Information Flow Security in Dynamic Contexts*

Riccardo Focardi Sabina Rossi

Dipartimento di Informatica,
Università Ca' Foscari di Venezia
via Torino 155, 30172 Venezia, Italy
{focardi,srossi}@dsi.unive.it

Abstract

We study information flow security in the setting of mobile agents. We propose a sufficient condition to security named *Persistent_BNDC*. A process is *Persistent_BNDC* when every of its reachable states satisfies a basic Non-Interference property called *BNDC*. By imposing that security persists during process execution, one is guaranteed that every potential migration is performed in a stable, secure state. We define a suitable bisimulation-based equivalence relation among processes, that allows us to express the new property as a single equivalence check, thus avoiding the universal quantifications over all the reachable states (required by *Persistent_BNDC*) and over all the possible hostile environments (implicit in the basic Non-Interference property *BNDC*). We prove that *Persistent_BNDC* is a sufficient condition to the security of mobile agents by (i) giving a sound and complete characterization of *Persistent_BNDC* in terms of *dynamic contexts*, i.e., execution contexts that can non-deterministically change at run-time, abstractly modelling arbitrary migrations; (ii) showing that *Persistent_BNDC* implies information flow security when agent mobility is explicitly expressed in the calculus.

1 Introduction

The protection of relevant data from undesired accesses is a typical security issue concerning both systems and networks. It can be seen as a problem of controlling how information flows among different entities. For example, protecting the confidentiality of information corresponds to guaranteeing that it never flows to unauthorized users; protecting the integrity, instead, can be seen as a problem of avoiding information flow from unauthorized users to the (container of the) data. Unfortunately, even when direct access to data is forbidden by access control policies or by cryptography, it might be the case that subtle, indirect, information flows are still possible. These unwanted flows can be based, e.g., on some observable system side-effects (giving rise to the so called *covert channels* [55, 37]), or on some weakness in cryptographic algorithms and protocols.

*This work is a revised and extended version of [20], and it has been partially supported by the MIUR project "Abstract Interpretation: Design and Applications" (AIDA).

Motivated by this need of controlling information flow as a whole (both direct and indirect), Goguen and Meseguer introduced the notion of *Non-Interference* [22]. Non-Interference formalizes the absence of information flow within deterministic systems. Given a system in which *confidential* (i.e., high level) and *public* (i.e., low level) information may coexist, *Non-Interference* requires that confidential inputs never affect the outputs on the public interface of the system, i.e., never interfere with the low level users. If such a property holds, one can conclude that no information flow is ever possible from high to low level.

Starting from Sutherland [54], many definitions extending the concept of Non-Interference to non-deterministic systems have been proposed in the literature. They are developed in different settings such as programming languages [4, 49, 50, 51], trace models [21, 27, 31, 32, 33, 34, 35, 36, 56], process calculi [10, 17, 25, 30, 46, 47, 48], probabilistic models [2, 13], timed models [19, 23], cryptographic protocols [1, 5, 18].

The specific formalization of Non-Interference we consider is *Bisimulation-based Non-Deducibility on Compositions* (*BNDC*, for short), which has been studied in [15, 17]. Intuitively, *BNDC* requires that the low level view of a system E is not affected by any (possibly malicious) high level process Π. In a process algebraic style, the definition has the following form:

$$\forall \Pi \in \mathcal{E}_H, \quad E \setminus H \approx (E|\Pi) \setminus H.$$

The formula above can be read as "the low level view of E is behaviourally equivalent to E composed with any possible high level process Π". Indeed, as we will see, restriction $\setminus H$ has the effect of forbidding any high level action and $\mathcal{E}_H$ represents the set of processes that only use high level actions, also called high level processes.

We start by the consideration that *BNDC* is not *persistent*, i.e., is not preserved during system execution. It might happen that a *BNDC*-secure system, after some execution steps, reaches a state which is not *BNDC*-secure.

Persistence is an important feature when processes may move in the middle of their computation. In fact, it is important to ensure that migration does not happen inside a critical section. In particular, since non-persistent properties are not preserved during computation, it might be the case that a secure (e.g. *BNDC*) process decides to migrate after reaching an insecure (i.e., non-*BNDC*) state. From the point of view of the new host, the incoming process is insecure and, consequently, it should not be executed. In other words, *BNDC* does not guarantee that processes always migrate in stable, secure, states.

To overcome this problem, we make *BNDC* persistent by requiring that every state which is reachable by the secure process still satisfies *BNDC*, i.e.,

$$\forall E' \text{ reachable from } E, \quad E' \text{ is } \mathit{BNDC}.$$

The property we obtain, called *Persistent_BNDC* (*P_BNDC*, for short), is of course persistent, but is also quite difficult to check in its naive form above. As a matter of fact, it contains a double universal quantification: the first over all the reachable states and the second over all the possibly malicious high level processes (required by the nested *BNDC* check).

The first interesting result that we prove is that *P_BNDC* may be equivalently defined as a simple Non-Interference check with a different underlying equivalence notion between processes, i.e., by adopting a different discriminating power on processes. This simpler characterization of *P_BNDC* has the following form:

$$E \setminus H \approx_{\setminus H} E$$

where $\approx_{\setminus H}$ is a behavioural equivalence between processes that may ignore high level actions, i.e., it admits to consider high level actions as they were internal invisible transitions.

The second contribution is to show that *P_BNDC* implies Non-Interference for mobile agents. This is done in two steps:

(i) we give a new, sound and complete, characterization of *P_BNDC* in terms of dynamic contexts, i.e., contexts that can arbitrarily change at run-time. Since, from the point of view of agents, a migration causes a change of the surrounding execution environment, we obtain that *P_BNDC* guarantees security even when agents non-deterministically migrate in every possible way during their executions. However, this is proved at a very high level of abstraction, modelling migrations as context changes not controlled by agents;

(ii) we thus consider a more concrete model, inspired from DPI-calculus [26], and obtained by adding explicit process mobility to the language. We formally prove that *P_BNDC* implies the natural extension of Non-Interference in this new concrete model of mobile agents. At the same time, we show that *BNDC*, in its original form, is not enough to guarantee that a migrating process is secure. As a matter of fact, *BNDC* does not guarantee that processes always migrate in stable, secure states, being it a non-persistent property.

These results turn out to be useful also on the verification side, since the automated check of *BNDC*, and of its generalization to mobility, is still an open problem even for finite-state systems. *P_BNDC*, instead, provides a sufficient condition to guarantee the security of mobile agents, which can be checked in polynomial time on the number of states (see, e.g., [44]).

The paper is organized as follows. In Section 2 we define the *Security Process Algebra* (SPA) language and recall the notion of *weak bisimulation* over SPA terms and the security property *BNDC*. In Section 3, we introduce the persistent variant of *BNDC* called *P_BNDC*, and we characterize it through a new definition of *weak bisimulation up to high level actions*. We also prove some compositionality results of *P_BNDC*, and show that *P_BNDC* is equivalent to the *SBSNNI* property proposed in [15, 17]. In Section 4 we show that persistence is suitable to deal with migrating processes by introducing a notion of dynamic high contexts (Section 4.1) and by extending the SPA language with primitives for mobility (Section 4.2). In Section 5, we apply *P_BNDC* to reason on a simple example process with migration primitives. Finally, in Section 6, we briefly discuss how *P_BNDC* can be efficiently verified and we draw some concluding remarks.

2 Basic Notions

In this section we briefly introduce the *Security Process Algebra* (SPA) language that we will use to specify and analyze security properties over concurrent systems. Moreover, we recall the security property for SPA processes named *Bisimulation-based Non-Deducibility on Compositions* (*BNDC*, for short) [17].

2.1 The SPA language

2.1.1 Syntax.

The SPA language [17] is a slight extension of Milner's CCS [38], where the set of visible actions is partitioned into high level actions and low level ones in order to specify multilevel systems. SPA syntax is based on the same elements as CCS that is: a set $\mathcal{L}$ of *visible* actions such that $\mathcal{L} = I \cup O$ where $I = \{a, b, \ldots\}$ is a set of *input* actions and $O = \{\bar{a}, \bar{b}, \ldots\}$ is a set of *output* actions; a complementation function $\bar{\cdot} : \mathcal{L} \rightarrow \mathcal{L}$, such that $\bar{\bar{a}} = a$, for all $a \in \mathcal{L}$; a special action τ which models internal computations, i.e., not visible outside the system. $Act = \mathcal{L} \cup \{\tau\}$ is the set of all *actions*. Function $\bar{\cdot}$ is extended to Act by defining $\bar{\tau} = \tau$. In order to obtain a partition of the visible actions into two levels we consider two sets, H and L, of high and low level actions which are closed with respect to $\bar{\cdot}$, i.e., $\overline{H} = H$ and $\overline{L} = L$; moreover they are disjoint and form a covering of $\mathcal{L}$, i.e., $H \cap L = \emptyset$ and $H \cup L = \mathcal{L}$.

The syntax of SPA *terms* (or *processes*) is defined as follows:

$$E ::= \mathbf{0} \mid a.E \mid E + E \mid E|E \mid E \setminus v \mid E[f] \mid Z$$

where $a \in Act$, $v \subseteq \mathcal{L}$, $f : Act \rightarrow Act$ is such that $f(\bar{\alpha}) = \overline{f(\alpha)}$ and $f(\tau) = \tau$. Moreover Z is a constant that must be associated with a definition [1] $Z \stackrel{\text{def}}{=} E$.

Intuitively, $\mathbf{0}$ is the empty process that does nothing; $a.E$ is a process that can perform an action a and then behaves as E; $E_1 + E_2$ represents the non deterministic choice between the two processes E_1 and E_2; $E_1 | E_2$ is the parallel composition of E_1 and E_2, where the executions of the two processes are interleaved, possibly synchronized on complementary input/output actions, producing an internal action τ; $E \setminus v$ is a process E prevented from performing actions in v; $E[f]$ is the process E whose actions are renamed *via* the relabelling function f.

To deal with security properties, we will also use the *hiding* operator / of CSP, which can be defined as a relabelling as follows: for a given set $v \subseteq \mathcal{L}$, $E/v \stackrel{\text{def}}{=} E[f_v]$ where $f_v(x) = x$ if $x \notin v$ and $f_v(x) = \tau$ if $x \in v$. In practice, E/v turns all actions in v into internal τ's.

2.1.2 Operational Semantics.

Let $\mathcal{E}$ be the set of SPA terms, ranged over by E and F. Let $\mathcal{L}(E)$ denote the *sort* of E, i.e., the set of the (possibly executable) actions occurring syntactically in E. The sets of

[1] Notice that, for automatic checks over SPA terms it is necessary to add a condition of *guardedness* on constants. This avoids infinite constant substitution loops.

Prefix	$\dfrac{-}{a.E \xrightarrow{a} E}$
Sum	$\dfrac{E_1 \xrightarrow{a} E_1'}{E_1 + E_2 \xrightarrow{a} E_1'} \qquad \dfrac{E_2 \xrightarrow{a} E_2'}{E_1 + E_2 \xrightarrow{a} E_2'}$
Parallel	$\dfrac{E_1 \xrightarrow{a} E_1'}{E_1 \mid E_2 \xrightarrow{a} E_1' \mid E_2} \qquad \dfrac{E_2 \xrightarrow{a} E_2'}{E_1 \mid E_2 \xrightarrow{a} E_1 \mid E_2'} \qquad \dfrac{E_1 \xrightarrow{a} E_1' \quad E_2 \xrightarrow{\bar{a}} E_2'}{E_1 \mid E_2 \xrightarrow{\tau} E_1' \mid E_2'}\, a \in \mathcal{L}$
Restriction	$\dfrac{E \xrightarrow{a} E'}{E \setminus v \xrightarrow{a} E' \setminus v}$ if $a \notin v$
Relabelling	$\dfrac{E \xrightarrow{a} E'}{E[f] \xrightarrow{f(a)} E'[f]}$
Constant	$\dfrac{E \xrightarrow{a} E'}{A \xrightarrow{a} E'}$ if $A \stackrel{\text{def}}{=} E$

Figure 1: The operational rules for SPA

high level processes and low level ones are defined as $\mathcal{E}_H \stackrel{\text{def}}{=} \{E \in \mathcal{E} \mid \mathcal{L}(E) \subseteq H \cup \{\tau\}\}$ and $\mathcal{E}_L \stackrel{\text{def}}{=} \{E \in \mathcal{E} \mid \mathcal{L}(E) \subseteq L \cup \{\tau\}\}$, respectively. Note that $\mathcal{E}_H \cup \mathcal{E}_L \subset \mathcal{E}$, i.e., there exist systems that execute both high and low level actions allowing communications between the two levels.

The operational semantics of SPA processes is given in terms of *Labelled Transition Systems*. A *Labelled Transition System* (LTS) is a triple $(S, A, \rightarrow)$ where S is a set of states, A is a set of labels (actions), $\rightarrow \subseteq S \times A \times S$ is a set of labelled transitions. The notation $(S_1, a, S_2) \in \rightarrow$ (or equivalently $S_1 \xrightarrow{a} S_2$) means that the system can move from the state S_1 to the state S_2 through the action a. An LTS is *finite* if it has a finite number of states and transitions. The operational semantics of SPA is the LTS $(\mathcal{E}, Act, \rightarrow)$, where the states are the terms of the algebra and the transition relation $\rightarrow \subseteq \mathcal{E} \times Act \times \mathcal{E}$ is defined by structural induction as the least relation generated by the axioms and inference rules reported in Figure 1. The operational semantics for a process E is the subpart of the SPA LTS reachable from the initial state E. To denote that two processes E_1 and E_2 have two isomorphic LTSs, meaning that they behave exactly in the same way, we write $E_1 \equiv E_2$.

2.1.3 Observational Equivalence.

The concept of *observation equivalence* between two processes is based on the idea that two systems have the same semantics if and only if they cannot be distinguished by an external observer. This is obtained by defining an equivalence relation over states/terms of the SPA LTS, equating two processes when they are indistinguishable. In this way the semantics of a term becomes an equivalence class of terms. In the literature there are various equivalences of this kind. In this paper we consider *weak bisimulation* equivalence, an observation equivalence which allows one to observe the nondeterministic structure of the LTSs and focus only on the observable actions.

The general notion of *bisimulation* [38] consists of a mutual step-by-step simulation, i.e., given two processes E and F, when E executes a certain action moving to E' then F must be able to simulate this single step by executing the same action and moving to a term F' which is again bisimilar to E', and vice-versa. A weak bisimulation is a bisimulation in which internal τ actions may be ignored, i.e., when F simulates an observable action of E, it can also execute some τ actions before or after that action; moreover τ actions may be simulated by a possibly empty sequence of τ's.

We use the following notations. If $t = a_1 \cdots a_n \in Act^*$, then we write $E \xrightarrow{t} E'$ if $E \xrightarrow{a_1} \cdots \xrightarrow{a_n} E'$. We say that E' is *reachable* from E when there exists $t \in Act^*$ such that $E \xrightarrow{t} E'$. If $a \in Act$, then we write $E \stackrel{a}{\Longrightarrow} E'$ for $E(\xrightarrow{\tau})^* \xrightarrow{a} (\xrightarrow{\tau})^* E'$ where $(\xrightarrow{\tau})^*$ denotes a (possibly empty) sequence of τ labelled transitions. We also write $E \stackrel{\hat{a}}{\Longrightarrow} E'$ for $E \stackrel{a}{\Longrightarrow} E'$ if $a \in \mathcal{L}$, and for $E(\xrightarrow{\tau})^* E'$ if $a = \tau$ (note that $\stackrel{\tau}{\Longrightarrow}$ requires at least one τ labelled transition while $\stackrel{\hat{\tau}}{\Longrightarrow}$ corresponds to $(\xrightarrow{\tau})^*$ and means zero or more τ labelled transitions).

The notion of *weak bisimulation* is defined as follows.

Definition 2.1 (**Weak Bisimulation).** A binary relation $\mathcal{S} \subseteq \mathcal{E} \times \mathcal{E}$ over processes is a *weak bisimulation* if $(E, F) \in \mathcal{S}$ implies, for all $a \in Act$,

- whenever $E \xrightarrow{a} E'$, then there exists F' such that $F \stackrel{\hat{a}}{\Longrightarrow} F'$ and $(E', F') \in \mathcal{S}$;
- whenever $F \xrightarrow{a} F'$, then there exists E' such that $E \stackrel{\hat{a}}{\Longrightarrow} E'$ and $(E', F') \in \mathcal{S}$.

Two processes $E, F \in \mathcal{E}$ are *observation equivalent*, denoted by $E \approx F$, if there exists a weak bisimulation $\mathcal{S}$ containing the pair (E, F).

In [38] it is proved that $\approx$ is the largest weak bisimulation and it is an equivalence relation.

2.2 The Security Property BNDC

The security property named *Bisimulation-based Non-Deducibility on Compositions* (*BNDC*, for short) [17] tries to capture every possible information flow from a *classified* (*high*) level of confidentiality to an *untrusted* (*low*) one. A strong requirement of this definition is that no information flow should be possible even in the presence of malicious processes that run at the classified level. The main motivation is to protect

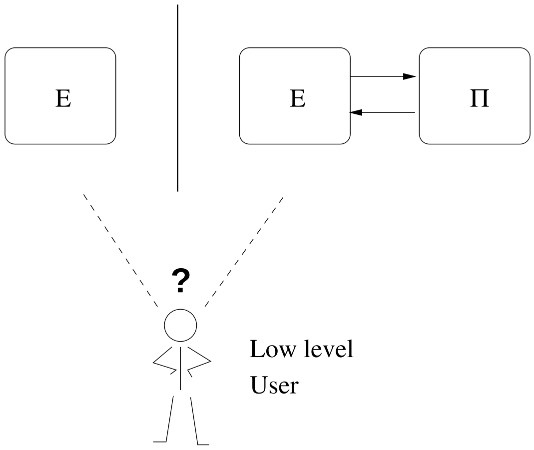

Figure 2: The BNDC property

a system also from internal attacks, which could be performed by the so called *Trojan Horse* programs, i.e., programs that appear honest but hide some malicious code inside them. These programs might be for example downloaded from the network or sent by e-mail, and executed by a high level user at the classified level.

The definition of *BNDC* is based on the basic idea of Non-Interference [22]: "No information flow is possible from high to low if what is done at the high level *cannot interfere* in any way with the low level".

More precisely, the notion of *BNDC* consists of checking the system against all high level potential interactions, representing every possible high level malicious program. A system E is *BNDC* if for every high level process Π a low level user cannot distinguish E from $E|\Pi$. In other words, a system E is *BNDC* if what a low level user sees of the system is not modified by composing any high level process Π with E.

The formal definition of *BNDC* is as follows.

Definition 2.2 (BNDC [17]). Let $E \in \mathcal{E}$.

$$E \in BNDC \text{ if } \forall\, \Pi \in \mathcal{E}_H,\; E/H \approx (E|\Pi) \setminus H.$$

The next proposition provides an equivalent characterization of *BNDC* which does not involve the hiding operator (see Lemma 2 in [17]).

Proposition 2.3 ([17]). *Let* $E \in \mathcal{E}$. $E \in BNDC$ *iff* $\forall\, \Pi \in \mathcal{E}_H,\; E \setminus H \approx (E|\Pi) \setminus H$.

The idea of *BNDC* is depicted if Figure 2. Let us show how *BNDC* works through some simple examples.

Example 2.4. The simplest case of flawed process is $E_1 \stackrel{\text{def}}{=} h.\bar{l}.\mathbf{0}$, where h is high and l is low. The process E_1 accepts the high level input h and, only after such an input is received, it gives $\bar{l}$ as output. We clearly have a direct causality between the high level input h and the low level output $\bar{l}$. As a consequence, a low level user knows that h has been performed by just observing the output $\bar{l}$. This system is not *BNDC*. It is sufficient to consider $\Pi \stackrel{\text{def}}{=} \bar{h}.\mathbf{0}$ and observe that $(E_1|\Pi) \setminus H \approx \bar{l}.\mathbf{0}$ while $E_1 \setminus H \approx \mathbf{0}$. Thus, $E_1 \setminus H \not\approx (E_1|\Pi) \setminus H$.

E_1 can be made secure by letting $\bar{l}$ be also executed independently from h as in $E_1' \stackrel{\text{def}}{=} h.\bar{l}.\mathbf{0} + \bar{l}.\mathbf{0}$. It is easy to prove that E_1' is *BNDC*. □

The next example aims at showing that *BNDC* is also able to detect partial information flows due to the possibility for a high level malicious process to block or unblock a system.

Example 2.5. Consider process $E_2 \stackrel{\text{def}}{=} l_1.h.\bar{l}_2.\mathbf{0} + l_1.\bar{l}_2.\mathbf{0}$, in which the high level input h is performed in between two low level actions l_1 and $\bar{l}_2$. Similarly to the previous example, branch $l_1.\bar{l}_2.\mathbf{0}$ aims at breaking the direct causality between h and $\bar{l}_2$, in fact, $\bar{l}_2$ may be executed even without h has been previously performed. However, consider the same process $\Pi \stackrel{\text{def}}{=} \bar{h}.\mathbf{0}$ as before. We have that $(E_2|\Pi) \setminus H \approx l_1.\bar{l}_2.\mathbf{0}$, but $E_2 \setminus H \approx l_1.\mathbf{0} + l_1.\bar{l}_2.\mathbf{0} \not\approx (E_2|\Pi) \setminus H$, showing that E_2 is not *BNDC*.

