cdot \text{in}^h_o(d,x) \mid \{k_1/z\})
\]
This process is trivially action-determinate. Thus, let us consider \( P_1 \xrightarrow{in(c,t)} c P_2 \), i.e.
\[
    P_2 = νk.(R_1(t\{k_1/z\}) \mid \text{in}^h_o(d,x_2) \mid \text{if } x_2 = k_2 \text{ then } \text{out}^h_o(c,k_3) \mid \{k_1/z\})
\]
If \( t \neq z \), i.e. \( t\{k_1/z\} \neq k_1 \), then \( R_1(t\{k_1/z\}) \) can only be reduced to the nil process and so \( P_2 \) would be trivially action-determinate. Thus, let us assume that \( t = z \).

Note that the determinacy of \( P_2 \) can only be broken if \( R_3 \) can be executed but after the following transitions
\[
P_2 \xrightarrow{νz.out(d,z_2) \cdot \text{in}(d,z_2) \cdot \text{in}(c,z_1) \cdot \text{in}(c,z_2)} c P_3 = νk.(R_3 \mid \{k_1/z \cdot k_2/z_2 \cdot k_3/z_3\})
\]
Therefore, we only need to show that \( R_3 \) is action-determinate. Once again, this may not be the case only if \( R_5(k_3) \) is executed. However, this is only possible with transitions as follows:
\[
P_3 \xrightarrow{νz.out(c,z_4) \cdot \text{in}(c,z_4) \cdot \text{in}(d,z_3) \cdot \text{in}(d,z_5)} c νk.R_5(k_3).
\]
Since \( R_5(k_3) \) is trivially action-determinate, we conclude that \( P \) is an action-determinate process.

The proof of \( Q \) being action-determinate is similar. □

Theorem 9. When restricted to \( AD \), we have that \( \|r\|_P = \|r\|_r \subseteq \|r\|_c \) for \( r \in \{\ell,t\} \).

Proof. Thanks to Lemmas 13 and 14, we already know that \( \|r\|_r = \|r\|_c \) and \( \|r\|_c \subseteq \|r\|_c \). The fact that the implication is strict follows from Lemma 15. □
Appendix J. Proof of Theorem 10

Lemma 16. Let \( P, Q \in \text{SAD} \) such that \( P \cong^r Q \). If there exists \( c \) such that \( P \xrightarrow{\nu \text{.out}(c,x)}_P P_1 \) then there exists \( d \) such that:

- \( P \xrightarrow{\nu \text{.out}(d,x)}_P P' \)
- \( Q \xrightarrow{\nu \text{.out}(d,x)}_P Q' \)
- \( P' \cong^r Q' \)

Proof. Consider the maximal trace \( P \xrightarrow{\nu \text{.out}(d,x)}_C P_2 \xrightarrow{tr}_c P_3 \) of \( P \) that starts with an output. Note that in fact \( P \xrightarrow{\nu \text{.out}(d,x)}_C P_2 \xrightarrow{tr}_c P_3 \) contains an internal communication transition, the we can transform this transition into two transitions \( P \xrightarrow{\nu \text{.out}(c,z)}_C P_2 \xrightarrow{\nu \text{.out}(c,z)}_C \) which would contradict the maximality hypothesis on \( P \xrightarrow{\nu \text{.out}(d,x)}_C P_2 \xrightarrow{tr}_c P_3 \). Since \( P \cong^r Q \), we know that \( Q \xrightarrow{tr}_c Q_2 \xrightarrow{tr}_c Q_3 \) with \( P_2 \cong^r Q_2 \) and \( P_3 \cong^r Q_3 \). If \( Q \xrightarrow{tr}_c Q_2 \xrightarrow{tr}_c Q_3 \) then the result holds. Otherwise, there is an internal communication transition on some channel \( c \) in \( Q \xrightarrow{\nu \text{.out}(d,x)}_C Q_2 \). Once again, we can replace this transition into two transitions \( Q \xrightarrow{\nu \text{.out}(c,z)}_C in(c,z)_c \) with \( tr' = \nu \text{.out}(c,z).\text{in}(c,z) \), yielding \( Q \xrightarrow{tr'}_c Q_2 \xrightarrow{tr}_c Q_3 \) or \( Q \xrightarrow{\nu \text{.out}(d,x)}_C tr'.tr \) start with an output. As \( P \cong^r Q \), \( tr'.\nu \text{.out}(d,x).tr \) or \( \nu \text{.out}(d,x).tr'.tr \) is also a trace of \( P \) which contradicts the maximality of \( \nu \text{.out}(d,x).tr \). □