Let us discuss why this system is considered insecure: notice that $E_2 \setminus H$ may (nondeterministically) block after the l_1 input while $(E_2|\Pi) \setminus H$ always executes $\bar{l}_2$. The problem is that E_2 nondeterministically chooses between the two possible branches when getting l_1 as input. After this choice is made, it may behave either like $h.\bar{l}_2.\mathbf{0}$, in which the execution of $\bar{l}_2$ depends on h, or like $\bar{l}_2.\mathbf{0}$, in which $\bar{l}_2$ is always performed. A low level user observing a deadlock after l_1 will know that h has *not* been executed, thus having (partial) information about high level activity.

Process E_2 may be "repaired" and made *BNDC*, in different ways:

(i) by including the possibility of choosing to execute $\bar{l}_2$ also in the second branch of the process, thus completely masking high level activity. Process $E_3 \stackrel{\text{def}}{=} l_1.h.\bar{l}_2.\mathbf{0} + l_1.(\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0})$ can be proved to be *BNDC* (see Appendix A);

(ii) by choosing to perform $\bar{l}_2$ or not *after* l_1 has been performed as in process $E_4 \stackrel{\text{def}}{=} l_1.(h.\bar{l}_2.\mathbf{0} + \bar{l}_2.\mathbf{0})$. After l_1 this system becomes the same as E_1' of previous example which can be easily proved to be *BNDC*; based on this observation it is also easy to prove that E_4 is *BNDC*. □

3 Persistent_BNDC

As discussed in the introduction, persistence is a way of strengthening security. In particular, when processes are allowed to move in the middle of their computation it seems reasonable to assume that if each state reachable from the initial process is secure then no malicious attacker will be able to compromise the system whatever are the environments visited by the system.

We first formalize the notion of persistence.

Definition 3.1. A property $\mathcal{P} \subseteq \mathcal{E}$ is persistent if

$$\forall E, E' \in \mathcal{E},\ E \in \mathcal{P} \text{ and } E' \text{ reachable from } E \text{ imply } E' \in \mathcal{P}.$$

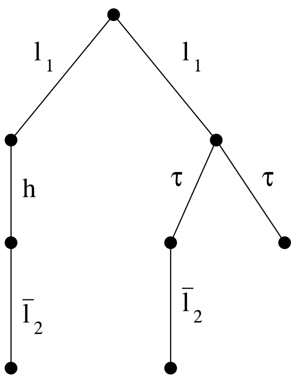

Figure 3: The process E_3

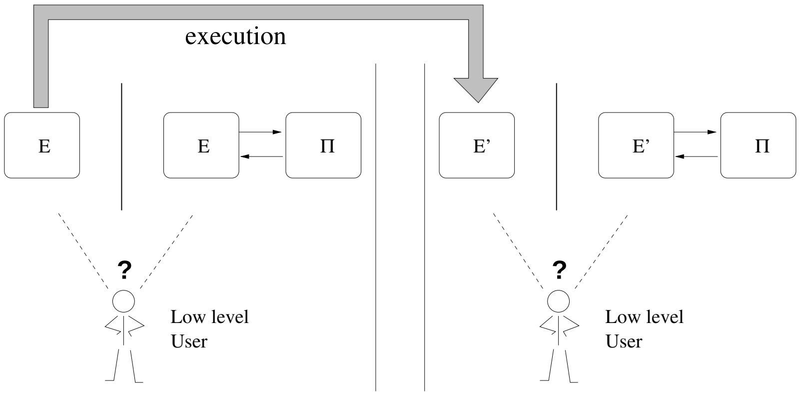

Figure 4: The P_BNDC property

We now study the natural persistent extension of *BNDC*. The security property we obtain is named *Persistent_BNDC* (*P_BNDC*, for short). The idea is that a system E is *P_BNDC* if for every high level process Π and for every state E' reachable from E a low level user cannot distinguish E' from $E'|\Pi$ (see Figure 4). This is equivalent to say that E is *P_BNDC* if for every state E' reachable from E, E' is *BNDC*.

Formally *P_BNDC* is defined as follows.

Definition 3.2 (**Persistent_BNDC**)**.** Let $E \in \mathcal{E}$. $E \in P\_BNDC$ if

$$\forall\, E' \text{ reachable from } E,\ E' \in BNDC.$$

The next example shows that *BNDC* is not persistent. As a consequence, *P_BNDC* is strictly stronger than *BNDC*, i.e., $P\_BNDC \subset BNDC$.

Example 3.3. We consider again the two *BNDC* processes of Example 2.5:

$$E_3 \stackrel{\text{def}}{=} l_1.h.\bar{l}_2.\mathbf{0} + l_1.(\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0})$$

$$E_4 \stackrel{\text{def}}{=} l_1.(h.\bar{l_2}.\mathbf{0} + \bar{l_2}.\mathbf{0})$$

Interestingly, we have that only E_4 is also *P_BNDC*. Intuitively, the reason why E_3 is not *P_BNDC* is related to how causality between high and low is hidden to make the process *BNDC*. In E_3 the second branch aims at simulating all the low level activity a high level user can achieve by playing with action h in the first branch. This simulation is only effective if we consider the process from the very beginning, since, after the non-deterministic choice only one of the branches remains active. In particular, suppose that E_3 reaches the state $h.\bar{l_2}.\mathbf{0}$ after executing the first l_1. It is clear that this state is not secure, as a direct causality between h and $\bar{l_2}$ is present (as in process E_1 of example 2.4). In particular $h.\bar{l_2}.\mathbf{0}$ is not *BNDC* and so E_3 is not *P_BNDC*.

Process E_4 is, instead, *P_BNDC* as formally shown in Example 3.12 of Section 3.2. Intuitively, persistence is achieved by postponing the nondeterministic choice: after executing l_1 the process remains in a secure state, still having a choice to execute l_1 independently of h. $\square$

3.1 P_BNDC and High Contexts

We introduce a novel bisimulation-based equivalence relation, named $\approx_{hc}$, that allows us to give a first characterization of *P_BNDC* with no quantification over all the reachable states. In particular, we show that $E \in P\_BNDC$ if and only if E and $E \setminus H$ are not distinguishable with respect to $\approx_{hc}$. Intuitively, two processes are $\approx_{hc}$-equivalent if they can simulate each other in any possible high context, i.e., in every context of the form $(\_ \mid \Pi) \setminus H$ where $\Pi \in \mathcal{E}_H$.

Let us first formally introduce the notion of high context.

Definition 3.4 (High context). A *high context* $C[\_]$ denotes a term of the form $(\_ \mid \Pi) \setminus H$ where $\Pi \in \mathcal{E}_H$, which can be regarded as a mapping from $\mathcal{E}$ to $\mathcal{E}$ that associates with each process $E \in \mathcal{E}$ the process $C[E] \equiv (E|\Pi) \setminus H$.

Observe that for any high context $C[\_]$ and process E, all the processes reachable from $C[E]$ have the form $C'[E']$ with $C'[\_]$ being a high context too.

We now introduce the concept of *weak bisimulation on high contexts*: the idea is that, given two processes E and F, when a high context $C[\_]$ filled with E executes a certain action moving E to E' then the same context filled with F is able to simulate this step moving F to F' so that E' and F' are again weakly bisimilar on high contexts, and vice-versa. This must be true for every possible high context $C[\_]$. It is important to note that the quantification over all possible high contexts is re-iterated for E' and F'.

We use the following notations. If $t \in Act^*$ and $C[\_]$ is a high context, then we write $E \xrightarrow{t}_C E'$ if there exists a high context $C'[\_]$ such that $C[E] \xrightarrow{t} C'[E']$, and we write $E \stackrel{t}{\Longrightarrow}_C E'$ if there exists a high context $C'[\_]$ such that $C[E] \stackrel{t}{\Longrightarrow} C'[E']$. Therefore, $E \stackrel{\hat{a}}{\Longrightarrow}_C E'$ stands for $C[E] \stackrel{a}{\Longrightarrow} C'[E']$ if $a \in \mathcal{L}$, while it stands for $C[E](\xrightarrow{\tau})^* C'[E']$ if $a = \tau$.

The notion of weak bisimulation on high contexts is defined as follows.

Definition 3.5 (Weak Bisimulation on high contexts). A binary relation $\mathcal{S} \subseteq \mathcal{E} \times \mathcal{E}$ over processes is a *weak bisimulation on high contexts* if $(E,F) \in \mathcal{S}$ implies, for all high contexts $C[\_]$ and for all $a \in Act$,

- whenever $E \xrightarrow{a}_C E'$, then there exists F' such that $F \overset{\hat{a}}{\Longrightarrow}_C F'$ and $(E',F') \in \mathcal{S}$;
- whenever $F \xrightarrow{a}_C F'$, then there exists E' such that $E \overset{\hat{a}}{\Longrightarrow}_C E'$ and $(E',F') \in \mathcal{S}$.

We say that two processes $E,F \in \mathcal{E}$ are *weakly bisimilar on high contexts*, written $E \approx_{hc} F$, if $(E,F) \in \mathcal{S}$ for some weak bisimulation on high contexts $\mathcal{S}$.

From the above definition we may equivalently define $\approx_{hc}$ as follows:

$$\approx_{hc} \quad = \quad \bigcup\{\mathcal{S} : \quad \mathcal{S} \text{ is a weak bisimulation on high contexts}\}.$$

It is easy to prove that

- relation $\approx_{hc}$ is the largest weak bisimulation on high contexts
- relation $\approx_{hc}$ is an equivalence relation.

The next Theorem gives a characterization of *P_BNDC* processes in terms of $\approx_{hc}$. To prove it we use the following lemma.

Lemma 3.6. *Let $E \in \mathcal{E}$ such that $E \approx_{hc} E \setminus H$. Then for all E' reachable from E there exists $E'' \setminus H$ reachable from $E \setminus H$ such that $E' \approx_{hc} E'' \setminus H$.*

Proof. Let $E \approx_{hc} E \setminus H$ and E' be reachable from E. The proof follows by induction on the length l of the path which leads from E to E'.

- *Base*. Let $l = 0$. In this case we can choose E'' equal to E; then $E \equiv E' \equiv E''$ and we know that $E \approx_{hc} E \setminus H$.
- *Inductive step*. Let $l > 0$. Let F be reachable from E with a path of length $l-1$ and $F \xrightarrow{a} E'$. By inductive hypothesis, there exists F' such that $F' \setminus H$ is reachable from $E \setminus H$ and $F \approx_{hc} F' \setminus H$. We distinguish two cases.

  Case 1. Let $a \notin H$. In this case, for any high context $C[\_]$, $F \xrightarrow{a}_C E'$. The fact that $F \approx_{hc} F' \setminus H$ implies that there exists $E'' \setminus H$ such that $F' \setminus H \overset{\hat{a}}{\Longrightarrow}_C E'' \setminus H$ and $E' \approx_{hc} E'' \setminus H$. Since $E'' \setminus H$ is reachable from $E \setminus H$ we have the thesis.

  Case 2. Let $a \in H$. Consider the high context $C[\_] \equiv (\_ \,|\bar{a}.\mathbf{0}) \setminus H$. In this case, $C[F] \equiv (F|\bar{a}.\mathbf{0}) \setminus H) \xrightarrow{\tau} (E'|\mathbf{0}) \setminus H)$, i.e., $F \xrightarrow{\tau}_C E'$. The fact that $F \approx_{hc} F' \setminus H$ implies that there exists $E'' \setminus H$ such that $F' \setminus H \overset{\hat{\tau}}{\Longrightarrow}_C E'' \setminus H$ and $E' \approx_{hc} E'' \setminus H$. Since $E'' \setminus H$ is reachable from $E \setminus H$ we have the thesis. □

We prove that a process E is *P_BNDC* if and only if E and $E \setminus H$ are indistinguishable with respect to the weak bisimulation on high contexts, i.e., whatever high context $C[\_]$ we consider, an observer cannot distinguish between $C[E]$ and $C[E \setminus H]$. Since *P_BNDC* is persistent, this property holds also for every state E' reachable from E. This is consistent with the fact that, in the definition of $\approx_{hc}$, the universal quantification on high level contexts is applied at each derivation step.

Theorem 3.7. *Let $E \in \mathcal{E}$. Then, $E \in P\_BNDC$ iff $E \setminus H \approx_{hc} E$.*

Proof. We first show that $E \setminus H \approx_{hc} E$ implies $E \in P\_BNDC$. In order to do it we prove that

$$\mathcal{S} \quad = \quad \{(E_1 \setminus H, (E_2|\Pi) \setminus H) \mid \quad \Pi \in \mathcal{E}_H \text{ and } E_1 \setminus H \approx_{hc} E_2\}$$

is a weak bisimulation. This is sufficient to say that $E \in P\_BNDC$. In fact, by Lemma 3.6, for every state E' reachable from E there exists a state $E'' \setminus H$ reachable from $E \setminus H$ such that $E'' \setminus H \approx_{hc} E'$. Hence, by definition of $\mathcal{S}$, we have that for all $\Pi \in \mathcal{E}_H$, $(E'' \setminus H, (E'|\Pi) \setminus H) \in \mathcal{S}$. Since $\mathcal{S}$ is a weak bisimulation, we have that for all $\Pi \in \mathcal{E}_H$, $E'' \setminus H \approx (E'|\Pi) \setminus H$ and, in particular, $E'' \setminus H \approx E' \setminus H$. Since $\approx$ is an equivalence relation, by symmetry and transitivity, we have that for every E' reachable from E and for all $\Pi \in \mathcal{E}_H$, $E' \setminus H \approx (E'|\Pi) \setminus H$, i.e., $E \in P\_BNDC$.

The fact that $\mathcal{S}$ is a weak bisimulation follows from the following four cases. Let $(E_1 \setminus H, (E_2|\Pi) \setminus H) \in \mathcal{S}$.

Case 1. $E_1 \setminus H \xrightarrow{a} E_1' \setminus H$ with $a \notin H$. Thus, for all contexts $C[\_]$, $E_1 \setminus H \xrightarrow{a}_C E_1' \setminus H$. By the hypothesis that $E_1 \setminus H \approx_{hc} E_2$, for all contexts $C[\_]$ there exists E_2' such that $E_2 \stackrel{\hat{a}}{\Longrightarrow}_C E_2'$ and $E_1' \setminus H \approx_{hc} E_2'$. Hence there exists E_2' such that $(E_2|\Pi) \setminus H \stackrel{\hat{a}}{\Longrightarrow} (E_2'|\Pi') \setminus H$ and, by definition of $\mathcal{S}$, $(E_1' \setminus H, (E_2'|\Pi') \setminus H) \in \mathcal{S}$.

Case 2. $(E_2|\Pi) \setminus H \xrightarrow{a} (E_2'|\Pi) \setminus H$ where also $E_2 \setminus H \xrightarrow{a} E_2' \setminus H$ and $a \notin H$. Let $C[\_]$ be the context $(\_|\mathbf{0}) \setminus H$. Hence $E_2 \xrightarrow{a}_C E_2'$. By the hypothesis that $E_1 \setminus H \approx_{hc} E_2$, there exists E_1' such that $E_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow}_C E_1' \setminus H$ and $E_1' \setminus H \approx_{hc} E_2'$. Since $C[E_1 \setminus H]$ can only perform actions of $E_1 \setminus H$ or τ actions, we have that $E_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow} E_1' \setminus H$ and, by definition of $\mathcal{S}$, $(E_1' \setminus H, (E_2'|\Pi) \setminus H) \in \mathcal{S}$.

Case 3. $(E_2|\Pi) \setminus H \xrightarrow{\tau} (E_2|\Pi') \setminus H$ with $\Pi \xrightarrow{\tau} \Pi'$. By definition of $\mathcal{S}$, it trivially follows that $(E_1 \setminus H, (E_2|\Pi') \setminus H) \in \mathcal{S}$.

Case 4. $(E_2|\Pi) \setminus H \xrightarrow{\tau} (E_2'|\Pi') \setminus H$ where $E_2 \xrightarrow{a} E_2'$, $\Pi \xrightarrow{\bar{a}} \Pi'$ and $a \in H$. Let $C[\_]$ be the context $(\_|\bar{a}.\mathbf{0}) \setminus H$. Hence $E_2 \xrightarrow{\tau}_C E_2'$. By the hypothesis that $E_1 \setminus H \approx_{hc} E_2$, there exists E_1' such that $E_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow}_C E_1' \setminus H$ and $E_1' \setminus H \approx_{hc} E_2'$. Since $C[E_1 \setminus H]$ can only perform actions of $E_1 \setminus H$ or τ actions, we have that $E_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow} E_1' \setminus H$ and, by definition of $\mathcal{S}$, $(E_1' \setminus H, (E_2'|\Pi) \setminus H) \in \mathcal{S}$.

We now show that if $E \in P\_BNDC$ then $E \setminus H \approx_{hc} E$. To this end it is sufficient to prove that

$$\mathcal{S} \quad = \quad \{(E_1 \setminus H, E_2) \mid E_1 \setminus H \approx E_2 \setminus H \text{ and } E_2 \in P\_BNDC\}$$

is a weak bisimulation on high contexts. This follows from the following cases. Let $C[\_]$ be an high context.

Case 1. $E_1 \setminus H \xrightarrow{a}_C E_1' \setminus H$ with $E_1 \setminus H \xrightarrow{a} E_1' \setminus H$. From the hypothesis that $E_1 \setminus H \approx E_2 \setminus H$, we have that there exists E_2' such that $E_2 \setminus H \stackrel{\hat{a}}{\Longrightarrow} E_2' \setminus H$ and $E_1' \setminus H \approx E_2' \setminus H$. Moreover, since $E_2 \in P\_BNDC$ also $E_2' \in P\_BNDC$. From the fact that $E_2 \setminus H \stackrel{\hat{a}}{\Longrightarrow} E_2' \setminus H$ we have that $E_2 \stackrel{\hat{a}}{\Longrightarrow} E_2'$. Since $a \notin H$, $E_2 \stackrel{\hat{a}}{\Longrightarrow}_C E_2'$ and, by definition of $\mathcal{S}$, $(E_1' \setminus H, E_2') \in \mathcal{S}$.

Case 2. $E_1 \setminus H \xrightarrow{\tau}_C E'_1 \setminus H$ with $E_1 \equiv E'_1$. In this case, by definition of $\mathcal{S}$, we immediately have $(E'_1 \setminus H, E_2) \in \mathcal{S}$.

Case 3. $E_2 \xrightarrow{a}_C E'_2$ with $E_2 \setminus H \xrightarrow{a} E'_2 \setminus H$. Since $E_1 \setminus H \approx E_2 \setminus H$, there exists E'_1 such that $E_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow} E'_1 \setminus H$ and $E'_1 \setminus H \approx E'_2 \setminus H$. Moreover, since $E_2 \in P\_BNDC$ also $E'_2 \in P\_BNDC$. Hence $E_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow}_C E'_1 \setminus H$ and, by definition of $\mathcal{S}$, $(E'_1 \setminus H, E'_2) \in \mathcal{S}$.

Case 4. $E_2 \xrightarrow{\tau}_C E'_2$ with $E_2 \xrightarrow{a} E'_2$ and $a \in H$. Then, $E_2/H \xrightarrow{\tau} E'_2/H$. From the fact that $E_1 \setminus H \approx E_2 \setminus H$ and $E_2 \in P\_BNDC$, we have that $E_1 \setminus H \approx E_2/H$ and thus there exists E'_1 such that $E_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow} E'_1 \setminus H$ and $E'_1 \setminus H \approx E'_2/H$. Moreover, since $E_2 \in P\_BNDC$ also $E'_2 \in P\_BNDC$ and hence $E'_1 \setminus H \approx E'_2 \setminus H$. Thus $E_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow}_C E'_1 \setminus H$ and, by definition of $\mathcal{S}$, $(E'_1 \setminus H, E'_2) \in \mathcal{S}$.

Case 5. $E_2 \xrightarrow{\tau}_C E'_2$ with $E_2 \equiv E'_2$. In this case, by definition of $\mathcal{S}$, we immediately have $(E_1 \setminus H, E'_2) \in \mathcal{S}$. □

3.2 Avoiding the Universal Quantifications

We show now how it is possible to give a characterization of *P_BNDC* avoiding both the universal quantification over all the possible high level processes, which is present in the *BNDC* basic definition, and the universal quantification over all the possible reachable states, required by the definition of *P_BNDC* itself.

In the previous subsection, we have shown how the idea of "being secure in every state" can be directly moved inside the bisimulation equivalence notion ($\approx_{hc}$). However, the notion of weak bisimulation on high contexts implicitly contains a quantification over all possible high contexts. We show here that the same equivalence notion ($\approx_{hc}$), may be expressed in a rather simpler way by exploiting local information only. This can be done by defining a novel equivalence relation which focuses only on observable actions that do not belong to H.

More in detail, we define an observation equivalence where actions from H *may* be ignored, i.e., they may be matched by zero or more τ actions. To this end, we use a transition relation which enables internal actions to be treated as high level ones.

Definition 3.8. Let $a \in Act$. We define the transition relation $\stackrel{\hat{a}}{\Longrightarrow}_{\setminus H}$ as follows:

$$\stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} \quad = \quad \begin{cases} \stackrel{\hat{a}}{\Longrightarrow} & \text{if } a \notin H \\ \stackrel{a}{\Longrightarrow} \cup \stackrel{\hat{\tau}}{\Longrightarrow} & \text{if } a \in H \end{cases}$$

Notice that the relation $\stackrel{\hat{a}}{\Longrightarrow}_{\setminus H}$ is a generalization of the relation $\stackrel{\hat{a}}{\Longrightarrow}$ used in the definition of weak bisimulation [38]. In fact, if $H = \emptyset$ then for all $a \in Act$, $E \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} E'$ coincides with $E \stackrel{\hat{a}}{\Longrightarrow} E'$.

We define the concept of weak bisimulation up to H.

Definition 3.9 (Weak Bisimulation up to H). A binary relation $\mathcal{S} \subseteq \mathcal{E} \times \mathcal{E}$ over processes is a *weak bisimulation up to H* if $(E, F) \in \mathcal{S}$ implies, for all $a \in Act$,

- whenever $E \xrightarrow{a} E'$, then there exists F' such that $F \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} F'$ and $(E', F') \in \mathcal{S}$;

- whenever $F \xrightarrow{a} F'$, then there exists E' such that $E \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} E'$ and $(E',F') \in \mathcal{S}$.

We say that two processes E,F are *weakly bisimilar up to* H, written $E \approx_{\setminus H} F$, if $(E,F) \in \mathcal{S}$ for some weak bisimulation $\mathcal{S}$ up to H.

From the above definition we may equivalently define $\approx_{\setminus H}$ as follows:

$$\approx_{\setminus H} = \bigcup\{\mathcal{S} : \mathcal{S} \text{ is a weak bisimulation up to } H\}.$$

It is easy to prove that $\approx_{\setminus H}$ is the largest weak bisimulation up to H and it is an equivalence relation.

The next theorem shows that the binary relations $\approx_{hc}$ and $\approx_{\setminus H}$ are equivalent. Intuitively, in the definition of $E \approx_{\setminus H} F$, each high level action of the process E may be simulated by a sequence of invisible actions of the process F. This corresponds to the fact that, in $E \approx_{hc} F$, an interaction of the process E with a high level context may be simulated by a sequence of internal transitions of F.

Theorem 3.10. *Let $E,F \in \mathcal{E}$. Then, $E \approx_{hc} F$ iff $E \approx_{\setminus H} F$.*

Proof. We first show that $E \approx_{hc} F$ implies $E \approx_{\setminus H} F$. To this end it is sufficient to prove that

$$\mathcal{S} \quad = \quad \{(E,F) \mid \quad E \approx_{hc} F\}$$

is a weak bisimulation up to H. This follows from the following two cases.

Case 1. $E \xrightarrow{a} E'$ with $a \notin H$. Let $C[\_]$ be the high context $(\_|\mathbf{0}) \setminus H$. Then $E \xrightarrow{a}_C E'$. From the fact that $E \approx_{hc} F$ it follows that there exists F' such that $F \stackrel{\hat{a}}{\Longrightarrow}_C F'$ and $E' \approx_{hc} F'$. By the choice of $C[\_]$, we also have that $F \stackrel{\hat{a}}{\Longrightarrow} F'$ and, since $a \notin H$, $F \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} F'$. Moreover, by definition of $\mathcal{S}$, $(E',F') \in \mathcal{S}$.

Case 2. $E \xrightarrow{a} E'$ with $a \in H$. Let $C[\_]$ be the high context $\equiv (\_|\bar{a}.\mathbf{0}) \setminus H$. Then $E \xrightarrow{\tau}_C E'$. From the fact that $E \approx_{hc} F$ it follows that there exists F' such that $F \stackrel{\hat{\tau}}{\Longrightarrow}_C F'$ and $E' \approx_{hc} F'$. By the choice of $C[\_]$, we also have that either $F \stackrel{\hat{\tau}}{\Longrightarrow} F'$ or $F \stackrel{\hat{a}}{\Longrightarrow} F'$ and, since $a \in H$, $F \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} F'$. Moreover, by definition of $\mathcal{S}$, $(E',F') \in \mathcal{S}$.

We now show that $E \approx_{\setminus H} F$ implies $E \approx_{hc} F$. To this end it is sufficient to prove that

$$\mathcal{S} \quad = \quad \{(E,F) \mid \quad E \approx_{\setminus H} F\}$$

is a weak bisimulation on high contexts. This follows from the following two cases.

Let $C[\_]$ be a high context.

Case 1. $E \xrightarrow{a}_C E'$ with $E \xrightarrow{a} E'$ and $a \notin H$. Since $E \approx_{\setminus H} F$, there exists F' such that $F \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} F'$ and $E' \approx_{\setminus H} F'$. Since $a \notin H$, we also have $F \stackrel{\hat{a}}{\Longrightarrow} F'$. Thus $F \stackrel{\hat{a}}{\Longrightarrow}_C F'$ and, by definition of $\mathcal{S}$, $(E',F') \in \mathcal{S}$.

Case 2. $E \xrightarrow{\tau}_C E'$ with $E \xrightarrow{a} E'$ and $a \in H$. Since $E \approx_{\setminus H} F$, there exists F' such that $F \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} F'$ and $E' \approx_{\setminus H} F'$. Thus either $F \stackrel{\hat{\tau}}{\Longrightarrow} F'$ or $F \stackrel{\hat{a}}{\Longrightarrow} F'$. Since the high context $C[\_]$ may synchronize on a by performing the complementary action $\bar{a}$, we have that $F \stackrel{\hat{\tau}}{\Longrightarrow}_C F'$ and $(E',F') \in \mathcal{S}$. $\square$

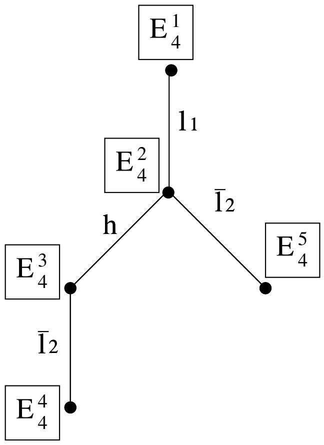

Figure 5: The process E_4

Theorem 3.10 allows us to identify a local property of processes (with no quantification on the states and on the high contexts) which is a necessary and sufficient condition for *P_BNDC*. This is stated by the following corollary.

Corollary 3.11. *Let $E \in \mathcal{E}$. $E \in P\_BNDC$ iff $E \setminus H \approx_{\setminus H} E$.*

Example 3.12. Consider the process $E_4 \stackrel{\text{def}}{=} l_1.(h.\bar{l}_2.\mathbf{0} + \bar{l}_2.\mathbf{0})$ of Example 2.5. We show that E_4 is *P_BNDC*.

We label the states reachable from E_4 as depicted in Figure 5 and construct the binary relation $\mathcal{S}$ as follows:

$$\mathcal{S} \quad = \quad \{(E_4^i \setminus H, E_4^i) \mid i \in [1 \ldots 5]\} \quad \cup \quad \{(E_4^2 \setminus H, E_4^3), (E_4^5 \setminus H, E_4^4)\}$$

We prove that $\mathcal{S}$ is a weak bisimulation up to H. This follows from the following cases. Let us first consider $a \in \{l_1, \bar{l}_2\}$. Notice that, since a is a low action, $\stackrel{\hat{a}}{\Longrightarrow}_{\setminus H}$ coincides with $\stackrel{\hat{a}}{\Longrightarrow}$. Thus, for all $i \in [1 \ldots 5]$ and $(E_4^i \setminus H, E_4^i) \in \mathcal{S}$ we trivially have that:

- $E_4^i \setminus H \stackrel{a}{\to} E_4^j \setminus H$. In this case, $E_4^i \stackrel{a}{\to} E_4^j$, and so $E_4^i \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} E_4^j$ and, by definition of $\mathcal{S}$, $(E_4^j \setminus H, E_4^j) \in \mathcal{S}$.
- $E_4^i \stackrel{a}{\to} E_4^j$. In this case, $E_4^i \setminus H \stackrel{a}{\to} E_4^j \setminus H$, and so $E_4^i \setminus H \stackrel{\hat{a}}{\Longrightarrow}_{\setminus H} E_4^j \setminus H$ and, by definition of $\mathcal{S}$, $(E_4^j \setminus H, E_4^j) \in \mathcal{S}$.

To conclude the proof we also need to consider the high level transitions and the remaining pairs $(E_4^2 \setminus H, E_4^3), (E_4^5 \setminus H, E_4^4) \in \mathcal{S}$:

- $(E_4^2 \setminus H, E_4^2) \in \mathcal{S}$ and $E_4^2 \stackrel{h}{\to} E_4^3$. In this case, the high level action is simulated by no moves, i.e., $E_4^2 \setminus H \stackrel{\hat{h}}{\Longrightarrow}_{\setminus H} E_4^2 \setminus H$ with $(E_4^2 \setminus H, E_4^3) \in \mathcal{S}$.

- $(E_4^2 \setminus H, E_4^3) \in \mathcal{S}$ and $E_4^2 \setminus H \xrightarrow{\bar{l}_2} E_4^5 \setminus H$. In this case, $E_4^3 \stackrel{\hat{\bar{l}}_2}{\Longrightarrow}_{\setminus H} E_4^4$ and, by definition of $\mathcal{S}$, $(E_4^5 \setminus H, E_4^4) \in \mathcal{S}$.

- $(E_4^2 \setminus H, E_4^3) \in \mathcal{S}$ and $E_4^3 \xrightarrow{\bar{l}_2} E_4^4$. In this case, $E_4^2 \setminus H \stackrel{\hat{\bar{l}}_2}{\Longrightarrow}_{\setminus H} E_4^5 \setminus H$ and $(E_4^5 \setminus H, E_4^4) \in \mathcal{S}$.

- $(E_4^5 \setminus H, E_4^4) \in \mathcal{S}$. Nothing to prove since both of these processes have no transitions. $\square$

In practice, we have proven that a process is *P_BNDC* if and only if it is equivalent – with respect to a particular bisimulation based equivalence relation – to the same process prevented from performing high level actions. This property is particularly appealing since it suggests the effective computability of *P_BNDC*. In particular, as we discuss in the concluding section, one may perform the *P_BNDC* check using already existing tools at a low time complexity.

3.3 Properties of P_BNDC

In this subsection we show that property *P_BNDC* is equivalent to the already proposed security property *SBSNNI* (Strong Bisimulation-based SNNI, where *SNNI* stands for Strong Non-deterministic Non-Interference, see [16, 17]) and we prove that it is compositional with respect to both parallel and prefix operators.

The security property *SBSNNI* was defined in [16, 17] as follows.

Definition 3.13 (SBSNNI). Let $E \in \mathcal{E}$.

$$E \in SBSNNI \text{ if } \forall\, E' \text{ reachable from } E,\ E' \setminus H \approx E'/H.$$

This property was introduced to automatically check *BNDC*, i.e., to bypass the quantification over all the possible malicious high level processes. As it follows from Proposition 3.16, *SBSNNI* is strictly stronger than *BNDC*, since, quite interestingly, it is equivalent to *P_BNDC*.

Before proving Proposition 3.16 we recall from [17] the next definition and result.

Definition 3.14. [17] Let $E \in \mathcal{E}$. Then $E \in BSNNI$ if $E \setminus H \approx E/H$.

Proposition 3.15. [17] *BNDC* $\subset$ *BSNNI*.

Notice that *SBSNNI* corresponds to requiring *BSNNI* over all the reachable states. Notice also that the inclusion of Proposition 3.15 is strict; this can be seen by considering, for instance, the process $E \stackrel{\text{def}}{=} l.h.l.h.l.\mathbf{0} + l.l.l.\mathbf{0} + l.\mathbf{0}$ which is *BSNNI* but not *BNDC*. In fact, by restricting or hiding all the high level actions we only observe the cases in which either zero or all h's are performed. Thus $E \setminus H \approx E/H \approx l.l.l.\mathbf{0} + l.\mathbf{0}$. *BNDC*, instead, also considers the "intermediate" situation in which only the first h is executed. Formally, $(E \mid \bar{h}.\mathbf{0}) \setminus H \approx l.l.\mathbf{0} + l.l.l.\mathbf{0} + l.\mathbf{0} \not\approx l.l.l.\mathbf{0} + l.\mathbf{0} \approx E \setminus H$.

Proposition 3.16. *P_BNDC* $=$ *SBSNNI*.

Proof. We first prove that $P\_BNDC \subseteq SBSNNI$. Let $E \in P\_BNDC$. By definition of P_BNDC, for all E' reachable from E, $E' \in BNDC$ and then, by Proposition 3.15, $E' \in BSNNI$. Hence, by Definition 3.14, for all E' reachable from E, $E' \setminus H \approx E'/H$, i.e., $E \in SBNNI$.

In order to prove that $SBSNNI \subseteq P\_BNDC$ we show that

$$\mathcal{S} \quad = \quad \{(E_1 \setminus H, (E_2|\Pi) \setminus H) \mid \quad \Pi \in \mathcal{E}_H \text{ and } E_1 \setminus H \approx E_2/H \text{ and } E_2 \in SBSNNI\}$$

is a weak bisimulation. This is sufficient to say that if $E \in SBSNNI$ then $E \in P\_BNDC$. In fact, by persistence of $SBSNNI$, this proves that for all E' reachable from E and for all $\Pi \in \mathcal{E}_H$, $E' \setminus H \approx (E'|\Pi) \setminus H$, i.e., $E' \in BNDC$.

The fact that $\mathcal{S}$ is a weak bisimulation follows from the following four cases. Let $(E_1 \setminus H, (E_2|\Pi) \setminus H) \in \mathcal{S}$.

Case 1. $E_1 \setminus H \xrightarrow{a} E_1' \setminus H$ with $a \notin H$. By the hypothesis that $E_1 \setminus H \approx E_2/H$, there exists E_2' such that $E_2/H \stackrel{\hat{a}}{\Longrightarrow} E_2'/H$ and $E_1' \setminus H \approx E_2'/H$. Since $E_2 \in SBSNNI$, $E_2 \setminus H \approx E_2/H$ and thus there exists E_2'' such that $E_2 \setminus H \stackrel{\hat{a}}{\Longrightarrow} E_2'' \setminus H$ and $E_2'/H \approx E_2'' \setminus H$. Hence $(E_2|\Pi) \setminus H \stackrel{\hat{a}}{\Longrightarrow} (E_2''|\Pi) \setminus H$. By persistence of $SBSNNI$, $E_2'' \in SBSNNI$ and in particular $E_2'' \setminus H \approx E_2''/H$; moreover, by transitivity of $\approx$, $E_1' \setminus H \approx E_2''/H$. Therefore, by definition of $\mathcal{S}$, $(E_1' \setminus H, (E_2''|\Pi) \setminus H) \in \mathcal{S}$.

Case 2. $(E_2|\Pi) \setminus H \xrightarrow{a} (E_2'|\Pi) \setminus H$ where also $E_2 \setminus H \xrightarrow{a} E_2' \setminus H$ and $a \notin H$. By the hypothesis that $E_2 \in SBSNNI$, it follows that $E_2 \setminus H \approx E_2/H$. Hence there exists E_2'' such that $E_2/H \stackrel{\hat{a}}{\Longrightarrow} E_2''/H$ and $E_2' \setminus H \approx E_2''/H$. By the hypothesis that $E_1 \setminus H \approx E_2/H$, there exists E_1' such that $E_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow} E_1' \setminus H$ and $E_1' \setminus H \approx E_2''/H$. By persistence of $SBSNNI$, $E_2' \in SBSNNI$ and in particular $E_2' \setminus H \approx E_2'/H$; moreover, by transitivity of $\approx$, $E_1' \setminus H \approx E_2'/H$. Therefore, by definition of $\mathcal{S}$, $(E_1' \setminus H, (E_2'|\Pi) \setminus H) \in \mathcal{S}$.

Case 3. $(E_2|\Pi) \setminus H \xrightarrow{\tau} (E_2|\Pi') \setminus H$ with $\Pi \xrightarrow{\tau} \Pi'$. By definition of $\mathcal{S}$, it trivially follows that $(E_1 \setminus H, (E_2|\Pi') \setminus H) \in \mathcal{S}$.

Case 4. $(E_2|\Pi) \setminus H \xrightarrow{\tau} (E_2'|\Pi') \setminus H$ where $E_2 \xrightarrow{a} E_2'$, $\Pi \xrightarrow{\bar{a}} \Pi'$ and $a \in H$. Hence $E_2/H \xrightarrow{\tau} E_2'/H$. By the hypothesis that $E_1 \setminus H \approx E_2/H$, there exists E_1' such that $E_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow} E_1' \setminus H$ and $E_1' \setminus H \approx E_2'/H$. By persistence of $SBSNNI$, $E_2' \in SBSNNI$ and thus, by definition of $\mathcal{S}$, $(E_1' \setminus H, (E_2'|\Pi) \setminus H) \in \mathcal{S}$. $\square$

In [17] it is proved that $SBSNNI$ is *compositional*, in the sense that it is preserved by the parallel and restriction operators (statements (1) and (2) of Proposition 3.17 below). It is easy to prove that P_BNDC is also compositional with respect to the prefix operator limited to low level actions (statement (3) of Proposition 3.17).

Proposition 3.17.

(1) *If* $E, F \in P\_BNDC$ *then* $(E|F) \in P\_BNDC$*;*

(2) *if* $E \in P\_BNDC$ *and* $v \subseteq \mathcal{L}$ *then* $E \setminus v \in P\_BNDC$*;*

(3) *if* $E \in P\_BNDC$ *and* $a \in L \cup \{\tau\}$ *then* $a.E \in P\_BNDC$*.*

Property *P_BNDC* is not compositional with respect to the nondeterministic choice operator as illustrated below.

Example 3.18. Let $E \stackrel{\text{def}}{=} h.\mathbf{0}$ with $h \in H$ and $F \stackrel{\text{def}}{=} l.\mathbf{0}$ with $l \in L$. It is easy to see that both E and F are *P_BNDC* but $E+F$ is not *P_BNDC*. In fact, the transition $E+F \xrightarrow{h} \mathbf{0}$ cannot be simulated by process $(E+F) \setminus H$ which cannot reach any state equivalent to $\mathbf{0}$ by performing (a possibly empty) sequence of silent actions. The problem lies in the fact that process E adds a deadlock state that may be triggered by a high level action h, making the whole system insecure. $\square$

Inspired by the above example, in [6] we introduce a novel Non-Interference property, named *PP_BNDC* which is compositional with respect to the nondeterministic choice operator. This property is a variation of *P_BNDC* whose underlying observation equivalence is the progressing bisimulation relation (see [40]). *PP_BNDC* is very useful for verification purposes because it is fully compositional with respect to SPA operators, but it might be too strong in some situations. To give a trivial example, the process $h.\mathbf{0}$ above is not *PP_BNDC* but it is intuitively secure, as it can only perform a single high level action, thus no interference is ever possible because nothing is observable from the low level interface. Hence, *PP_BNDC* is mainly useful as a sufficient condition to check desired security properties like, e.g., *BNDC* and *P_BNDC*.

4 Security of Mobile Agents

In this section we show that the notion of *P_BNDC* can be successfully applied to the study of mobile process security. We proceed in two steps: First, we formally prove that *P_BNDC* guarantees security even when the high context is completely dynamic, i.e., when it can arbitrarily change at run-time. Since, from the point of view of agents, a migration causes a change of the surrounding execution environment, we obtain that *P_BNDC* guarantees security even when agents non-deterministically migrate in every possible way during their executions.

This first step gives a new, sound and complete, characterization of *P_BNDC* in terms of dynamic contexts, substantiating the intuition that the persistent property *P_BNDC* is strong enough to handle agent migrations. However, this is proved at a very high level of abstraction, modelling migrations as context changes not controlled by agents.

Our second step is, thus, to consider a more concrete model obtained by adding explicit process mobility to the SPA language. We formally prove that *P_BNDC* implies the natural extension of Non-Interference in this concrete model of mobile agents.

4.1 Towards mobility: a characterization of *P_BNDC* through Dynamic Contexts

In this section we show that *P_BNDC* is equivalent to an extension of *BNDC* to *dynamic high contexts*, i.e., high contexts that may change arbitrarily at any step of the computation. This dynamic behaviour of the contexts is modelled through the new *dynamic parallel composition* operator that allows contexts to arbitrarily change thanks to a special internal action $\tau_{\mathcal{D}} \notin Act$. A dynamic high context is a high context where dynamic parallel composition is used in place of standard parallel composition.