Lemma 17. Let \( P, Q \in \text{SAD} \) such that \( P \cong^r Q \). If

- for all \( d, P', P \xrightarrow{\nu \text{.out}(d,x)}_P P' \)
- for all \( d, Q', Q \xrightarrow{\nu \text{.out}(d,x)}_P Q' \)

then for all \( c, M \), for all \( P \xrightarrow{\text{in}(c,M)}_P P' \), there exists \( Q \xrightarrow{\text{in}(c,M)}_P Q' \) such that \( P' \cong^r Q' \).

Proof. Since \( P \xrightarrow{\text{in}(c,M)}_P P' \) and \( P \cong^r Q \) then there exists \( Q \xrightarrow{\text{in}(c,M)}_P Q' \). Thanks to our hypothesis on \( Q \) we know that \( Q \xrightarrow{\text{in}(c,M)}_P Q'' \xrightarrow{\text{in}(c,M)}_P Q' \) (No internal communication are possible directly on \( Q \) since not output is available). But \( P \cong^r Q \) and \( Q \xrightarrow{\text{in}(c,M)}_P Q'' \) implies that \( P \xrightarrow{\text{in}(c,M)}_P P'' \) with \( P'' \cong^r Q'' \). Once again by our hypothesis on \( P \), we have that \( P \xrightarrow{\text{in}(c,M)}_P P'' \). As \( P \in \text{SAD}, P \in D(p) \), As \( P \xrightarrow{\text{in}(c,M)}_P P'' \) and \( P \xrightarrow{\text{in}(c,M)}_P P' \), we deduce \( P'' \cong^r P' \) which implies \( P'' \cong^r P' \) from Theorem 13. To summarize, we have:

- \( P'' \cong^r P' \)
- \( P'' \cong^r Q' \)
- \( Q'' \cong^r Q' \)
- \( P'' \xrightarrow{\text{in}(c,M)}_P P'' \)
- \( P'' \cong^r P' \)
If in fact $Q'' \xrightarrow{\nu^* P} Q'$ then the result directly holds. Else there exists an internal communication transition in $Q'' \xrightarrow{\nu.z.\text{out}(d,z).\text{in}(d,z)} P Q'$ for some $\phi$.

Take $Q' \xrightarrow{\nu} Q_1$ a maximal trace of $Q'$, we deduce that $\nu.z.\text{out}(d,z).\text{in}(d,z).\text{tr}$ is a trace of $Q''$. Since $P'' \approx_c^i Q''$, we know that $\nu.z.\text{out}(d,z).\text{in}(d,z).\text{tr}$ is a trace of $P''$. With $P'' \xrightarrow{\nu^* P''} P''$, we deduce that $\nu.z.\text{out}(d,z).\text{in}(d,z).\text{tr}$ is a trace of $P''$. As $P'' \approx_c^i Q'$, $\nu.z.\text{out}(d,z).\text{in}(d,z).\text{tr}$ is also a trace of $Q'$ which gives a contradiction with $\text{tr}$ being a maximal trace of $Q'$.

**Lemma 18.** Let $P, Q \in SAD$ such that $P \approx_c^i Q$. If there exists $c$ such that

- $P \xrightarrow{\nu.z.\text{out}(c,x)} P'$,
- $Q \xrightarrow{\nu.z.\text{out}(c,x)} Q'$, and
- $P' \approx_c^i Q'$

then for all $d$, $P \xrightarrow{\nu.y.\text{out}(d,y)} P_1$ implies $Q \xrightarrow{\nu.y.\text{out}(d,y)} Q_1$ and $P_1 \approx_c^i Q_1$.

**Proof.** Consider $P \xrightarrow{\nu.y.\text{out}(d,y)} P_1$. If $c = d$ then $P$ being strongly action determinate implies that $P_1 = P'$ and so the result trivially holds. Therefore, let us consider that $c \neq d$.