Definition 4.1 (**Dynamic high context).** A *dynamic high context* $C_{\mathcal{D}}[\_]$ is a term $(\_ \mid_{\mathcal{D}} \Pi) \setminus H$ where $\Pi \in \mathcal{E}_H$ and $\mid_{\mathcal{D}}$ represents the *dynamic parallel composition* which extends the semantics of parallel composition as follows:

$$\frac{E \mid \Pi \xrightarrow{a} E' \mid \Pi'}{E \mid_{\mathcal{D}} \Pi \xrightarrow{a} E' \mid_{\mathcal{D}} \Pi'} \qquad \frac{\Pi'' \in \mathcal{E}_H}{E \mid_{\mathcal{D}} \Pi \xrightarrow{\tau_{\mathcal{D}}} E' \mid_{\mathcal{D}} \Pi''}$$

We now define a variant of weak bisimulation that requires context changes to be simulated by no moves. We consider the set $\mathcal{E}_{\mathcal{D}} = \mathcal{E} \cup \{C_{\mathcal{D}}[E] \mid \forall C_{\mathcal{D}}[\_], \forall E \in \mathcal{E}\}$ of all SPA processes possibly executed in dynamic high contexts.

Definition 4.2 (**Dynamic Weak Bisimulation).** A binary relation $\mathcal{S} \subseteq \mathcal{E}_{\mathcal{D}} \times \mathcal{E}_{\mathcal{D}}$ is a *dynamic weak bisimulation* if $(E,F) \in \mathcal{S}$ implies, for all $a \in Act$, $a \neq \tau_{\mathcal{D}}$,

- whenever $E \xrightarrow{a} E'$, then there exists F' such that $F \stackrel{\hat{a}}{\Longrightarrow} F'$ and $(E',F') \in \mathcal{S}$;
- whenever $F \xrightarrow{a} F'$, then there exists E' such that $E \stackrel{\hat{a}}{\Longrightarrow} E'$ and $(E',F') \in \mathcal{S}$;
- whenever $E \xrightarrow{\tau_{\mathcal{D}}} E'$ then $(E',F) \in \mathcal{S}$.
- whenever $F \xrightarrow{\tau_{\mathcal{D}}} F'$ then $(E,F') \in \mathcal{S}$.

$E,F \subseteq \mathcal{E}_{\mathcal{D}}$ are *dynamically weakly bisimilar*, denoted by $E \approx_{\mathcal{D}} F$, if there exists a dynamic weak bisimulation $\mathcal{S}$ containing the pair (E,F).

It is possible to prove that $\approx_{\mathcal{D}}$ is the largest dynamic weak bisimulation and it is symmetric and transitive, but, differently from usual bisimulation notions, it is not reflexive, as shown by the following example.

Example 4.3. Consider the (insecure) process $h.l$ and the dynamic high context $C_{\mathcal{D}}[\_] = (\_ \mid_{\mathcal{D}} \mathbf{0}) \setminus H$. We have that $C_{\mathcal{D}}[h.l] \not\approx_{\mathcal{D}} C_{\mathcal{D}}[h.l]$, i.e., $\approx_{\mathcal{D}}$ is not reflexive. In fact, if we had $(h.l \mid_{\mathcal{D}} \mathbf{0}) \setminus H \approx_{\mathcal{D}} (h.l \mid_{\mathcal{D}} \mathbf{0}) \setminus H$, it should also hold that $(h.l \mid_{\mathcal{D}} \bar{h}) \setminus H \approx_{\mathcal{D}} (h.l \mid_{\mathcal{D}} \mathbf{0}) \setminus H$, since $(h.l \mid_{\mathcal{D}} \mathbf{0}) \setminus H \xrightarrow{\tau_{\mathcal{D}}} (h.l \mid_{\mathcal{D}} \bar{h}) \setminus H$. But this is not true since $(h.l \mid_{\mathcal{D}} \bar{h}) \setminus H \xrightarrow{\tau} (l \mid_{\mathcal{D}} \mathbf{0}) \setminus H$ and $(h.l \mid_{\mathcal{D}} \mathbf{0}) \setminus H$ cannot simulate this step. In fact, it cannot perform any τ's and simulating by not moving is unsuccessful given that action l, possibly performed by $(l \mid_{\mathcal{D}} \mathbf{0}) \setminus H$, cannot be simulated by $(h.l \mid_{\mathcal{D}} \mathbf{0}) \setminus H$. $\square$

The example above shows that $\approx_{\mathcal{D}}$ is not reflexive in general. This is due to the fact that the condition on $\tau_{\mathcal{D}}$ transitions is very strong, requiring that context changes do not change the observable behaviour. As we will see, $\approx_{\mathcal{D}}$ is reflexive whenever the process inside the dynamic context is secure. In fact, the behaviour of a secure process is required to be the same with respect to all high level contexts.

We call $BNDC_{\mathcal{D}}$ the extension of $BNDC$ to dynamic contexts and dynamic bisimulation:

Definition 4.4 ($BNDC_{\mathcal{D}}$)**.** Let $E \in \mathcal{E}$.

$$E \in BNDC_{\mathcal{D}} \text{ if } \forall \text{ dynamic high contexts } C_{\mathcal{D}}[\_],\ E \setminus H \approx_{\mathcal{D}} C_{\mathcal{D}}[E].$$

The following lemma states that checking equivalence with respect to one dynamic context is sufficient to prove equivalence with respect to all dynamic contexts. It directly derives from the fact that $C_{\mathcal{D}}[F] \xrightarrow{\tau_{\mathcal{D}}} C'_{\mathcal{D}}[F]$, and $\tau_{\mathcal{D}}$ is required to be simulated by zero moves:

Lemma 4.5. *Let $E, F \in \mathcal{E}$. If $E \approx_{\mathcal{D}} C_{\mathcal{D}}[F]$ iff $E \approx_{\mathcal{D}} C'_{\mathcal{D}}[F]$ for all dynamic contexts $C'_{\mathcal{D}}[\_]$.*

We give two simple characterizations of $BNDC_{\mathcal{D}}$ that help understanding the power of dynamic contexts and bisimulation. The first characterization states that, since contexts are dynamic, it is sufficient to check just one of them (e.g., $\mathbf{0}$); the second one shows that reflexivity of $\approx_{\mathcal{D}}$ holds if and only if E is secure.

Proposition 4.6. *Let $E \in \mathcal{E}$. $E \in BNDC_{\mathcal{D}}$ iff one of the following holds:*

- $E \setminus H \approx_{\mathcal{D}} (E \mid_{\mathcal{D}} \mathbf{0}) \setminus H$;
- $\forall C_{\mathcal{D}}[\_], C_{\mathcal{D}}[E] \approx_{\mathcal{D}} C_{\mathcal{D}}[E]$.

Proof. The first condition is a direct consequence of Lemma 4.5. The $\Rightarrow$ implication of the second one derives by transitivity and symmetry of $\approx_{\mathcal{D}}$ since $E \setminus H \approx_{\mathcal{D}} C_{\mathcal{D}}[E]$ implies $C_{\mathcal{D}}[E] \approx_{\mathcal{D}} E \setminus H$ and, by transitivity, $C_{\mathcal{D}}[E] \approx_{\mathcal{D}} C_{\mathcal{D}}[E]$. The $\Leftarrow$ implication, instead, can be proved by showing that $\mathcal{R} = \{(E \setminus H, C_{\mathcal{D}}[F]) \mid C'_{\mathcal{D}}[E] \approx_{\mathcal{D}} C_{\mathcal{D}}[F], \forall C'_{\mathcal{D}}[\_]\}$ is a dynamic weak bisimulation. The proof is trivial by case analysis and by exploiting the null context $C'_{\mathcal{D}}[\_] = (\_ \mid_{\mathcal{D}} \mathbf{0}) \setminus H$. $\square$

We illustrate how dynamic contexts work through the following simple example.

Example 4.7. Consider again the process E_3 of Example 2.5 and illustrated in Figure 3.

$$E_3 = l_1.h.\bar{l}_2.\mathbf{0} + l_1.(\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0})$$

We have seen that E_3 is *BNDC* but not *P_BNDC*. The problem is after the leftmost l_1, since the continuation process $h.\bar{l}_2.\mathbf{0}$ is not secure. We show that $BNDC_{\mathcal{D}}$ captures this problem, too. We reason by contradiction: let us assume that E_3 is $BNDC_{\mathcal{D}}$. Then, if $C_{\mathcal{D}}[E_3] \stackrel{l_1}{\Longrightarrow} C_{\mathcal{D}}[h.\bar{l}_2.\mathbf{0}]$, for a certain dynamic high context $C_{\mathcal{D}}[\_]$, we have that $E_3 \setminus H \stackrel{\hat{l}_1}{\Longrightarrow} E'_3 \setminus H$ and $C_{\mathcal{D}}[h.\bar{l}_2.\mathbf{0}] \approx_{\mathcal{D}} E'_3 \setminus H$. By Lemma 4.5 and by transitivity of $\approx_{\mathcal{D}}$, we have that $C_{\mathcal{D}}[h.\bar{l}_2.\mathbf{0}] \approx_{\mathcal{D}} C'_{\mathcal{D}}[h.\bar{l}_2.\mathbf{0}]$, for all high dynamic contexts $C'_{\mathcal{D}}[\_]$. So, in particular, $(h.\bar{l}_2.\mathbf{0} \mid_{\mathcal{D}} \mathbf{0}) \setminus H \approx_{\mathcal{D}} (h.\bar{l}_2.\mathbf{0} \mid_{\mathcal{D}} \bar{h}) \setminus H$, giving a contradiction. $\square$

The next theorem shows that *P_BNDC* and $BNDC_{\mathcal{D}}$ are the same. Thus, *P_BNDC* guarantees security even when contexts may change arbitrarily.

Theorem 4.8. *Let $E \in \mathcal{E}$. $E \in BNDC_{\mathcal{D}}$ iff $E \in P\_BNDC$.*

Proof. ($\Leftarrow$) We show that $E \setminus H \approx_{hc} E$ implies $C_{\mathcal{D}}[E] \approx_{\mathcal{D}} E \setminus H$, for all dynamic high contexts $C_{\mathcal{D}}[\_]$. In order to do it we prove that

$$\mathcal{S} \quad = \quad \{(E_1 \setminus H, C_{\mathcal{D}}[E_2]) \ : \ E_1 \setminus H \approx_{hc} E_2 \text{ and } C_{\mathcal{D}}[\_] \text{ is a dynamic context}\}$$

is a dynamic weak bisimulation. This is sufficient to conclude that $C_{\mathcal{D}}[E] \approx_{\mathcal{D}} E \setminus H$, for all dynamic high contexts $C_{\mathcal{D}}[\_]$.

The fact that $\mathcal{S}$ is a dynamic weak bisimulation follows from the following five cases.

Let $(E_1 \setminus H, C_{\mathcal{D}}[E_2]) \in \mathcal{S}$.

Case 1. $E_1 \setminus H \xrightarrow{a} E'_1 \setminus H$ with $a \notin H$. Thus, for all high contexts $C[\_]$, we have $E_1 \setminus H \xrightarrow{a}_C E'_1 \setminus H$. By the hypothesis that $E_1 \setminus H \approx_{hc} E_2$, for all high contexts $C[\_]$ there exists E'_2 such that $E_2 \stackrel{\hat{a}}{\Longrightarrow}_C E'_2$ and $E'_1 \setminus H \approx_{hc} E'_2$. As a consequence, $C_{\mathcal{D}}[E_2] \stackrel{\hat{a}}{\Longrightarrow} C'_{\mathcal{D}}[E'_2]$ and $(E'_1 \setminus H, C'_{\mathcal{D}}[E'_2]) \in \mathcal{S}$.

Case 2. $C_{\mathcal{D}}[E_2] \xrightarrow{a} C_{\mathcal{D}}[E'_2]$ with $E_2 \setminus H \xrightarrow{a} E'_2 \setminus H$ and $a \notin H$ and $a \neq \tau_{\mathcal{D}}$. Let $C[\_] \equiv (\_|\mathbf{0}) \setminus H$. Hence $E_2 \xrightarrow{a}_C E'_2$. By the hypothesis that $E_1 \setminus H \approx_{hc} E_2$, there exists E'_1 such that $E_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow}_C E'_1 \setminus H$ and $E'_1 \setminus H \approx_{hc} E'_2$. Since $C[E_1 \setminus H]$ can only perform actions of $E_1 \setminus H$ and τ actions of the context, $E_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow} E'_1 \setminus H$ and, by definition of $\mathcal{S}$, $(E'_1 \setminus H, C_{\mathcal{D}}[E'_2]) \setminus H) \in \mathcal{S}$.

Case 3. $C_{\mathcal{D}}[E_2] \xrightarrow{\tau} C'_{\mathcal{D}}[E_2]$. By definition of $\mathcal{S}$, it immediately follows that $(E_1 \setminus H, C'_{\mathcal{D}}[E_2]) \in \mathcal{S}$.

Case 4. $C_{\mathcal{D}}[E_2] \xrightarrow{\tau} C'_{\mathcal{D}}[E'_2]$ where $E_2 \xrightarrow{a} E'_2$ and $a \in H$. Consider the high context $C[\_] \equiv (\_|\bar{a}.\mathbf{0}) \setminus H$. Hence $E_2 \xrightarrow{\tau}_C E'_2$. By the fact that $E_1 \setminus H \approx_{hc} E_2$, there exists E'_1 such that $E_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow}_C E'_1 \setminus H$ and $E'_1 \setminus H \approx_{hc} E'_2$. Since $C[E_1 \setminus H]$ can only perform actions of $E_1 \setminus H$ and τ actions of the context, we have $E_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow} E'_1 \setminus H$ and $(E'_1 \setminus H, C'_{\mathcal{D}}[E'_2]) \in \mathcal{S}$.

Case 5. $C_{\mathcal{D}}[E_2] \xrightarrow{\tau_{\mathcal{D}}} C'_{\mathcal{D}}[E_2]$.By definition of $\mathcal{S}$, it immediately follows that $(E_1 \setminus H, C'_{\mathcal{D}}[E_2]) \in \mathcal{S}$.

($\Rightarrow$) We now show that $C_{\mathcal{D}}[E] \approx_{\mathcal{D}} E \setminus H$, for all dynamic high contexts $C_{\mathcal{D}}[\_]$, implies $E \setminus H \approx_{hc} E$. In order to do it we prove that

$$\mathcal{R} \quad = \quad \{(E_1 \setminus H, E_2) \ : \ E_1 \setminus H \approx_{\mathcal{D}} C_{\mathcal{D}}[E_2] \text{ for all } C_{\mathcal{D}}[\_]\}$$

is a weak bisimulation on high contexts. This is clearly sufficient to conclude that $E \approx_{hc} E \setminus H$.

The fact that $\mathcal{R}$ is a weak bisimulation on high contexts follows from the following four cases.

Let $(E_1 \setminus H, C_{\mathcal{D}}[E_2]) \in \mathcal{R}$.

Case 1. $C[E_1 \setminus H] \xrightarrow{a} C[E'_1 \setminus H]$ with $E_1 \setminus H \xrightarrow{a} E'_1 \setminus H$ and $a \notin H$. By considering the dynamic context $C_{\mathcal{D}}$ obtained by C by replacing the parallel with $|_{\mathcal{D}}$, we trivially have that $C_{\mathcal{D}}[E_2] \xrightarrow{a} C'_{\mathcal{D}}[E'_2]$ with $E'_1 \setminus H \approx_{\mathcal{D}} C'_{\mathcal{D}}[E'_2]$. Since a cannot be equal to $\tau_{\mathcal{D}}$, we also obtain that $C[E_2] \xrightarrow{a} C'[E'_2]$. By Lemma 4.5, we obtain that $(E'_1 \setminus H, E'_2) \in \mathcal{R}$.

Case 2. $C[E_2] \xrightarrow{a} C[E_2']$ with $E_2 \setminus H \xrightarrow{a} E_2' \setminus H$ and $a \notin H$. We have $C_{\mathcal{D}}[E_2] \xrightarrow{a} C_{\mathcal{D}}[E_2']$ and $E_1 \setminus H \stackrel{\hat{a}}{\Longrightarrow} E_1' \setminus H$ with $E_1' \setminus H \approx_{\mathcal{D}} C_{\mathcal{D}}[E_2']$. From the fact that $C[E_1 \setminus H] \stackrel{\hat{a}}{\Longrightarrow} C[E_1' \setminus H]$ and by Lemma 4.5, we obtain that $(E_1' \setminus H, E_2') \in \mathcal{R}$.

Case 3. $C[E_2] \xrightarrow{\tau} C'[E_2]$ or $C[E_1 \setminus H] \xrightarrow{\tau} C'[E_1 \setminus H]$. There is no need to simulate these actions as $(E_1 \setminus H, E_2) \in \mathcal{R}$.

Case 4. $C[E_2] \xrightarrow{\tau} C'[E_2']$ where $E_2 \xrightarrow{a} E_2'$ and $a \in H$. Consider the dynamic high context $C_{\mathcal{D}}$ obtained from C by replacing the parallel with $|_{\mathcal{D}}$. Hence $C_{\mathcal{D}}[E_2] \xrightarrow{\tau} C'_{\mathcal{D}}[E_2']$ and $E_1 \setminus H \stackrel{\hat{\tau}}{\Longrightarrow} E_1' \setminus H$ with $E_1' \setminus H \approx_{\mathcal{D}} C'_{\mathcal{D}}[E_2']$. By Lemma 4.5 we obtain that $(E_1' \setminus H, C'_{\mathcal{D}}[E_2']) \in \mathcal{R}$. $\square$

4.2 Security of Mobile Agents in the MSPA calculus

In this section we extend the SPA calculus with a primitive for mobility, inspired by DPI-calculus [26], and call the new calculus MSPA (for *Mobility SPA*).

The syntax of MSPA processes is the same of SPA processes extended with the new migration primitive *goto l* with $l \in Loc$ where Loc is a set of locations (i.e., a set names ranged over by $l, k, s, \ldots$). As for DPI, processes must be distributed among locations in order to be executed. Distributed processes are called *systems* and have the following syntax:

$$M, N ::= l[\![E]\!] \mid M|N \mid M \setminus v$$

where E is a MSPA process, $l \in Loc$, $v \subseteq \mathcal{L}$. Intuitively, $l[\![E]\!]$ denotes the MSPA process E being executed at location l, $M|N$ is the parallel composition of systems M and N, and $M \setminus v$ is the system M in which actions of set v are prevented, independently of the location in which they are performed.

The semantics of MSPA processes is given by simply extending the set of SPA actions with the new mobility actions of the form *goto l* and by adding new axioms for them. Formally, the set of all actions of the MSPA language is $Act \cup \{goto\ l \mid l \in Loc\}$, and the set of rules of Figure 1 is extended with the new *Migration* axiom:

$$\frac{-}{goto\ l.E \xrightarrow{goto\ l} E}$$

The new migration primitive affects the execution of systems whose operational semantics is given through the rules reported in Figure 6. We denote by $Act_{\mathcal{M}}$ the set of actions of distributed systems that is $\{a@l \mid a \in \mathcal{L},\ l \in Loc\} \cup \{\ goto\ l \mid l \in Loc\} \cup \{\tau\}$.

Intuitively, a system $l[\![E]\!]$ executes all the I/O of agent E localized at location l (denoted by symbol @); whenever a *goto k* is performed by an agent E, the location of E becomes k; parallel systems may proceed by interleaving their execution sequences, possibly synchronizing on complementary input/output actions performed at the same location l; finally, as done for agents, action execution may be prevented through the restriction operator.