In such a case, we deduce $P = \nu.\text{out}(c,u).R_1 \parallel \text{out}(d,v).R_2 \parallel R_3$ and $Q = \nu'.\text{out}(c,u').S_1 \parallel S$. Since $P \xrightarrow{\nu.z.\text{out}(c,x)} P'$ and $Q \xrightarrow{\nu.z.\text{out}(c,x)} Q'$, we deduce that $P' = \nu.\text{out}(c,u').S_1 \parallel \text{out}(d,v).R_2 \parallel R_3 \parallel \{u'/x\}$ and $Q' = \nu'.\text{out}(c,u').S_1 \parallel S \parallel \{u'/x\}$. Note that $P' \approx_c^i Q'$ implies that one of the following two cases:

1. $S = \text{out}(d,v').S_2 \parallel S_3$
2. $S_1 = \text{out}(c,u').S_2 \parallel S_3$

As $P' \approx_c^i Q'$, we know that $P \xrightarrow{\nu.z.\text{out}(c,x)} \nu.\text{out}(c,x) P_2$ and $Q \xrightarrow{\nu.z.\text{out}(c,x)} \nu'.\text{out}(d,y) Q_2$ where

- $P_2 = R_1 \parallel R_2 \parallel R_3 \parallel \{u/x, v/y\}$
- $Q_2 = S_1 \parallel S_2 \parallel S_3 \parallel \{u'/x, v'/y\}$
- $\nu.\text{out}(c,x) P_2 \approx_c^i \nu'.\text{out}(d,y) Q_2$

Moreover, by $P \approx_c^i Q$, we deduce that $Q \xrightarrow{\nu.z.\text{out}(c,x)} \nu.\text{out}(c,x) Q_4$ with $\nu.\text{out}(c,x) P_2 \approx_c^i \nu'.\text{out}(d,y) Q_3$ and $P_1 \approx_c^i Q_4$.

In both cases 1 and 2, since $Q$ is strongly action determinate, the outputs $\text{out}(c,u')$ and $\text{out}(d,v')$ in $Q$ are necessarily executed either within an internal communication or with the output rule in the transition $Q \xrightarrow{\nu.z.\text{out}(c,x)} \nu.\text{out}(c,x) Q_4$ and $\text{out}(d,v')$. In both cases, we can make apparent the internal communications in $Q \xrightarrow{\nu.z.\text{out}(c,x)} \nu.\text{out}(c,x) Q_4$ and output first $\text{out}(c,u')$ and $\text{out}(c,u')$ giving us the following trace:

$Q \xrightarrow{\nu.z.\text{out}(c,x)} \nu.\text{out}(c,x) Q_2 \xrightarrow{\nu'} \nu'.\text{out}(d,y) Q_3$. In both cases, we can make apparent the internal communications in $Q \xrightarrow{\nu.z.\text{out}(c,x)} \nu.\text{out}(c,x) Q_4$ and output first $\text{out}(c,u')$ and $\text{out}(c,u')$ giving us the following trace:

$Q \xrightarrow{\nu.z.\text{out}(c,x)} \nu.\text{out}(c,x) Q_2 \xrightarrow{\nu'} \nu'.\text{out}(d,y) Q_3$. In both cases, we can make apparent the internal communications in $Q \xrightarrow{\nu.z.\text{out}(c,x)} \nu.\text{out}(c,x) Q_4$ and output first $\text{out}(c,u')$ and $\text{out}(c,u')$ giving us the following trace:
for some \( x', y' \) such that \( vy'.out(d, y) \) and \( vx'.out(c, x) \) are included in \( vy'.out(d, y').vx'.out(c, x').tr \) and \( Q_2 = S_1 | S_2 | S_3 | \{ u'/x', v'/y' \} \)

Since we assumed that \( Q \overset{vy.out(d, y), c}_\equiv Q_4 \) contains some internal communications, we deduce that \( tr \neq \varepsilon \). To summarized, we showed that:

- \( \nu \pi. P_2 \overset{\varepsilon}{\sim}_D \nu \pi'. Q_2 \)
- \( \nu \pi. P_2 \overset{\varepsilon}{\sim}_D \nu \pi'. Q_3 \)
- \( \nu \pi'. Q_2 \overset{\varepsilon}{\sim}_D \nu \pi'(Q_3 | \Phi) \)
- \( tr \neq \varepsilon \)
- \( Q_2' \) is the process \( Q_2 \) where \( x \) and \( y \) in the domain of the frame have been replaced by \( x' \) and \( y' \) respectively.

Consider now a maximal trace of \( \nu \pi'. Q_3 \), i.e. \( \nu \pi'. Q_3 \overset{tr'}{\Rightarrow}_D Q_5 \). Hence, \( tr' \) is a trace of \( \nu \pi'. (Q_3 | \Phi) \).

Hence, \( tr.tr' \) is a trace of \( \nu \pi'. Q_3 \). Thus, \( tr.tr' \{ x'/x'; y'/y' \} \) is a trace of \( \nu \pi'. Q_3 \). As \( \nu \pi. P_2 \overset{\varepsilon}{\sim}_D \nu \pi'. Q_2 \)
implies \( \nu \pi. P_2 \overset{\varepsilon}{\sim}_D \nu \pi'. Q_2 \) (Lemma 13), \( tr.tr' \{ x'/x'; y'/y' \} \) is a trace of \( \nu \pi. P_2 \) and so a trace of \( \nu \pi'. Q_3 \)
thanks to \( \nu \pi. P_2 \overset{\varepsilon}{\sim}_D \nu \pi'. Q_3 \). Since we assumed that \( tr \neq \emptyset \), we obtain a contradiction with \( tr' \)
being a maximal trace of \( \nu \pi'. Q_3 \).

**Corollary 4.** Let \( P, Q \in S.AD \) such that \( P \overset{\varepsilon}{\approx}_c Q \). Assume that for all \( P', Q' \), \( |P'| + |Q'| < |P| + |Q| \)
and \( P' \overset{\varepsilon}{\approx}_c Q' \) implies \( P \overset{\varepsilon}{\approx}_c Q' \).

For all \( d, P \overset{vy.out(d, y)}{\Rightarrow}_D P_1 \) implies \( Q \overset{vy.out(d, y)}{\Rightarrow}_D P_1 \) and \( P_1 \overset{\varepsilon}{\approx}_c Q_1 \).

**Proof.** Let \( P \overset{vy.out(d, y)}{\Rightarrow}_D P_0 \). By Lemma 16, we know that there exists \( c \) such that \( P \overset{\varepsilon}{\approx}_c P', Q \overset{\varepsilon}{\approx}_c P' \) and \( P' \approx_c Q' \). Since \( |P'| + |Q'| < |P| + |Q| \), we deduce that \( P' \approx_c Q' \). Note that \( P, Q \in D(c) \). Hence, there exists \( P_1, P_2, Q_1, Q_2 \) such that \( P \overset{\varepsilon}{\Rightarrow}_D P_1 \overset{vy.out(c, x)}{\Rightarrow}_D P_2 \overset{\varepsilon}{\Rightarrow}_D P' \) and \( Q \overset{\varepsilon}{\Rightarrow}_D Q_1 \overset{vy.out(c, x)}{\Rightarrow}_D Q_2 \overset{\varepsilon}{\Rightarrow}_D Q' \) such that \( P \approx_c P_1, P_2 \approx_c P', Q \approx_c Q_1 \) and \( Q_2 \approx_c Q' \). Note that by Lemma 13, \( P \approx_c P_1 \) and \( Q \approx_c Q_1 \) implies \( P \approx_c P_1 \) and \( Q \approx_c Q_1 \). Hence \( P_1 \approx_c Q_1 \) and \( P_2 \approx_c Q_2 \). We conclude by applying Lemma 18.

**Lemma 19.** Let \( P, Q \in S.AD \) such that \( P \overset{x}{\Rightarrow}_D P_1 \) and \( Q \overset{x}{\Rightarrow}_D P \) and \( P \overset{\varepsilon}{\approx}_c Q \). Assume that for all \( P', Q' \), \( |P'| + |Q'| < |P| + |Q| \) and \( P' \overset{\varepsilon}{\approx}_c Q' \) implies \( P \overset{\varepsilon}{\approx}_c Q' \). We have \( skel(P) = skel(Q) \).