Example 4.9. Consider once more the process $E_3 = l_1.h.\bar{l}_2.\mathbf{0} + l_1.(\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0})$ of Examples 2.5 and 3.3 and depicted in Figure 3. Assume, now, that after performing the

$$\text{Localized I/O} \qquad \frac{E \xrightarrow{a} E'}{l[\![E]\!] \xrightarrow{a@l} l[\![E']\!]} \quad a \in \mathcal{L}$$

$$\text{Silent Action} \qquad \frac{E \xrightarrow{\tau} E'}{l[\![E]\!] \xrightarrow{\tau} l[\![E']\!]}$$

$$\text{Move} \qquad \frac{E \xrightarrow{goto\ k} E'}{l[\![E]\!] \xrightarrow{goto\ k} k[\![E']\!]}$$

$$\text{Parallel Systems} \qquad \frac{M \xrightarrow{\alpha} M'}{M|N \xrightarrow{\alpha} M'|N} \qquad \frac{N \xrightarrow{\alpha} N'}{M|N \xrightarrow{\alpha} M|N'} \quad \alpha \in Act_{\mathcal{M}}$$

$$\frac{M \xrightarrow{a@l} M' \quad N \xrightarrow{\bar{a}@l} N'}{M|N \xrightarrow{\tau} M'|N'} \quad a \in \mathcal{L}$$

$$\text{System Restriction} \qquad \frac{M \xrightarrow{a@l} M'}{M \setminus v \xrightarrow{a@l} M' \setminus v} \quad a \in \mathcal{L},\ a \notin v \qquad \frac{M \xrightarrow{\tau} M'}{M \setminus v \xrightarrow{\tau} M' \setminus v}$$

Figure 6: The operational rules for MSPA systems

leftmost l_1 the process moves from the initial site s_1 to another site s_2. We can write this fact through the new mobility primitive *goto s* as follows: $E_5 = l_1.goto\ s_2.h.\bar{l}_2.\mathbf{0} + l_1.(\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0})$. If we execute E_5 in location s_1 we obtain the following execution paths:

$$\begin{aligned} s_1[\![E_5]\!] &\xrightarrow{l_1@s_1} s_1[\![goto\ s_2.h.\bar{l}_2.\mathbf{0}]\!] \xrightarrow{goto\ s_2} s_2[\![h.\bar{l}_2.\mathbf{0}]\!] \xrightarrow{h@s_2} s_2[\![\bar{l}_2.\mathbf{0}]\!] \xrightarrow{\bar{l}_2@s_2} s_2[\![\mathbf{0}]\!] \\ &\xrightarrow{l_1@s_1} s_1[\![\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0}]\!] \xrightarrow{\tau} s_1[\![\mathbf{0}]\!] \\ &\qquad\qquad\qquad\qquad\qquad\quad \xrightarrow{\tau} s_1[\![\bar{l}_2.\mathbf{0}]\!] \xrightarrow{\bar{l}_2@s_1} s_1[\![\mathbf{0}]\!] \end{aligned}$$

Notice that, after *goto* s_2 is performed, the execution proceeds in location s_2 and this fact is made observable by the $@s_2$ suffix in the arrow labels. Notice also that $h.\bar{l}_2.\mathbf{0}$ is executed on site s_2, while the (masquerading) right branch $(\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0})$ is executed in the initial site s_1. As a consequence, a low level user on site s_2, by observing the execution of $\bar{l}_2$, will precisely know if h has been executed or not. □

This example shows that good processes should never move inside their *critical sections*, i.e., when they are executing code that, for security reasons, is bound to other code.

The way we deal with mobility is a bit simplified with respect to DPI: in MSPA, when a process E performs a move action *goto l*, then E as a whole moves to the new location l, even if E is composed of different parallel components; in DPI, instead, it is the single sequential component that moves to the new location. For example, the MSPA system $k[\![goto\ l.E' \mid b.\mathbf{0}]\!]$ moves to $l[\![E' \mid b.\mathbf{0}]\!]$, while in DPI, the same system moves to $l[\![E']\!] \mid k[\![b.\mathbf{0}]\!]$. This does not mean that a *goto* action moves all the processes in the starting location to the new locations (which would be quite difficult to implement). For example, $k[\![goto\ l.E']\!] \mid k[\![b.\mathbf{0}]\!]$ behaves like the DPI process above and moves to $l[\![E']\!] \mid k[\![b.\mathbf{0}]\!]$. Our view of mobility seems to better fit the idea of guaranteeing security even when the environment is dynamically reconfigured at runtime, which is equivalent to say that the process (as a whole) moves into a different environment. Of course, it would be interesting to also explore what happens when the full DPI mobility is taken into account. We leave this aspect as a future work.

We extend the notion of weak-bisimulation to MSPA systems as expected, and we denote it by $\approx_{\mathcal{M}}$.

Definition 4.10 (Weak Bisimulation on MSPA Systems). A binary relation $\mathcal{S}$ over MSPA systems is a *weak bisimulation* if $(M,N) \in \mathcal{S}$ implies, for all $\alpha \in Act_{\mathcal{M}}$,

- whenever $M \xrightarrow{\alpha} M'$, then there exists N' such that $N \stackrel{\hat{\alpha}}{\Longrightarrow} N'$ and $(M',N') \in \mathcal{S}$;
- whenever $N \xrightarrow{\alpha} N'$, then there exists M' such that $M \stackrel{\hat{\alpha}}{\Longrightarrow} M'$ and $(M',N') \in \mathcal{S}$.

Two MSPA systems M and N are *weakly bisimilar*, denoted by $M \approx_{\mathcal{M}} N$, if there exists a weak bisimulation $\mathcal{S}$ containing the pair (M,N).

The notion of *BNDC* can be naturally adapted to distributed systems as follows. First, the set of high level actions H is now defined as the set of all actions of the form $a@l$ where a is high. Since, in practice, it is very difficult to completely hide a migration from one host to another, we assume that *goto*'s are visible, i.e. low, actions. As a consequence, we consider malicious high level processes that do not move and are distributed on the sites, adhering to the very general form $l_1[\![\Pi_1]\!] \mid \dots \mid l_n[\![\Pi_n]\!]$ with $\Pi_1,\dots,\Pi_n \in \mathcal{E}_H$. We denote by $\mathcal{E}_H^{\mathcal{M}}$ the set of those high level systems. Processes that we analyse may move from site to site finding themselves under the attack of different, dynamically changing, attackers. The notion of *Mobility BNDC* (*M_BNDC*, for short) is defined as follows.

Definition 4.11 (*M_BNDC*)**.** Let E be a MSPA process. E is *M_BNDC* if for all $l \in Loc$ and for all $M \in \mathcal{E}_H^{\mathcal{M}}$, $(l[\![E]\!] \mid M) \setminus H \approx_{\mathcal{M}} l[\![E]\!] \setminus H$.

We can show that *BNDC* does not imply *M_BNDC*. Intuitively, if migration happens in particular "unstable states", i.e., states whose behaviour is influenced by high level activity, the sudden change of high level behaviour caused by migration could generate some (even subtle) difference in the low level behaviour. This is not captured by *BNDC* as it fixes a unique high level attacker in advance, which is not affected by migrations. The following example illustrates a process which is *BNDC* but not *M_BNDC*.

Example 4.12. Consider the process $E_6 = l_1.(h.l_2.\mathbf{0} + goto\ s_2.h.l_3.\mathbf{0}) + l_1.(\tau.(\tau.l_2.\mathbf{0} + goto\ s_2.l_3.\mathbf{0}) + \tau.goto\ s_2.\mathbf{0} + goto\ s_2.(\tau.\mathbf{0} + \tau.l_3.\mathbf{0}))$ depicted in Figure 7, where h is the only high level action. We can prove that E_6 is *BNDC* (see Appendix B) but

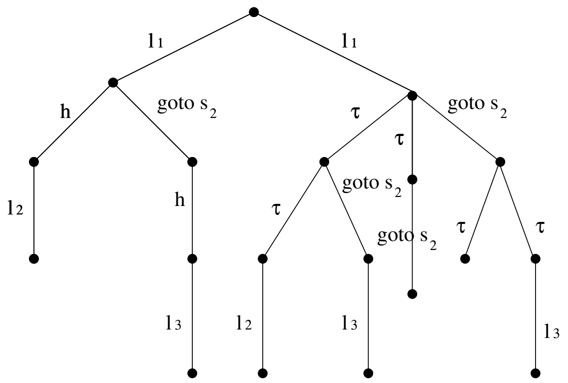

Figure 7: The process E_6

not *M_BNDC*. In fact, by considering the distributed high level malicious system $M = s_1[\![\bar{h}.\mathbf{0}]\!] \mid s_2[\![\mathbf{0}]\!]$, we obtain that $(s_1[\![E_6]\!] \mid M) \setminus H \not\approx_{\mathcal{M}} s_1[\![E_6]\!] \setminus H$. Systems $(s_1[\![E_6]\!] \mid M) \setminus H$ and $s_1[\![E_6]\!] \setminus H$ are depicted in Figure 8. Notice that the left-most $l_1 @ s_1$ transition of the former system moves it into a state where it is possible to perform either a $l_2 @ s_1$ action or a *goto* s_2 one. The latter system cannot simulate this $l_1 @ s_1$ transition by reaching a state where it is possible to (only) choose between executing either $l_2 @ s_1$ or *goto* s_2.

Intuitively, since *M_BNDC* considers different attackers in different sites, the high level behaviour may suddenly change after a migration causing an "unexpected" low level behaviour. This is the case for the above mentioned process $(s_1[\![E_6]\!] \mid M) \setminus H$ whose behaviour, after the left-most l_1 action, depends on the execution site: in s_1 action h is always allowed, while in s_2 it is always forbidden. This "inconsistency" is revealed to the low level observers by the presence of a state where it is only possible to choose between $l_2 @ s_1$ and *goto* s_2. □

Property *M_BNDC* is difficult to check due to the presence of a universal quantification over all the possible high level systems M. Here we show that *P_BNDC* is enough to guarantee that a MSPA process is *M_BNDC*. Indeed, *P_BNDC* requires that every execution state is secure. As a consequence, we are guaranteed that in a *P_BNDC*-secure process migration always happens in secure, stable, states.

Theorem 4.13. *$E \in$ P_BNDC implies $E \in$ M_BNDC.*

Proof. Let $E \in P\_BNDC$. Then for all E' reachable from E, $E' \in P\_BNDC$, i.e., $E' \setminus H \approx_{\setminus H} E'$. It is sufficient to prove that

$$\mathcal{S} \quad = \quad \{(l[\![E_1]\!] \setminus H, (l[\![E_2]\!] \mid M) \setminus H): \quad E_1 \setminus H \approx_{\setminus H} E_2,\ l \in Loc,\ M \in \mathcal{E}_H^{\mathcal{M}}\}$$

is a $\approx_{\mathcal{M}}$ bisimulation.

The fact that $\mathcal{S}$ is a $\approx_{\mathcal{M}}$ bisimulation follows from the following cases.

Let $((l[\![E_1]\!] \setminus H, (l[\![E_2]\!] \mid M) \setminus H) \in \mathcal{S}$.

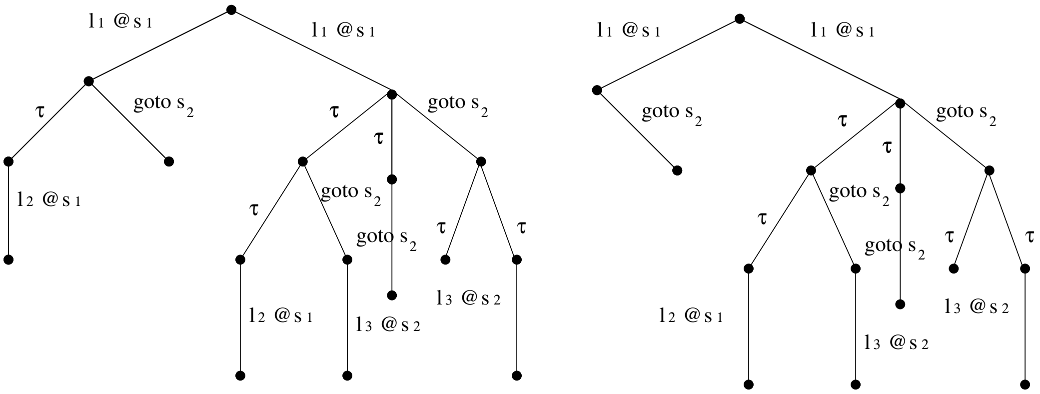

Figure 8: The processes $(s_1[\![E_6]\!] \mid M)\setminus H$ and $s_1[\![E_6]\!]\setminus H$

Case 1. $l[\![E_1]\!]\setminus H \xrightarrow{a@l} l[\![E_1']\!]\setminus H$ where $E_1 \xrightarrow{a} E_1'$ and $a \in L$. Thus, $E_1 \setminus H \xrightarrow{a} E_1' \setminus H$. By the hypothesis that $E_1 \setminus H \approx_{\setminus H} E_2$, there exists E_2' such that $E_2 \xRightarrow{\hat{a}} E_2'$ and $E_1' \setminus H \approx_{\setminus H} E_2'$. Therefore it holds $(l[\![E_2]\!] \mid M)\setminus H \xRightarrow{\widehat{a@l}} (l[\![E_2']\!] \mid M)\setminus H$ and hence, by definition of $\mathcal{S}$, $(l[\![E_1']\!]\setminus H, (l[\![E_2']\!] \mid M)\setminus H) \in \mathcal{S}$.

Case 2. $l[\![E_1]\!]\setminus H \xrightarrow{\tau} l[\![E_1']\!]\setminus H$ where $E_1 \xrightarrow{\tau} E_1'$. Thus, $E_1 \setminus H \xrightarrow{\tau} E_1' \setminus H$. By the hypothesis that $E_1 \setminus H \approx_{\setminus H} E_2$, there exists E_2' such that $E_2 \xRightarrow{\hat{\tau}} E_2'$ and $E_1' \setminus H \approx_{\setminus H} E_2'$. Thus $(l[\![E_2]\!] \mid M)\setminus H \xRightarrow{\hat{\tau}} (l[\![E_2']\!] \mid M)\setminus H$ and hence, by definition of $\mathcal{S}$, $(l[\![E_1']\!]\setminus H, (l[\![E_2']\!] \mid M)\setminus H) \in \mathcal{S}$.

Case 3. $l[\![E_1]\!]\setminus H \xrightarrow{goto\ k} k[\![E_1']\!]\setminus H$ where $E_1 \xrightarrow{goto\ k} E_1'$. Thus, $E_1 \setminus H \xrightarrow{goto\ k} E_1' \setminus H$. By the hypothesis that $E_1 \setminus H \approx_{\setminus H} E_2$, there exists E_2' such that $E_2 \xRightarrow{\widehat{goto\ k}} E_2'$ and $E_1' \setminus H \approx_{\setminus H} E_2'$. Thus $(l[\![E_2]\!] \mid M)\setminus H \xRightarrow{\widehat{goto\ k}} (l[\![E_2']\!] \mid M)\setminus H$ and hence, by definition of $\mathcal{S}$, $(l[\![E_1']\!]\setminus H, (l[\![E_2']\!] \mid M)\setminus H) \in \mathcal{S}$.

Case 4. $(l[\![E_2]\!] \mid M)\setminus H \xrightarrow{a@l} (l[\![E_2']\!] \mid M)\setminus H$ where $E_2 \xrightarrow{a} E_2'$ and $a \in L$. By the hypothesis that $E_1 \setminus H \approx_{\setminus H} E_2$, there exists E_1' such that $E_1 \setminus H \xRightarrow{\hat{a}} E_1' \setminus H$. Thus $l[\![E_1]\!]\setminus H \xRightarrow{\widehat{a@l}} l[\![E_1']\!]\setminus H$ and, by definition of $\mathcal{S}$, $(l[\![E_1']\!]\setminus H, (l[\![E_2']\!] \mid M)\setminus H) \in \mathcal{S}$.

Case 5. $(l[\![E_2]\!] \mid M)\setminus H \xrightarrow{\tau} (l[\![E_2']\!] \mid M)\setminus H$ where $E_2 \xrightarrow{\tau} E_2'$. By the hypothesis that $E_1 \setminus H \approx_{\setminus H} E_2$, there exists E_1' such that $E_1 \setminus H \xRightarrow{\hat{\tau}} E_1' \setminus H$. Thus $l[\![E_1]\!]\setminus H \xRightarrow{\hat{\tau}} l[\![E_1']\!]\setminus H$ and hence, by definition of $\mathcal{S}$, $(l[\![E_1']\!]\setminus H, (l[\![E_2']\!] \mid M)\setminus H) \in \mathcal{S}$.

Case 6. $(l[\![E_2]\!] \mid M)\setminus H \xrightarrow{\tau} (l[\![E_2]\!] \mid M')\setminus H$ where $M \xrightarrow{\tau} M'$. In this case, by definition of $\mathcal{S}$, we immediately obtain $(l[\![E_1]\!]\setminus H, (l[\![E_2]\!] \mid M')\setminus H) \in \mathcal{S}$.

Case 7. $(l[\![E_2]\!] \mid M)\setminus H \xrightarrow{\tau} (l[\![E_2']\!] \mid M')\setminus H$ where $E_2 \xrightarrow{a@l} E_2'$, $M \xrightarrow{\bar{a}@l} M'$ and $a \in H$. By the hypothesis that $E_1 \setminus H \approx_{\setminus H} E_2$, there exists E_1' such that $E_1 \setminus H \xRightarrow{\hat{\tau}} E_1' \setminus H$. Thus $l[\![E_1]\!]\setminus H \xRightarrow{\hat{\tau}} l[\![E_1']\!]\setminus H$ and hence, by definition of $\mathcal{S}$, $(l[\![E_1']\!]\setminus H, (l[\![E_2']\!] \mid M)\setminus H) \in \mathcal{S}$.

Case 8. $(l[\![E_2]\!] \mid M) \setminus H \xrightarrow{goto\ k} (k[\![E'_2]\!] \mid M) \setminus H$ where $E_2 \xrightarrow{goto\ k} E'_2$. By the hypothesis that $E_1 \setminus H \approx_{\setminus H} E_2$, there exists E'_1 such that $E_1 \setminus H \stackrel{\hat{goto\ k}}{\Longrightarrow} E'_1 \setminus H$. Thus $l[\![E_1]\!] \setminus H \stackrel{\hat{goto\ k}}{\Longrightarrow} k[\![E'_1]\!] \setminus H$ and, by definition of $\mathcal{S}$, $(k[\![E'_1]\!] \setminus H, (k[\![E'_2]\!] \mid M) \setminus H) \in \mathcal{S}$. $\square$

We show that property *P_BNDC* is strictly stronger than *M_BNDC*.

Example 4.14. Consider again the process $E_3 = l_1.h.\bar{l}_2.\mathbf{0} + l_1.(\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0})$ of Examples 2.5 and 3.3 and depicted in Figure 3. This process is *M_BNDC* (as shown in Appendix C) but not *P_BNDC* (as shown in Example 3.3). $\square$

5 An Example with Mobility

In this section, we give a non trivial example of a *P_BNDC* process. Our first aim is to give evidence that the proposed property is not too restrictive and can be used to validate interesting system specifications. Then, we show that *P_BNDC* is also useful for reasoning about mobile code. In particular we add migration primitives in different places and we study how the security of the resulting mobile process is affected.

The process we consider is inspired by the Access Monitor of [16, 17] and represents a very simple implementation of multilevel security over two binary memory cells (called Objects), a high and a low level one. For the sake of readability we first express the process using a value-passing extension of SPA. Then, we show how this specification translates to the basic SPA calculus.

$$\begin{array}{rcl} Agent & \stackrel{\text{def}}{=} & Object_h(0) \mid Object_l(0) \\ Object_h(x) & \stackrel{\text{def}}{=} & \overline{r_{hh}}(x).Object_h(x) + w_{hh}(y).Object_h(y) + w_{lh}(y).Object_h(y) \\ Object_l(x) & \stackrel{\text{def}}{=} & \overline{r_{hl}}(x).Object_l(x) + \overline{r_{ll}}(x).Object_l(x) + w_{ll}(y).Object_l(y) \end{array}$$

Process *Agent* is the parallel composition of the two cells $Object_h(0)$ and $Object_l(0)$ which initially contain value 0. Cell $Object_h(x)$ represents a high level cell which can communicate its value only to high level users through the output action $\overline{r_{hh}}$ ("r" stands for read) and can be updated by both high and low level users through the two inputs $w_{hh}(y)$ and $w_{lh}(y)$, respectively ("w" stands for write). Notice that the first subscript indicates the level of the user, while the second one represents the level of the accessed object. The low level cell $Object_l(x)$ communicates its value to both high and low level users through actions $\overline{r_{hl}}(x)$ and $\overline{r_{ll}}(x)$ and can only be updated by low level users via $w_{ll}(y)$. This process implements the *no-write-down/no-read-up* rules of [3], since no high level user can write down to the low level cell and no low level user can read from the high level cell.

The way value-passing SPA is translated into SPA is fully described in [16, 17]. The idea is to have a different action and a different SPA process for each possible value. We assume that the two memory cells described above only contain binary values.

$$
\begin{array}{rcl}
\mathit{Agent} & \stackrel{\mathrm{def}}{=} & \mathit{Object}_h\_0 \mid \mathit{Object}_l\_0 \\
\mathit{Object}_h\_0 & \stackrel{\mathrm{def}}{=} & \overline{r_{hh}\_0}.\mathit{Object}_h\_0 \\
 & + & w_{hh}\_0.\mathit{Object}_h\_0 + w_{hh}\_1.\mathit{Object}_h\_1 \\
 & + & w_{lh}\_0.\mathit{Object}_h\_0 + w_{lh}\_1.\mathit{Object}_h\_1 \\
\mathit{Object}_h\_1 & \stackrel{\mathrm{def}}{=} & \overline{r_{hh}\_1}.\mathit{Object}_h\_1 \\
 & + & w_{hh}\_0.\mathit{Object}_h\_0 + w_{hh}\_1.\mathit{Object}_h\_1 \\
 & + & w_{lh}\_0.\mathit{Object}_h\_0 + w_{lh}\_1.\mathit{Object}_h\_1 \\
\mathit{Object}_l\_0 & \stackrel{\mathrm{def}}{=} & \overline{r_{hl}\_0}.\mathit{Object}_l\_0 \\
 & + & \overline{r_{ll}\_0}.\mathit{Object}_l\_0 \\
 & + & w_{ll}\_0.\mathit{Object}_l\_0 + w_{ll}\_1.\mathit{Object}_l\_1 \\
\mathit{Object}_l\_1 & \stackrel{\mathrm{def}}{=} & \overline{r_{hl}\_1}.\mathit{Object}_l\_1 \\
 & + & \overline{r_{ll}\_1}.\mathit{Object}_l\_1 \\
 & + & w_{ll}\_0.\mathit{Object}_l\_0 + w_{ll}\_1.\mathit{Object}_l\_1
\end{array}
$$

Table 1: The simple multilevel process expressed in SPA.