**Proof.** Consider first that for all \( d, P \overset{vy.out(d, x)}{\Rightarrow}_D Q \) and \( Q \overset{vy.out(d, x)}{\Rightarrow}_D Q \). By applying Lemma 17, we deduce that \( skel(P) = skel(Q) \). By applying Corollary 4, we also deduce that \( \{ out(d) \in skel(P) \mid d \in Ch \} = \{ out(d) \in skel(Q) \mid d \in Ch \} \). Therefore, it only remains to prove that \( \{ in(d) \in skel(P) \mid d \in Ch \} = \{ in(d) \in skel(Q) \mid d \in Ch \} \).

Consider \( P \overset{in(d, M)}{\Rightarrow}_c P' \) such that \( in(d) \notin skel(Q) \). Since \( P \overset{\varepsilon}{\approx}_c Q \), then \( Q \overset{in(d, M)}{\Rightarrow}_c Q' \) for some \( Q' \). As \( in(d) \notin skel(Q) \) and \( Q \overset{vy.out(d, x)}{\Rightarrow}_D P' \), we deduce that \( Q \overset{\varepsilon}{\Rightarrow}_c Q' \overset{vy.out(d, x)}{\Rightarrow}_c P' \) where \( Q \overset{\varepsilon}{\Rightarrow}_c Q'' \)
is an internal communication on some public channel \( c \). Since \( \{ out(d) \in skel(P) \mid d \in Ch \} = \{ out(d) \in skel(Q) \mid d \in Ch \} \), we deduce that:

- \( P = \nu \tilde{k}.(out^{ho}(c, u).R_1 | in^{ho}(d, x).R_2 | R_3) \).
\[ Q = \nu k'.(\text{out}(c, v).S_1 | \text{in}(c, x).S_2) \]

Moreover, Corollary 4 also tells us that \( P_1 = \nu \tilde{k}.(R_1 | \text{in}(d, x).R_2 | R_3 | \{u/z\}) \approx P_1 \nu \tilde{k}'.(S_1 | \text{in}(c, x).S_2 | S_3 | \{v/z\}) = Q_1. \]

Since \( P, Q \in \mathcal{D}(p) \), we can assume w.l.o.g. that \( P_1 \not\approx_{p, Q_1} \not\approx_{p, Q_2}. \]

As \( P_1 \not\approx_{p, Q_1} \), \( \text{skel}(P_1) = \text{skel}(Q_1) \). Hence \( \text{in}(d) \in \text{skel}(S_1) \) or \( \text{in}(d) \in \text{skel}(S_3) \). As we assumed \( \text{in}(d) \notin \text{skel}(Q) \) or \( \text{in}(d) \in \text{skel}(S_1) \). Therefore, \( Q = \nu k'.(\text{out}(c, v).(\text{in}(d, y).S_1 | S_2) | \text{in}(c, x).S_2) \).

Note that \( Q \not\rightarrow_{p, c} Q_2 = \nu \tilde{k}'.(\text{out}(c, v).(\text{in}(d, y).S_1' | S_2' | S_3 | \{N/y\}) | S_3). \) As \( Q \not\approx_{p, Q_2} \), there exists \( P \not\approx_{p, Q_2} P_2 \) and \( Q_2 \not\approx_{p, Q_2} P_2 \) for some \( P_2 \). Since \( |Q_2| + |P_2| < |P| + |Q| \), \( Q_2 \not\approx_{p, Q_2} P_2 \) implies \( Q_2 \not\approx_{p, Q_2} P_2 \). Note that in \( Q_2 \), \( \text{out}(c, v) \) and \( \text{in}(d, y) \) are sequential. As \( Q_2 \not\approx_{p, Q_2} P_2 \), \( P_2 \) should also contain a similar sequence of \( \text{out}(c, u') \) and \( \text{in}(d, y') \) for some \( u' \) and \( y' \). Since \( P \not\approx_{p, Q_2} P_2 \), we deduce that this sequence should appear either in \( \text{out}(c, u).R_1 \) or in \( \text{in}(d, x).R_2 \) or in \( R_3 \). However, by definition of strong action determinate, there cannot be any instance of \( \text{in}(d, y) \) in \( R_1 \) or \( R_2 \) because of \( \text{in}(d, x).R_2 \). Similarly, there cannot be any instance of \( \text{out}(c, u') \) for some \( u' \) in \( \text{in}(d, x).R_2 \) or \( R_3 \) because of \( \text{out}(c, u).R_1 \). Thus we obtain a contradiction and so \( \text{in}(d) \notin \text{skel}(Q) \). \[ \square \]