Thus, for example, the process $\mathit{Object}_h(x)$ is translated to the pair Object_h_0,Object_h_1 and the corresponding action $\overline{r_{hh}}(x)$ is translated to $\overline{r_{hh}\_0}$ and $\overline{r_{hh}\_1}$, accordingly. The full translation of the three value-passing processes is reported in Table 1, in which we assume that $\{r_{hh}\_0, w_{hh}\_0, r_{hh}\_1, w_{hh}\_1, r_{hl}\_0, w_{hl}\_0, r_{hl}\_1, w_{hl}\_1\} \subseteq H$ and all the other actions are in L.

Using the CoPS tool [44], we can check that process *Agent* is *P_BNDC*. It is also possible to check that if either *read-up* or *write-down* is enabled (by suitably modifying the two cells) the process is not *P_BNDC*, as expected. For example, adding a read-up access correspond to adding a line $\overline{r_{lh}\_0}.\mathit{Object}_h\_0$ in process Object_h_0 and a line $\overline{r_{lh}\_1}.\mathit{Object}_h\_1$ in process Object_h_1. This introduces a direct causality between high level inputs (w_{hh}_0, w_{hh}_1) and low level outputs, representing a direct information leakage. Indeed, read-up allows low level users to directly read high level values from the high level cell.

It is now interesting to see what happens if we add migration actions. We first observe that, in a secure process, the ability of migrating should not depend on the high level state of the process. To see this we let the process migrate only when the high level cell contains value 1. This can be achieved by adding, in the specification of Object_h_1, the line $goto\ s.\mathit{Object}_h\_1$. We call this process Agent^m_1. Recall that, according to the MSPA semantics, a *goto* action migrates the process Agent^m_1 as a whole. Thus adding a goto in the high level cell also affects the low level one.

It can be easily seen that Agent^m_1 is not *P_BNDC*. More specifically, we show that Agent^m_1 is not *M_BNDC*. Notice that *goto s* is observable by low level users and it clearly depends on the high level w_{hh}_1 action. Indeed, initially, the high level cell contains value 0, and a way of storing value 1 in such a cell (thus enabling the migration), is to perform the high level action w_{hh}_1. Formally, consider the high level system $M = l[\![\overline{w_{hh}\_1}.\mathbf{0}]\!]$. We have that

$$(l[\![Agent_1^m]\!] \mid M) \setminus H \not\approx_{\mathcal{M}} l[\![Agent_1^m]\!] \setminus H \ .$$

As a matter of fact,

$$(l[\![Agent_1^m]\!] \mid M) \setminus H \xrightarrow{\tau} (l[\![Object_h\_1 \mid Object_l\_0]\!] \mid l[\![\mathbf{0}]\!]) \setminus H$$

and the reached state can execute a *goto s* action. This first internal τ step cannot be simulated by $l[\![Agent_1^m]\!] \setminus H$ because the only way of for this process to reach a state where *goto s* is executable, is to perform a w_{lh}_1 action. Intuitively, the first τ move represents the high level write performed by M which is revealed by the observable *goto* action.

In order to add migration in the high level cell without compromising the security, it is sufficient to also modify the $Object_h\_0$ process by adding $goto\ s.Object_h\_0$. In this way, migration does not depend on the actual high level value. We can check that this process is *P_BNDC* using CoPS.

If we try to add migration in the low level cell, we discover that the process preserves the *P_BNDC* property even if we let it migrate depending on the low level value. This reflects the intuition that migration should not depend on the high level state but may depend on the low level one.

Another interesting experiment is to add migration just before the execution of a high level action. For example, let us modify process *Agent* by adding a *goto s* just before the $\overline{r_{hl}\_0}$ of process $Object_l\_0$, i.e., by changing the corresponding line as $goto\ s.\overline{r_{hl}\_0}.Object_l\_0$. This makes the obtained $Agent_2^m$ process non-*P_BNDC*, as the reachable process $\overline{r_{hl}\_0}.Object_l\_0$ is clearly not secure. We also show that, because of the above mentioned insecurity of a reachable state, $Agent_2^m$ in not even *M_BNDC*. Consider the high level system $M = s[\![r_{hl}\_0.\mathbf{0}]\!]$. We have that

$$(l[\![Agent_2^m]\!] \mid M) \setminus H \not\approx_{\mathcal{M}} l[\![Agent_2^m]\!] \setminus H \ .$$

As a matter of fact, we have that

$$\begin{array}{rcl} (l[\![Agent_2^m]\!] \mid M) \setminus H & \xrightarrow{goto\ s} & (s[\![Object_h\_0 \mid \overline{r_{hl}\_0}.Object_l\_0]\!] \mid M) \setminus H \\ & \xrightarrow{\tau} & (s[\![Agent_2^m]\!] \mid s[\![\mathbf{0}]\!]) \setminus H \ . \end{array}$$

Notice that the reached state represents $Agent_2^m$ running at location s with no high level activity. The only way for $l[\![Agent_2^m]\!] \setminus H$ to simulate this migration, moving to location s, is to try to perform the *goto* action. However this moves the system into the state $s[\![Object_h\_0 \mid \overline{r_{hl}\_0}.Object_l\_0]\!] \setminus H$ where no interaction is possible with the low level cell. In fact, the low level cell is in a deadlock state. This potential deadlock controlled by high level users can be exploited to construct covert channels as shown in [17].

6 Conclusion and Related Work

In this paper we have studied a security property, named *P_BNDC*, which is based on the idea of Non-Interference and is persistent, i.e., it is preserved in all the states reached during process executions. We have characterized *P_BNDC* through a local property (with no quantification either on the states or on the high contexts) which is based on a new notion of *weak bisimulation up to high level actions*, denoted by $\approx_{\setminus H}$.

This result reduces the problem of verifying *P_BNDC* to the problem of checking a weak bisimulation between two processes. In the case of finite state processes, this can be efficiently solved either through model-checking or by a strong bisimulation checker, as described in [6] and briefly explained below.

- The model-checking technique can be used as follows: one can exploit the well-known greatest fixpoint characterization of bisimulation-like relations [41] to derive modal mu-calculus formulae characterizing finite-state processes up to the equivalence relation $\approx_{\setminus H}$. A model checker can be then employed to directly verify *P_BNDC*. Indeed, if $\phi^{\approx_{\setminus H}}$ is a characteristic formula for a finite state process E up to $\approx_{\setminus H}$, then $E \in$ *P_BNDC* if and only if $E \setminus H \models \phi^{\approx_{\setminus H}}$ (see [52, 53] for more detail).

- *P_BNDC* can be also proved by following the method proposed in [52] where the verification of a process equivalence is reduced to the problem of verifying a strong bisimulation between two transformed processes. Given this transformation, the strong bisimulation test can be performed using efficient algorithms for strong bisimulation (see, e.g., [43, 28, 7, 29, 14]).

Actually, the compositional security checker (CoSeC) described in [16] may be used to automatically verify *P_BNDC* over finite state processes. This is done by checking the equivalent property *SBSNNI* and requires to verify a bisimulation relation over all the possible reachable states. As explained above, *P_BNDC* may be verified more efficiently, by simply showing that two processes are weak bisimilar up to high level actions. The recently developed tool CoPS [44] implements this new verification technique.

We have proved that *P_BNDC* is compositional with respect to the parallel operator. This is useful to check the security of a complex system bottom-up, i.e., starting from its simpler sub-components. This compositional check is implemented both in CoSeC and in CoPS and it often dramatically reduces the verification time. *P_BNDC* is not compositional with respect to the nondeterministic choice operator. To improve this aspect, in a companion paper [6] we have studied variations of *P_BNDC* based on different equivalence relations. The main result is that, by adopting the *progressing-bisimulation congruence* [40], we obtain a fully [2] compositional property.

Finally, we have shown that *P_BNDC* is a property suitable for reasoning on the security of mobile processes. To this end, we have extended SPA with mobility and we have shown that, in this extended calculus, *P_BNDC* implies an extension of the *BNDC*

[2]Except for the high level prefixing.

property to migrating processes, in which execution sites may host different high level malicious processes running locally.

Persistence is not a typical feature of Non-Interference properties. For example, many properties based on trace models, like *generalized non-inference* [36], *non inference* [42], *generalized Non-Interference* [33], *separability* [36], the *perfect security property* [57], are not persistent. An interesting exception is the variant of *BNDC* proposed by Lowe in [30], in order to obtain a property which is persistent with respect to every possible refinement. In that work, persistence is exploited to guarantee that solving the non-deterministic choice, i.e., refining the process, does not introduce new information leakages. Persistence is instead used very frequently to give sufficient conditions to Non-Interference. For example, the techniques based on type-systems, like [1, 4, 10, 25, 49, 50, 51], define sufficient static conditions which are invariant with respect to execution and imply the desired dynamic property.

Non-Interference properties have already been developed for process calculi that express mobility. For example, in [12, 8] two notions of Non-Interference are defined for Boxed Ambients [9] and Mobile Ambients [11], respectively; in [25, 24, 45], other notions of Non-Interference for π-calculus [39] are studied. All of these approaches aim at defining type systems that can be used to prove Non-Interference properties. Thus, the given proof method is sound but not complete, as there might be systems that do not type-check but are secure. On the other hand, *P_BNDC* is a sound and complete characterization of persistent Non-Interference, which is also decidable over fine-state systems. Thus, an interesting future work would be to study *P_BNDC* for the above mentioned calculi and other calculi for mobility like DPI-calculus [26]. Since, observation equivalences that take into account mobility have already been studied for the above mentioned calculi, our impression is that the most difficult task is to give a precise notion of *what* are the high and the low interfaces.

Acknowledgements.

We would like to thank the anonymous referees for their very helpful comments and suggestions.

References

[1] Martín Abadi. Secrecy by typing in security protocols. *Journal of the ACM*, 46(5):749–786, 1999.

[2] A. Aldini, M. Bravetti, and R. Gorrieri. A Process-algebraic Approach for the Analysis of Probabilistic Non-interference. *Journal of Computer Security*, 12(2):191–245, 2004.

[3] D. E. Bell and L. J. La Padula. Secure computer systems: Unified exposition and multics interpretation. Technical Report ESD-TR-75-306, MITRE MTR-2997, 1976.

[4] V. Benzaken, M. Burelle, and G. Castagna. Information flow security for XML transformations. In *Proc. of Asian Computing Science Conference (ASIAN'03)*, volume 2896 of *LNCS*, pages 33–53. Springer-Verlag, Berlin, 2003.

[5] C. Bodei, P. Degano, F. Nielson, and H. Riis Nielson. Static analysis for secrecy and non-interference in networks of processes. In Victor E. Malyshkin, editor, *Proc. of International Conference on Parallel Computing Technologies*, volume 2127 of *Lecture Notes in Computer Science*, pages 27–41. Springer-Verlag, Berlin, 2001.

[6] A. Bossi, R. Focardi, C. Piazza, and S. Rossi. Verifying Persistent Security Properties. *Computer Languages, Systems and Structures*, 30(3-4):231–258, 2004.

[7] A. Bouali and R. de Simone. Symbolic Bisimulation Minimization. In G. von Bochmann and D. K. Probst, editors, *Proc. of International Conference on Computer Aided Verification (CAV'92)*, volume 663 of *LNCS*, pages 96–108. Springer-Verlag, Berlin, 1992.

[8] C. Braghin, A. Cortesi, and R. Focardi. Information Leakage Detection in Boundary Ambients. In *Proc. of Computing: The Australasian Theory Symposium (CATS'03)*. Electronic Notes in Theoretical Computer Science, Elsevier, 2003.

[9] M. Bugliesi, G. Castagna, and S. Crafa. Boxed Ambients. In *TACS'01 Proc. of the 4th Int. Conference on Theoretical Aspects of Computer Science*, number 2215 in LNCS, pages 38–63. Springer-Verlag, 2001.

[10] M. Bugliesi, S. Crafa, M. Merro, and V. Sassone. Communication Interference in Mobile Boxed Ambients. In M. Agrawal and A. Seth, editors, *Proc. of Int. Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'02)*, volume 2556 of *LNCS*, pages 71–84. Springer-Verlag, Berlin, 2002.

[11] L. Cardelli and A. D. Gordon. Mobile Ambients. In M. Nivat, editor, *Proc. of Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 1378 of *LNCS*, pages 140–155. Springer-Verlag, 1998.

[12] S. Crafa, M. Bugliesi, and G. Castagna. Information Flow Security for Boxed Ambients. In *F-WAN: Int. Workshop on Foundations of Wide Area Networks*, number 66(3) in Electronic Notes in Theoretical Computer Science, Elsevier, 2002.

[13] A. Di Pierro, C. Hankin, and H.Wiklicky. Approximate Non-Interference. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW'02)*, pages 3–17. IEEE Computer Society Press, 2002.

[14] A. Dovier, C. Piazza, and A. Policriti. A Fast Bisimulation Algorithm. In G. Berry, H. Comon, and A. Finkel, editors, *Proc. of International Conference on Computer Aided Verification (CAV'01)*, volume 2102 of *LNCS*, pages 79–90. Springer-Verlag, Berlin, 2001.

[15] R. Focardi and R. Gorrieri. A Classification of Security Properties for Process Algebras. *Journal of Computer Security*, 3(1):5–33, 1994/1995.

[16] R. Focardi and R. Gorrieri. The Compositional Security Checker: A Tool for the Verification of Information Flow Security Properties. *IEEE Transactions on Software Engineering*, 23(9):550–571, 1997.

[17] R. Focardi and R. Gorrieri. Classification of Security Properties (Part I: Information Flow). In R. Focardi and R. Gorrieri, editors, *Foundations of Security Analysis and Design*, volume 2171 of *LNCS*. Springer-Verlag, Berlin, 2001.

[18] R. Focardi, R. Gorrieri, and F. Martinelli. Non Interference for the Analysis of Cryptographic Protocols. In U. Montanari, J.D.P. Rolim, and E. Welzl, editors, *Proc. of International Colloquium on Automata, Languages and Programming (ICALP'00)*, volume 1853 of *LNCS*, pages 744–755. Springer-Verlag, Berlin, 2000.

[19] R. Focardi, R. Gorrieri, and F. Martinelli. Real-time information flow analysis. *IEEE Journal on Selected Areas in Communications*, 21(1), January 2003.

[20] R. Focardi and S. Rossi. Information Flow Security in Dynamic Contexts. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW'02)*, pages 307–319. IEEE Computer Society Press, 2002.

[21] S. N. Foley. A Universal Theory of Information Flow. In *Proc. of the IEEE Symposium on Security and Privacy (SSP'87)*, pages 116–122. IEEE Computer Society Press, 1987.

[22] J. A. Goguen and J. Meseguer. Security Policy and Security Models. In *Proc. of the 1982 Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society Press, 1982.

[23] R. Gorrieri, E. Locatelli, and F. Martinelli. A simple language for real-time cryptographic protocol analysis. In P. Degano, editor, *Proc. of European Symposium on Programming (ESOP'03)*, volume 2618 of *LNCS*, pages 114–128. Springer-Verlag, Berlin, 2003.

[24] M. Hennessy. The Security Picalculus and Non-Interference. Technical Report CS-05-2000, University of Sussex, School of Cognitive and Computing Sciences, Brighton BN1 9QH, UK, Nov. 2000.

[25] M. Hennessy and J. Riely. Information Flow vs. Resource Access in the Asynchronous Pi-calculus. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 24(5):566–591, 2002.

[26] M. Hennessy and J. Riely. Resource Access Control in Systems of Mobile Agents. *Information and Computation*, 173:82–120, 2002.

[27] D. M. Johnson and F. J. Thayer. Security and the Composition of Machines. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW'88)*, pages 72–89. IEEE Computer Society Press, 1988.

[28] P. C. Kannellakis and S. A. Smolka. CCS Expressions, Finite State Processes, and Three Problems of Equivalence. *Information and Computation*, 86(1):43–68, 1990.

[29] D. Lee and M. Yannakakis. Online Minimization of Transition Systems. In *Proc. of the 24th ACM Symposium on Theory of Computing (STOC'92)*, pages 264–274. ACM Press, 1992.

[30] G. Lowe. Quantifying Information Flow. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW'02)*, pages 18–31. IEEE Computer Society Press, 2002.

[31] H. Mantel. Possibilistic Definitions of Security - An Assembly Kit -. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW'00)*, pages 185–199. IEEE Computer Society Press, 2000.

[32] H. Mantel. Unwinding Possibilistic Security Properties. In *Proc. of the European Symposium on Research in Computer Security (ESoRiCS'00)*, volume 2895 of *LNCS*, pages 238–254. Springer-Verlag, Berlin, 2000.

[33] D. McCullough. Specifications for Multi-Level Security and a Hook-Up Property. In *Proc. of the IEEE Symposium on Security and Privacy (SSP'87)*, pages 161–166. IEEE Computer Society Press, 1987.

[34] D. McCullough. Noninterference and the Composability of Security Properties. In *Proceedings, 1988 IEEE Symposium on Security and Privacy*, pages 177–186. IEEE Computer Society Press, April 1988.

[35] J. McLean. Security Models and Information Flow. In *Proc. of the IEEE Symposium on Security and Privacy (SSP'90)*, pages 180–187. IEEE Computer Society Press, 1990.

[36] J. McLean. A General Theory of Composition for Trace Sets Closed under Selective Interleaving Functions. In *Proc. of the IEEE Symposium on Security and Privacy (SSP'94)*, pages 79–93. IEEE Computer Society Press, 1994.

[37] J. K. Millen. Finite-State Noiseless Covert Channels. In *Proceedings of the Computer Security Foundations Workshop II*, pages 81–86. the MITRE Corporation, IEEE Computer Society Press, 1989.

[38] R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.

[39] R. Milner, J. Parrow, and J. Walker. A Calculus of Mobile Processes, I and II. *Information and Computation*, 100(1):1–40,41–77, September 1992.

[40] U. Montanari and V. Sassone. CCS Dynamic Bisimulation is Progressing. In *Proc. of the Int. Symposium on Mathematical Foundations of Computer Science (MFCS'91)*, volume 520 of *LNCS*, pages 346–356. Springer-Verlag, Berlin, 1991.

[41] M. Müller-Olm. Derivation of Characteristic Formulae. *Electronic Notes in Theoretical Computer Science, Elsevier*, 18, 1998.

[42] C. O'Halloran. A Calculus of Information Flow. In *Proc. of the European Symposium on Research in Security and Privacy (ESoRiCS'90)*, pages 180–187. AFCET, 1990.

[43] R. Paige and R. E. Tarjan. Three Partition Refinement Algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

[44] C. Piazza, E. Pivato, and S. Rossi. CoPS: Checker of Persistent Security. In *Proc. of Int. Conference on Tools and Algorithms for Construction and Analysis of Systems (TACAS'04)*, volume 2988 of *LNCS*, pages 144–152. Springer-Verlag, Berlin, 2004.

[45] F. Pottier. A Simple View of Type-Secure Information Flow in the π-Calculus. In *Proc. of the 15th IEEE Computer Security Foundations Workshop (CSFW15)*, pages 320–330. IEEE Computer Society Press, 2002.

[46] A. W. Roscoe, J. C. P. Woodcock, and L. Wulf. Non-Interference through Determinism. *Journal of Computer Security*, 4(1), 1996.

[47] P. Ryan and S. Schneider. Process Algebra and Non-Interference. *Journal of Computer Security*, 9(1/2):75–103, 2001.

[48] P. Y. A. Ryan. A CSP Formulation of Non-Interference and Unwinding. *Cipher*, pages 19–27, 1991.

[49] A. Sabelfed and D. Sands. Probabilistic Noninterference for Multi-threaded Programs. In *Proc. of the IEEE Computer Security Foundations Workshop*, pages 200–215. IEEE Computer Society Press, 2000.

[50] A. Sabelfeld and A. C. Myers. Language-Based Information-Flow Security. *IEEE Journal on Selected Areas in Communication*, 21(1):5–19, 2003.

[51] G. Smith and D. M. Volpano. Secure Information Flow in a Multi-threaded Imperative Language. In *Proc. of 25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL98)*, pages 355–364. ACM Press, 1998.

[52] B. Steffen and A. Ingòlfsdòttir. Characteristic Formulae for Processes with Divergence. *Information and Computation*, 110(1):149–163, 1994.