**Lemma 20.** Let \( P, Q \in \mathcal{SAD} \) such that \( P \not\rightarrow_{p, Q} \) and \( Q \not\rightarrow_{p, Q}. \) If \( P \not\approx_{p, Q} \) and \( \text{skel}(P) = \text{skel}(Q) \) then \( P \not\approx_{p, Q}. \)

**Proof.** Thanks to Lemma 17, we know that the result trivially holds if \( \text{skel}(P) \) contains only inputs. Thus, let us consider \( P = \nu k.(\text{out}(c, u).R_1 | R_2) \). Note that by Corollary 4 and since \( P \in \mathcal{D}(p) \), we know that \( Q = \nu \tilde{k}'.(\text{out}(c, v).S_1 | S_2) \) with \( \nu \tilde{k}.(R_1 | R_2 | \{u/z\}) \not\approx_{p} \nu k'.(S_1 | S_2 | \{v/z\}). \)

Therefore, let us define the relation \( \mathcal{R} \) such that \( P' \mathcal{R} Q' \) iff either \( P' \not\approx_{p} Q' \) or the following properties hold:

- \( \text{skel}(P') = \text{skel}(Q') \),
- \( P' = \nu k.(\text{out}(c, u).R_1 | R_2) \), \( Q = \nu \tilde{k}'.(\text{out}(c, v).S_1 | S_2) \) for some \( R_2 \), \( S_2 \), and
- \( \nu k.(R_1 | R_2 | \{u/z\}) \not\approx_{p} \nu k'.(S_1 | S_2 | \{v/z\}). \)

Note that \( P \mathcal{R} Q \). We show that \( \mathcal{R} \subseteq \not\approx_{p} \). Consider \( P' \mathcal{R} Q' \). By definition of \( \mathcal{R} \), if \( P' \not\approx_{p} Q' \) then the result trivially holds. Therefore, we consider the other part of the definition of \( \mathcal{R} \) and we prove the \( P', Q' \) satisfies the definition of \( \not\approx_{p} \)

- We know that \( \nu \tilde{k}.(R_1 | R_2 | \{u/z\}) \not\approx_{p} \nu k'.(S_1 | S_2 | \{v/z\}). \) Hence we deduce that \( \phi(P') \sim \phi(Q') \).
- Assume \( P' \not\rightarrow_{p, c} Q' \). Since \( P' \) is strongly action determinate, we deduce that \( P' = \nu k'.(R_1 | R_2 | \{u/z\}) \). We also have \( Q' \not\approx_{p, c} Q'' = \nu \tilde{k}'.(S_1 | S_2 | \{v/z\}) \) and we also have \( P'' \not\approx_{p, Q''} Q'' \). Hence \( P'' \mathcal{R} Q''. \)
- Assume \( P' \not\rightarrow_{p, c} Q'' \). In such a case, \( P'' = \nu k.(\text{out}(c, u).R_1 | R_2) \) for some \( R_2 \). Note that \( P'' \not\approx_{p, c} Q'' \). Since \( \nu k.(R_1 | R_2 | \{u/z\}) \not\approx_{p} \nu k'.(S_1 | S_2 | \{v/z\}) \) and \( \text{skel}(P') = \text{skel}(Q') \), we deduce that \( Q' \not\approx_{p, Q''} Q'' \). However, note that we also have
When restricted to I/O-unambiguous processes, we have that

\[ \text{Theorem 11.} \]

**Proof.** From Theorems 5, 6 and 4, we already know that \( P \approx_r Q \). Hence, we conclude that \( P \approx_r Q \).

Appendix K. Proof of Theorem 11

**Theorem 11.** When restricted to I/O-unambiguous processes, we have that \( \approx_r \subseteq \approx_r \) for \( r \in \{ l, t \} \).

**Proof.** We first focus on the proof of \( \approx_r \subseteq \approx_r \) for \( r \in \{ l, m, t \} \) and \( s_1, s_2 \in \{ c, p, e \} \). We first focus on the proof of \( \approx_r \subseteq \approx_r \) for \( r \in \{ l, o, m, t \} \) and \( s_1, s_2 \in \{ c, p, e \} \).