[53] C. Stirling. Modal and Temporal Logics for Processes. In E. Brinksma, R. Cleaveland, K. G. T. Margaria Larsen, and B. Steffen, editors, *Logics for Concurrency: Structures versus Automata*, volume 1043 of *LNCS*, pages 149–237. Springer-Verlag, Berlin, 1996.

[54] D. Sutherland. A Model of Information. In *Proc. of the 9th National Computer Security Conference*, pages 175–183, 1986.

[55] C. R. Tsai, V. D. Gligor, and C. S. Chandersekaran. On the Identification of Covert Storage Channels in Secure Systems. *IEEE Transactions on Software Engineering*, pages 569–580, June 1990.

[56] J. T. Wittbold and D. M. Johnson. Information Flow in Nondeterministic Systems. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pages 144–161. IEEE Computer Society Press, 1990.

[57] A. Zakinthinos and E. S. Lee. A General Theory of Security Properties. In *Proc. of the IEEE Symposium on Security and Privacy (SSP'97)*, pages 74–102. IEEE Computer Society Press, 1997.

A Proof that process E_3 of Example 2.5 is *BNDC*

In this appendix we prove that process $E_3 \stackrel{\text{def}}{=} l_1.h.\bar{l}_2.\mathbf{0} + l_1.(\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0})$, introduced in Example 2.5, is *BNDC*.

In order to do it, we first partition the set $\mathcal{E}_H$ of all high level processes into the following three sets:

$$
\begin{array}{lcl}
\mathcal{E}_H^{\tau,\bar{h}} & = & \{\Pi \in \mathcal{E}_H \mid \Pi \stackrel{\tau}{\Longrightarrow} \Pi' \wedge \Pi' \stackrel{\bar{h}}{\not\Longrightarrow} \text{ and } \Pi \stackrel{\bar{h}}{\Longrightarrow}\} \\
\mathcal{E}_H^{\tau} & = & \{\Pi \in \mathcal{E}_H \mid \Pi \stackrel{\hat{\tau}}{\Longrightarrow} \Pi' \wedge \Pi' \stackrel{\bar{h}}{\not\Longrightarrow} \text{ and } \Pi \notin \mathcal{E}_H^{\tau,\bar{h}}\} \\
\mathcal{E}_H^{\bar{h}} & = & \{\Pi \in \mathcal{E}_H \mid \Pi \stackrel{\bar{h}}{\Longrightarrow} \text{ and } \Pi \notin \mathcal{E}_H^{\tau,\bar{h}}\}
\end{array}
$$

Then we label the states reachable from E_3 as depicted in Figure 9 and construct the binary relation $\mathcal{S}$ as follows:

$$
\begin{array}{rcll}
\mathcal{S} & = & \{((E_3^1|\Pi)\setminus H, E_3^1\setminus H) \mid \Pi \in \mathcal{E}_H\} & \cup \quad \{((E_3^2|\Pi)\setminus H, E_3^5\setminus H) \mid \Pi \in \mathcal{E}_H^{\tau,\bar{h}}\} \\
& \cup & \{((E_3^2|\Pi)\setminus H, E_3^6\setminus H) \mid \Pi \in \mathcal{E}_H^{\bar{h}}\} & \cup \quad \{((E_3^2|\Pi)\setminus H, E_3^8\setminus H) \mid \Pi \in \mathcal{E}_H^{\tau}\} \\
& \cup & \{((E_3^3|\Pi)\setminus H, E_3^6\setminus H) \mid \Pi \in \mathcal{E}_H\} & \cup \quad \{((E_3^4|\Pi)\setminus H, E_3^7\setminus H) \mid \Pi \in \mathcal{E}_H\} \\
& \cup & \{((E_3^5|\Pi)\setminus H, E_3^5\setminus H) \mid \Pi \in \mathcal{E}_H\} & \cup \quad \{((E_3^6|\Pi)\setminus H, E_3^6\setminus H) \mid \Pi \in \mathcal{E}_H\} \\
& \cup & \{((E_3^7|\Pi)\setminus H, E_3^7\setminus H) \mid \Pi \in \mathcal{E}_H\} & \cup \quad \{((E_3^8|\Pi)\setminus H, E_3^8\setminus H) \mid \Pi \in \mathcal{E}_H\} \\
& \cup & \{((E_3^8|\Pi)\setminus H, E_3^2\setminus H) \mid \Pi \in \mathcal{E}_H\}. &
\end{array}
$$

We prove that $\mathcal{S}$ is a weak bisimulation, i.e., if $(E,F) \in \mathcal{S}$ then,

- whenever $E \stackrel{a}{\to} E'$, then there exists F' such that $F \stackrel{\hat{a}}{\Longrightarrow} F'$ and $(E',F') \in \mathcal{S}$;
- whenever $F \stackrel{a}{\to} F'$, then there exists E' such that $E \stackrel{\hat{a}}{\Longrightarrow} E'$ and $(E',F') \in \mathcal{S}$.

This follows from the following cases.

1. Consider $((E_3^1|\Pi)\setminus H, E_3^1\setminus H) \in \mathcal{S}$.
   - $(E_3^1|\Pi)\setminus H \stackrel{l_1}{\to} (E_3^2|\Pi)\setminus H$. We distinguish three cases.
     (a) Let $\Pi \in \mathcal{E}_H^{\tau,\bar{h}}$. In this case, $E_3^1\setminus H \stackrel{l_1}{\to} E_3^5\setminus H$ and $((E_3^2|\Pi)\setminus H, E_3^5\setminus H) \in \mathcal{S}$.
     (b) Let $\Pi \in \mathcal{E}_H^{\bar{h}}$. In this case, $E_3^1\setminus H \stackrel{l_1}{\Longrightarrow} E_3^6\setminus H$ and $((E_3^2|\Pi)\setminus H, E_3^6\setminus H) \in \mathcal{S}$.
     (c) Let $\Pi \in \mathcal{E}_H^{\tau}$. In this case, $E_3^1\setminus H \stackrel{l_1}{\Longrightarrow} E_3^8\setminus H$ and $((E_3^2|\Pi)\setminus H, E_3^8\setminus H) \in \mathcal{S}$.
   - $(E_3^1|\Pi)\setminus H \stackrel{l_1}{\to} (E_3^5|\Pi)\setminus H$. In this case, $E_3^1\setminus H \stackrel{l_1}{\to} E_3^5\setminus H$ and, by definition of $\mathcal{S}$, we have $((E_3^5|\Pi)\setminus H, E_3^5\setminus H) \in \mathcal{S}$.
   - $(E_3^1|\Pi)\setminus H \stackrel{\tau}{\to} (E_3^1|\Pi')\setminus H$ with $\Pi \stackrel{\tau}{\to} \Pi'$. In this case, by definition of $\mathcal{S}$, we have $((E_3^1|\Pi')\setminus H, E_3^1\setminus H) \in \mathcal{S}$.

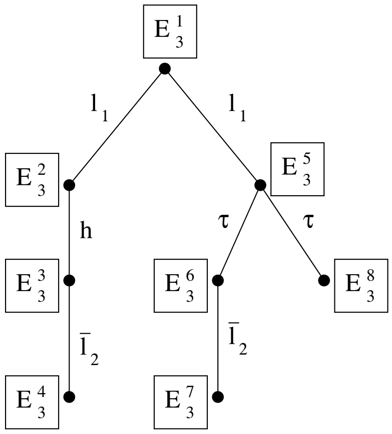

Figure 9: The process E_3 and its nodes

- $E_3^1 \setminus H \xrightarrow{l_1} E_3^2 \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H$, $(E_3^1|\Pi) \setminus H \xRightarrow{l_1} (E_3^8|\Pi) \setminus H$ and, by definition of $\mathcal{S}$, we have $((E_3^8|\Pi) \setminus H, E_3^2 \setminus H) \in \mathcal{S}$.
- $E_3^1 \setminus H \xrightarrow{l_1} E_3^5 \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H$, $(E_3^1|\Pi) \setminus H \xrightarrow{l_1} (E_3^5|\Pi) \setminus H$ and, by definition of $\mathcal{S}$, we have $((E_3^5|\Pi) \setminus H, E_3^5 \setminus H) \in \mathcal{S}$.

2. Consider $((E_3^2|\Pi) \setminus H, E_3^5 \setminus H) \in \mathcal{S}$ with $\Pi \in \mathcal{E}_H^{\tau,\bar{h}}$.

   - $(E_3^2|\Pi) \setminus H \xrightarrow{\tau} (E_3^3|\Pi') \setminus H$ where $E_3^2 \xrightarrow{h} E_3^3$ and $\Pi \xrightarrow{\bar{h}} \Pi'$. In this case, $E_3^5 \setminus H \xrightarrow{\tau} E_3^6 \setminus H$ and, by definition of $\mathcal{S}$, we have $((E_3^3|\Pi') \setminus H, E_3^6 \setminus H) \in \mathcal{S}$.
   - $(E_3^2|\Pi) \setminus H \xrightarrow{\tau} (E_3^2|\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. We distinguish three cases.
     (a) Let $\Pi' \in \mathcal{E}_H^{\tau,\bar{h}}$. In this case, we have $((E_3^2|\Pi') \setminus H, E_3^5 \setminus H) \in \mathcal{S}$.
     (b) Let $\Pi' \in \mathcal{E}_H^{\bar{h}}$. In this case, $E_3^5 \setminus H \xrightarrow{\tau} E_3^6 \setminus H$ and $((E_3^2|\Pi') \setminus H, E_3^6 \setminus H) \in \mathcal{S}$.
     (c) Let $\Pi' \in \mathcal{E}_H^{\tau}$. In this case, $E_3^5 \setminus H \xrightarrow{\tau} E_3^8 \setminus H$ and $((E_3^2|\Pi') \setminus H, E_3^8 \setminus H) \in \mathcal{S}$.
   - $E_3^5 \setminus H \xrightarrow{\tau} E_3^6 \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H^{\tau,\bar{h}}$ there exists Π' such that $\Pi \xRightarrow{h} \Pi'$. Thus, $(E_3^2|\Pi) \setminus H \xRightarrow{\tau} (E_3^3|\Pi') \setminus H$ and $((E_3^3|\Pi') \setminus H, E_3^6 \setminus H) \in \mathcal{S}$.
   - $E_3^5 \setminus H \xrightarrow{\tau} E_3^8 \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H^{\tau,\bar{h}}$ there exists Π' such that

$\Pi \stackrel{\tau}{\Longrightarrow} \Pi'$ and $\Pi' \stackrel{h}{\not\Longrightarrow}$, i.e., $\Pi' \in \mathcal{E}^{\tau}_{H}$. Thus, $(E^2_3|\Pi)\setminus H \stackrel{\tau}{\Longrightarrow} (E^2_3|\Pi')\setminus H$ and, by definition of $\mathcal{S}$, $((E^2_3|\Pi')\setminus H, E^8_3\setminus H) \in \mathcal{S}$.

3. Consider $((E^2_3|\Pi)\setminus H, E^6_3\setminus H) \in \mathcal{S}$ with $\Pi \in \mathcal{E}^{\bar{h}}_{H}$.

   - $(E^2_3|\Pi)\setminus H \xrightarrow{\tau} (E^3_3|\Pi')\setminus H$ where $E^2_3 \xrightarrow{h} E^3_3$ and $\Pi \xrightarrow{\bar{h}} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E^3_3|\Pi')\setminus H, E^6_3\setminus H) \in \mathcal{S}$.
   - $(E^2_3|\Pi)\setminus H \xrightarrow{\tau} (E^2_3|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, also $\Pi' \in \mathcal{E}^{\bar{h}}_{H}$ and by definition of $\mathcal{S}$, we have $((E^2_3|\Pi')\setminus H, E^6_3\setminus H) \in \mathcal{S}$.
   - $E^6_3\setminus H \xrightarrow{\bar{l}_2} E^7_3\setminus H$. In this case, for all $\Pi \in \mathcal{E}^{\bar{h}}_{H}$ there exists Π' such that $\Pi \stackrel{h}{\Longrightarrow} \Pi'$. Thus, $(E^2_3|\Pi)\setminus H \stackrel{\tau}{\Longrightarrow} (E^3_3|\Pi')\setminus H \xrightarrow{\bar{l}_2} (E^4_3|\Pi')\setminus H$ and, by definition of $\mathcal{S}$, $((E^4_3|\Pi')\setminus H, E^7_3\setminus H) \in \mathcal{S}$.

4. Consider $((E^2_3|\Pi)\setminus H, E^5_3\setminus H) \in \mathcal{S}$ with $\Pi \in \mathcal{E}^{\tau}_{H}$.

   - $(E^2_3|\Pi)\setminus H \xrightarrow{\tau} (E^2_3|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, also $\Pi' \in \mathcal{E}^{\tau}_{H}$ and, by definition of $\mathcal{S}$, we have $((E^2_3|\Pi')\setminus H, E^8_3\setminus H) \in \mathcal{S}$.

5. Consider $((E^3_3|\Pi)\setminus H, E^6_3\setminus H) \in \mathcal{S}$.

   - $(E^3_3|\Pi)\setminus H \xrightarrow{\bar{l}_2} (E^4_3|\Pi)\setminus H$. In this case, $E^6_3\setminus H \xrightarrow{\bar{l}_2} E^7_3\setminus H$ and, by definition of $\mathcal{S}$, $((E^4_3|\Pi)\setminus H, E^7_3\setminus H) \in \mathcal{S}$.
   - $(E^3_3|\Pi)\setminus H \xrightarrow{\tau} (E^3_3|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E^3_3|\Pi')\setminus H, E^6_3\setminus H) \in \mathcal{S}$.
   - $E^6_3\setminus H \xrightarrow{\bar{l}_2} E^7_3\setminus H$. In this case, $(E^3_3|\Pi)\setminus H \xrightarrow{\bar{l}_2} (E^4_3|\Pi)\setminus H$ and, by definition of $\mathcal{S}$, $((E^4_3|\Pi)\setminus H, E^7_3\setminus H) \in \mathcal{S}$.

6. Consider $((E^4_3|\Pi)\setminus H, E^7_3\setminus H) \in \mathcal{S}$.

   - $(E^4_3|\Pi)\setminus H \xrightarrow{\tau} (E^4_3|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E^4_3|\Pi')\setminus H, E^7_3\setminus H) \in \mathcal{S}$.

7. Consider $((E^5_3|\Pi)\setminus H, E^5_3\setminus H) \in \mathcal{S}$.

   - $(E^5_3|\Pi)\setminus H \xrightarrow{\tau} (E^6_3|\Pi)\setminus H$. In this case, $E^5_3\setminus H \xrightarrow{\tau} E^6_3\setminus H$ and, by definition of $\mathcal{S}$, $((E^6_3|\Pi)\setminus H, E^6_3\setminus H) \in \mathcal{S}$.
   - $(E^5_3|\Pi)\setminus H \xrightarrow{\tau} (E^8_3|\Pi)\setminus H$. In this case, $E^5_3\setminus H \xrightarrow{\tau} E^8_3\setminus H$ and, by definition of $\mathcal{S}$, $((E^8_3|\Pi)\setminus H, E^8_3\setminus H) \in \mathcal{S}$.
   - $(E^5_3|\Pi)\setminus H \xrightarrow{\tau} (E^5_3|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E^5_3|\Pi')\setminus H, E^5_3\setminus H) \in \mathcal{S}$.
   - $E^5_3\setminus H \xrightarrow{\tau} E^6_3\setminus H$. In this case, for all $\Pi \in \mathcal{E}_{H}$, $(E^5_3|\Pi)\setminus H \xrightarrow{\tau} (E^6_3|\Pi)\setminus H$ and, by definition of $\mathcal{S}$, $((E^6_3|\Pi)\setminus H, E^6_3\setminus H) \in \mathcal{S}$.

- $E_3^5 \setminus H \xrightarrow{\tau} E_3^8 \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H$, $(E_3^5|\Pi) \setminus H \xrightarrow{\tau} (E_3^8|\Pi) \setminus H$ and, by definition of $\mathcal{S}$, $((E_3^8|\Pi) \setminus H, E_3^8 \setminus H) \in \mathcal{S}$.

8. Consider $((E_3^6|\Pi) \setminus H, E_3^6 \setminus H) \in \mathcal{S}$.

   - $(E_3^6|\Pi) \setminus H \xrightarrow{\bar{l}_2} (E_3^7|\Pi) \setminus H$. In this case, $E_3^6 \setminus H \xrightarrow{\bar{l}_2} E_3^7 \setminus H$ and, by definition of $\mathcal{S}$, $((E_3^7|\Pi) \setminus H, E_3^7 \setminus H) \in \mathcal{S}$.
   - $(E_3^6|\Pi) \setminus H \xrightarrow{\tau} (E_3^6|\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_3^6|\Pi') \setminus H, E_3^6 \setminus H) \in \mathcal{S}$.
   - $E_3^6 \setminus H \xrightarrow{\bar{l}_2} E_3^7 \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H$, $(E_3^6|\Pi) \setminus H \xrightarrow{\bar{l}_2} (E_3^7|\Pi) \setminus H$ and, by definition of $\mathcal{S}$, $((E_3^7|\Pi) \setminus H, E_3^7 \setminus H) \in \mathcal{S}$.

9. Consider $((E_3^7|\Pi) \setminus H, E_3^7 \setminus H) \in \mathcal{S}$.

   - $(E_3^7|\Pi) \setminus H \xrightarrow{\tau} (E_3^7|\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_3^7|\Pi') \setminus H, E_3^7 \setminus H) \in \mathcal{S}$.

10. Consider $((E_3^8|\Pi) \setminus H, E_3^8 \setminus H) \in \mathcal{S}$.

    - $(E_3^8|\Pi) \setminus H \xrightarrow{\tau} (E_3^8|\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_3^8|\Pi') \setminus H, E_3^8 \setminus H) \in \mathcal{S}$.

11. Consider $((E_3^8|\Pi) \setminus H, E_3^2 \setminus H) \in \mathcal{S}$.

    - $(E_3^8|\Pi) \setminus H \xrightarrow{\tau} (E_3^8|\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_3^8|\Pi') \setminus H, E_3^2 \setminus H) \in \mathcal{S}$.

The property that E_3 is *BNDC* follows from the fact that $E_3 \equiv E_3^1$ and $\mathcal{S}$ contains all pairs of the form $((E_3|\Pi) \setminus H, E_3 \setminus H)$ with $\Pi \in \mathcal{E}_H$.

B Proof that process E_6 of Example 4.12 is *BNDC*

In this appendix we show that process $E_6 = l_1.(h.l_2.\mathbf{0} \ + \ goto\ s_2.h.l_3.\mathbf{0}) + l_1.(\tau.(\tau.l_2.\mathbf{0} + goto\ s_2.l_3.\mathbf{0}) + \tau.goto\ s_2.\mathbf{0} + goto\ s_2.(\tau.\mathbf{0} + \tau.l_3.\mathbf{0}))$ of Example 4.12 is *BNDC*.

Again, we consider the partition of the set $\mathcal{E}_H$ of all high level processes into the three sets $\mathcal{E}_H^{\tau,\bar{h}}$, $\mathcal{E}_H^{\tau}$, $\mathcal{E}_H^{\bar{h}}$ defined in Appendix A.

Then we label the states reachable from E_6 as depicted in Figure 10 and construct the binary relation $\mathcal{S}$ as follows:

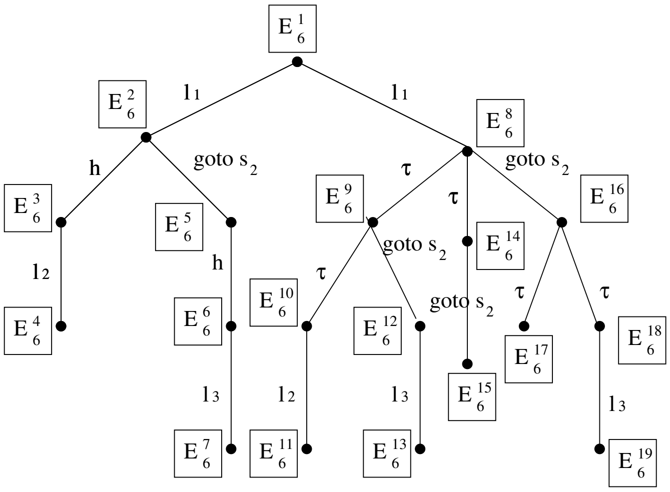

Figure 10: The process E_6

$$
\begin{aligned}
\mathcal{S} \;=\; & \{((E_6^1|\Pi)\setminus H, E_6^1\setminus H) \mid \Pi\in\mathcal{E}_H\} \cup \{((E_6^2|\Pi)\setminus H, E_6^8\setminus H) \mid \Pi\in\mathcal{E}_H^{\tau,\bar{h}}\} \\
\cup\; & \{((E_6^2|\Pi)\setminus H, E_6^{14}\setminus H) \mid \Pi\in\mathcal{E}_H^{\tau}\} \cup \{((E_6^2|\Pi)\setminus H, E_6^9\setminus H) \mid \Pi\in\mathcal{E}_H^{\bar{h}}\} \\
\cup\; & \{((E_6^3|\Pi)\setminus H, E_6^{10}\setminus H) \mid \Pi\in\mathcal{E}_H\} \cup \{((E_6^4|\Pi)\setminus H, E_6^{11}\setminus H) \mid \Pi\in\mathcal{E}_H\} \\
\cup\; & \{((E_6^5|\Pi)\setminus H, E_6^{16}\setminus H) \mid \Pi\in\mathcal{E}_H^{\tau,\bar{h}}\} \cup \{((E_6^5|\Pi)\setminus H, E_6^{17}\setminus H) \mid \Pi\in\mathcal{E}_H^{\tau}\} \\
\cup\; & \{((E_6^5|\Pi)\setminus H, E_6^{15}\setminus H) \mid \Pi\in\mathcal{E}_H^{\tau}\} \cup \{((E_6^5|\Pi)\setminus H, E_6^{18}\setminus H) \mid \Pi\in\mathcal{E}_H^{\bar{h}}\} \\
\cup\; & \{((E_6^5|\Pi)\setminus H, E_6^{12}\setminus H) \mid \Pi\in\mathcal{E}_H^{\bar{h}}\} \cup \{((E_6^6|\Pi)\setminus H, E_6^{12}\setminus H) \mid \Pi\in\mathcal{E}_H\} \\
\cup\; & \{((E_6^6|\Pi)\setminus H, E_6^{18}\setminus H) \mid \Pi\in\mathcal{E}_H\} \cup \{((E_6^7|\Pi)\setminus H, E_6^{13}\setminus H) \mid \Pi\in\mathcal{E}_H\} \\
\cup\; & \{((E_6^7|\Pi)\setminus H, E_6^{19}\setminus H) \mid \Pi\in\mathcal{E}_H\} \cup \{((E_6^{14}|\Pi)\setminus H, E_6^2\setminus H) \mid \Pi\in\mathcal{E}_H\} \\
\cup\; & \{((E_6^{15}|\Pi)\setminus H, E_6^5\setminus H) \mid \Pi\in\mathcal{E}_H\} \\
\cup\; & \{((E_6^i|\Pi)\setminus H, E_6^i\setminus H) \mid \Pi\in\mathcal{E}_H \text{ and } i\in[8..19]\}
\end{aligned}
$$

In order to prove that $\mathcal{S}$ is a weak bisimulation, we need to consider the following cases:

1. Consider $((E_6^1|\Pi)\setminus H, E_6^1\setminus H)\in\mathcal{S}$.

   - $(E_6^1|\Pi)\setminus H \xrightarrow{l_1} (E_6^2|\Pi)\setminus H$. We distinguish three cases.

     (a) If $\Pi\in\mathcal{E}_H^{\tau,\bar{h}}$, then $E_6^1\setminus H \xrightarrow{l_1} E_6^8\setminus H$ and $((E_6^2|\Pi)\setminus H, E_6^8\setminus H)\in\mathcal{S}$.

     (b) If $\Pi\in\mathcal{E}_H^{\tau}$, then $E_6^1\setminus H \stackrel{l_1}{\Longrightarrow} E_6^{14}\setminus H$ and $((E_6^2|\Pi)\setminus H, E_6^{14}\setminus H)\in\mathcal{S}$.

(c) If $\Pi \in \mathcal{E}_H^{\bar{h}}$, then $E_6^1 \setminus H \stackrel{l_1}{\Longrightarrow} E_6^9 \setminus H$ and $((E_6^2|\Pi)\setminus H, E_6^9 \setminus H) \in \mathcal{S}$.

- $(E_6^1|\Pi)\setminus H \xrightarrow{l_1} (E_6^8|\Pi)\setminus H$. In this case, $E_6^1 \setminus H \xrightarrow{l_1} E_6^8 \setminus H$ and, by definition of $\mathcal{S}$, we have $((E_6^8|\Pi)\setminus H, E_6^8 \setminus H) \in \mathcal{S}$.
- $(E_6^1|\Pi)\setminus H \xrightarrow{\tau} (E_6^1|\Pi')\setminus H$ with $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we have $((E_6^1|\Pi')\setminus H, E_6^1 \setminus H) \in \mathcal{S}$.
- $E_6^1 \setminus H \xrightarrow{l_1} E_6^2 \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H$, $(E_6^1|\Pi)\setminus H \stackrel{l_1}{\Longrightarrow} (E_6^{14}|\Pi)\setminus H$ and, by definition of $\mathcal{S}$, we have $((E_6^{14}|\Pi)\setminus H, E_6^2 \setminus H) \in \mathcal{S}$.
- $E_6^1 \setminus H \xrightarrow{l_1} E_6^8 \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H$, $(E_6^1|\Pi)\setminus H \xrightarrow{l_1} (E_6^8|\Pi)\setminus H$ and, by definition of $\mathcal{S}$, we have $((E_6^8|\Pi)\setminus H, E_6^8 \setminus H) \in \mathcal{S}$.

2. Consider $((E_6^2|\Pi)\setminus H, E_6^8 \setminus H) \in \mathcal{S}$ with $\Pi \in \mathcal{E}_H^{\tau,\bar{h}}$.

- $(E_6^2|\Pi)\setminus H \xrightarrow{\tau} (E_6^3|\Pi')\setminus H$ where $E_6^2 \xrightarrow{h} E_6^3$, $\Pi \xrightarrow{\bar{h}} \Pi'$. In this case, $E_6^8 \setminus H \stackrel{\tau}{\Longrightarrow} E_6^{10} \setminus H$ and, by definition of $\mathcal{S}$, $((E_6^3|\Pi')\setminus H, E_6^{10} \setminus H) \in \mathcal{S}$.
- $(E_6^2|\Pi)\setminus H \xrightarrow{goto\ s_2} (E_6^5|\Pi)\setminus H$ with $\Pi \in \mathcal{E}_H^{\tau,\bar{h}}$. In this case, $E_6^8 \setminus H \xrightarrow{goto\ s_2} E_6^{16} \setminus H$ and, by definition of $\mathcal{S}$, $((E_6^5|\Pi)\setminus H, E_6^{16} \setminus H) \in \mathcal{S}$.
- $(E_6^2|\Pi)\setminus H \xrightarrow{\tau} (E_6^2|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. We distinguish three cases.
  (a) If $\Pi' \in \mathcal{E}_H^{\tau,\bar{h}}$, then, by definition of $\mathcal{S}$, $((E_6^2|\Pi')\setminus H, E_6^8 \setminus H) \in \mathcal{S}$.
  (b) If $\Pi' \in \mathcal{E}_H^{\tau}$ then, $E_6^8 \setminus H \xrightarrow{\tau} E_6^{14} \setminus H$ and $((E_6^2|\Pi')\setminus H, E_6^{14} \setminus H) \in \mathcal{S}$.
  (c) If $\Pi' \in \mathcal{E}_H^{\bar{h}}$ then, $E_6^8 \setminus H \xrightarrow{\tau} E_6^9 \setminus H$ and $((E_6^2|\Pi')\setminus H, E_6^9 \setminus H) \in \mathcal{S}$.
- $E_6^8 \setminus H \xrightarrow{\tau} E_6^9 \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H^{\tau,\bar{h}}$, there exists Π' such that $\Pi \stackrel{\tau}{\Longrightarrow} \Pi'$ with $\Pi' \in \mathcal{E}_H^{\bar{h}}$. Thus, $(E_6^2|\Pi)\setminus H \Longrightarrow (E_6^2|\Pi')\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^2|\Pi')\setminus H, E_6^9 \setminus H) \in \mathcal{S}$.
- $E_6^8 \setminus H \xrightarrow{\tau} E_6^{14} \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H^{\tau,\bar{h}}$, there exists Π' such that $\Pi \stackrel{\tau}{\Longrightarrow} \Pi'$ with $\Pi' \in \mathcal{E}_H^{\tau}$. Thus, $(E_6^2|\Pi)\setminus H \stackrel{\tau}{\Longrightarrow} (E_6^2|\Pi')\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^2|\Pi')\setminus H, E_6^{14} \setminus H) \in \mathcal{S}$.
- $E_6^8 \setminus H \xrightarrow{goto\ s_2} E_6^{16} \setminus H$. In this case, for all $\Pi \in \mathcal{E}_H^{\tau,\bar{h}}$, $(E_6^2|\Pi)\setminus H \xrightarrow{goto\ s_2} (E_6^5|\Pi)\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^5|\Pi)\setminus H, E_6^{16} \setminus H) \in \mathcal{S}$.

3. Consider $((E_6^2|\Pi)\setminus H, E_6^{14} \setminus H) \in \mathcal{S}$ with $\Pi \in \mathcal{E}_H^{\tau}$.

- $(E_6^2|\Pi)\setminus H \xrightarrow{goto\ s_2} (E_6^5|\Pi)\setminus H$ with $\Pi \in \mathcal{E}_H^{\tau}$. In this case, for all $\Pi \in \mathcal{E}_H^{\tau}$, $(E_6^{14}|\Pi)\setminus H \xrightarrow{goto\ s_2} (E_6^{15}|\Pi)\setminus H$ and, by definition of $\mathcal{S}$, we have $((E_6^5|\Pi)\setminus H, E_6^{15} \setminus H) \in \mathcal{S}$.
- $(E_6^2|\Pi)\setminus H \xrightarrow{\tau} (E_6^2|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$ and $\Pi' \in \mathcal{E}_H^{\tau}$. In this case, we immediately have $((E_6^2|\Pi')\setminus H, E_6^{14} \setminus H) \in \mathcal{S}$.
- $E_6^{14} \setminus H \xrightarrow{goto\ s_2} E_6^{15} \setminus H$. In this case, $(E_6^2|\Pi)\setminus H \xrightarrow{goto\ s_2} (E_6^5|\Pi)\setminus H$ with $\Pi \in \mathcal{E}_H^{\tau}$, and, by definition of $\mathcal{S}$, $((E_6^5|\Pi)\setminus H, E_6^{15} \setminus H) \in \mathcal{S}$.

4. Consider $((E_6^2|\Pi)\setminus H, E_6^9\setminus H)\in\mathcal{S}$ with $\Pi\in\mathcal{E}_H^{\bar{h}}$.

   - $(E_6^2|\Pi)\setminus H \xrightarrow{\tau} (E_6^3|\Pi')\setminus H$ where $E_6^2 \xrightarrow{h} E_6^3$, $\Pi \xrightarrow{\bar{h}} \Pi'$. In this case, $E_6^9\setminus H \xrightarrow{\tau} E_6^{10}\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^3|\Pi')\setminus H, E_6^{10}\setminus H)\in\mathcal{S}$.
   - $(E_6^2|\Pi)\setminus H \overset{goto\ s_2}{\rightarrow} (E_6^5|\Pi)\setminus H$ with $\Pi\in\mathcal{E}_H^{\bar{h}}$. In this case, $E_6^9\setminus H \overset{goto\ s_2}{\rightarrow} E_6^{12}\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^5|\Pi)\setminus H, E_6^{12}\setminus H)\in\mathcal{S}$.
   - $(E_6^2|\Pi)\setminus H \xrightarrow{\tau} (E_6^2|\Pi')\setminus H$ where $\Pi\xrightarrow{\tau}\Pi'$ and $\Pi'\in\mathcal{E}_H^{\bar{h}}$. In this case, we immediately have $((E_6^2|\Pi')\setminus H, E_6^9\setminus H)\in\mathcal{S}$.
   - $E_6^9\setminus H \xrightarrow{\tau} E_6^{10}\setminus H$. In this case, for all $\Pi\in\mathcal{E}_H^{\bar{h}}$, $(E_6^2|\Pi)\setminus H \overset{\tau}{\Longrightarrow} (E_6^3|\Pi')\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^3|\Pi')\setminus H, E_6^{10}\setminus H)\in\mathcal{S}$.
   - $E_6^9\setminus H \overset{goto\ s_2}{\rightarrow} E_6^{12}\setminus H$. In this case, for all $\Pi\in\mathcal{E}_H^{\bar{h}}$, $(E_6^2|\Pi)\setminus H \overset{goto\ s_2}{\rightarrow} (E_6^5|\Pi)\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^5|\Pi)\setminus H, E_6^{12}\setminus H)\in\mathcal{S}$.

5. Consider $((E_6^3|\Pi)\setminus H, E_6^{10}\setminus H)\in\mathcal{S}$.

   - $(E_6^3|\Pi)\setminus H \xrightarrow{l_2} (E_6^4|\Pi)\setminus H$. In this case, $E_6^{10}\setminus H \xrightarrow{l_2} E_6^{11}\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^4|\Pi)\setminus H, E_6^{11}\setminus H)\in\mathcal{S}$.
   - $(E_6^3|\Pi)\setminus H \xrightarrow{\tau} (E_6^3|\Pi')\setminus H$ where $\Pi\xrightarrow{\tau}\Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^3|\Pi')\setminus H, E_6^{10}\setminus H)\in\mathcal{S}$.
   - $E_6^{10}\setminus H \xrightarrow{l_2} E_6^{11}\setminus H$. In this case, $(E_6^3|\Pi)\setminus H \xrightarrow{l_2} (E_6^4|\Pi)\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^4|\Pi)\setminus H, E_6^{11}\setminus H)\in\mathcal{S}$.

6. Consider $((E_6^4|\Pi)\setminus H, E_6^{11}\setminus H)\in\mathcal{S}$.

   - $(E_6^4|\Pi)\setminus H \xrightarrow{\tau} (E_6^4|\Pi')\setminus H$ where $\Pi\xrightarrow{\tau}\Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^4|\Pi')\setminus H, E_6^{11}\setminus H)\in\mathcal{S}$.

7. Consider $((E_6^5|\Pi)\setminus H, E_6^{16}\setminus H)\in\mathcal{S}$ with $\Pi\in\mathcal{E}_H^{\tau,\bar{h}}$.

   - $(E_6^5|\Pi)\setminus H \xrightarrow{\tau} (E_6^6|\Pi')\setminus H$ where $E_6^5 \xrightarrow{h} E_6^6$, $\Pi \xrightarrow{\bar{h}} \Pi'$. In this case, $E_6^{16}\setminus H \xrightarrow{\tau} E_6^{18}\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^6|\Pi')\setminus H, E_6^{18}\setminus H)\in\mathcal{S}$.
   - $(E_6^5|\Pi)\setminus H \xrightarrow{\tau} (E_6^5|\Pi')\setminus H$ where $\Pi\xrightarrow{\tau}\Pi'$. We distinguish three cases.
     (a) If $\Pi'\in\mathcal{E}_H^{\tau,\bar{h}}$, then, by definition of $\mathcal{S}$, $((E_6^5|\Pi')\setminus H, E_6^{16}\setminus H)\in\mathcal{S}$.
     (b) If $\Pi'\in\mathcal{E}_H^{\tau}$ then, $E_6^{16}\setminus H \xrightarrow{\tau} E_6^{17}\setminus H$ and $((E_6^5|\Pi')\setminus H, E_6^{17}\setminus H)\in\mathcal{S}$.
     (c) If $\Pi'\in\mathcal{E}_H^{\bar{h}}$ then, $E_6^{16}\setminus H \xrightarrow{\tau} E_6^{18}\setminus H$ and $((E_6^5|\Pi')\setminus H, E_6^{18}\setminus H)\in\mathcal{S}$.
   - $E_6^{16}\setminus H \xrightarrow{\tau} E_6^{17}\setminus H$. In this case, for all $\Pi\in\mathcal{E}_H^{\tau,\bar{h}}$, there exists Π' such that $\Pi \overset{\tau}{\Longrightarrow} \Pi'$ with $\Pi'\in\mathcal{E}_H^{\tau}$. Thus, $(E_6^5|\Pi)\setminus H \overset{\tau}{\Longrightarrow} (E_6^5|\Pi')\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^5|\Pi')\setminus H, E_6^{17}\setminus H)\in\mathcal{S}$.
   - $E_6^{16}\setminus H \xrightarrow{\tau} E_6^{18}\setminus H$. In this case, for all $\Pi\in\mathcal{E}_H^{\tau,\bar{h}}$, there exists Π' such that $\Pi \overset{\tau}{\Longrightarrow} \Pi'$ with $\Pi'\in\mathcal{E}_H^{\bar{h}}$. Thus, $(E_6^5|\Pi)\setminus H \overset{\tau}{\Longrightarrow} (E_6^5|\Pi')\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^5|\Pi')\setminus H, E_6^{18}\setminus H)\in\mathcal{S}$.

8. Consider $((E_6^5|\Pi)\setminus H, E_6^{17}\setminus H) \in \mathcal{S}$ with $\Pi \in \mathcal{E}_H^\tau$.

   - $(E_6^5|\Pi)\setminus H \xrightarrow{\tau} (E_6^5|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$ and $\Pi' \in \mathcal{E}_H^\tau$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^5|\Pi')\setminus H, E_6^{17}\setminus H) \in \mathcal{S}$.

9. Consider $((E_6^5|\Pi)\setminus H, E_6^{15}\setminus H) \in \mathcal{S}$ with $\Pi \in \mathcal{E}_H^\tau$.

   - $(E_6^5|\Pi)\setminus H \xrightarrow{\tau} (E_6^5|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$ and $\Pi' \in \mathcal{E}_H^\tau$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^5|\Pi')\setminus H, E_6^{15}\setminus H) \in \mathcal{S}$.

10. Consider $((E_6^5|\Pi)\setminus H, E_6^{18}\setminus H) \in \mathcal{S}$ with $\Pi \in \mathcal{E}_H^{\bar{h}}$.

    - $(E_6^5|\Pi)\setminus H \xrightarrow{\tau} (E_6^6|\Pi')\setminus H$ where $E_6^5 \xrightarrow{h} E_6^6$ and $\Pi \xrightarrow{\bar{h}} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^6|\Pi')\setminus H, E_6^{18}\setminus H) \in \mathcal{S}$.
    - $(E_6^5|\Pi)\setminus H \xrightarrow{\tau} (E_6^5|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$ and $\Pi' \in \mathcal{E}_H^{\bar{h}}$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^5|\Pi')\setminus H, E_6^{18}\setminus H) \in \mathcal{S}$.
    - $E_6^{18}\setminus H \xrightarrow{l_3} E_6^{19}\setminus H$. In this case, $(E_6^5|\Pi)\setminus H \xRightarrow{l_3} (E_6^7|\Pi')\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^7|\Pi')\setminus H, E_6^{19}\setminus H) \in \mathcal{S}$.

11. Consider $((E_6^5|\Pi)\setminus H, E_6^{12}\setminus H) \in \mathcal{S}$ with $\Pi \in \mathcal{E}_H^{\bar{h}}$.

    - $(E_6^5|\Pi)\setminus H \xrightarrow{\tau} (E_6^6|\Pi')\setminus H$ where $E_6^5 \xrightarrow{h} E_6^6$ and $\Pi \xrightarrow{\bar{h}} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^6|\Pi')\setminus H, E_6^{12}\setminus H) \in \mathcal{S}$.
    - $(E_6^5|\Pi)\setminus H \xrightarrow{\tau} (E_6^5|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$ and $\Pi' \in \mathcal{E}_H^{\bar{h}}$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^5|\Pi')\setminus H, E_6^{12}\setminus H) \in \mathcal{S}$.
    - $E_6^{12}\setminus H \xrightarrow{l_3} E_6^{13}\setminus H$. In this case, $(E_6^5|\Pi)\setminus H \xRightarrow{l_3} (E_6^7|\Pi')\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^7|\Pi')\setminus H, E_6^{13}\setminus H) \in \mathcal{S}$.

12. Consider $((E_6^6|\Pi)\setminus H, E_6^{12}\setminus H) \in \mathcal{S}$.

    - $(E_6^6|\Pi)\setminus H \xrightarrow{l_3} (E_6^7|\Pi)\setminus H$. In this case, $E_6^{12}\setminus H \xrightarrow{l_3} E_6^{13}\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^7|\Pi)\setminus H, E_6^{13}\setminus H) \in \mathcal{S}$.
    - $(E_6^6|\Pi)\setminus H \xrightarrow{\tau} (E_6^6|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^6|\Pi')\setminus H, E_6^{12}\setminus H) \in \mathcal{S}$.
    - $E_6^{12}\setminus H \xrightarrow{l_3} E_6^{13}\setminus H$. In this case, $(E_6^6|\Pi)\setminus H \xrightarrow{l_3} (E_6^7|\Pi)\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^7|\Pi)\setminus H, E_6^{13}\setminus H) \in \mathcal{S}$.

13. Consider $((E_6^6|\Pi)\setminus H, E_6^{18}\setminus H) \in \mathcal{S}$.

    - $(E_6^6|\Pi)\setminus H \xrightarrow{l_3} (E_6^7|\Pi)\setminus H$. In this case, $E_6^{18}\setminus H \xrightarrow{l_3} E_6^{19}\setminus H$ and, by definition of $\mathcal{S}$, $((E_6^7|\Pi)\setminus H, E_6^{19}\setminus H) \in \mathcal{S}$.
    - $(E_6^6|\Pi)\setminus H \xrightarrow{\tau} (E_6^6|\Pi')\setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^6|\Pi')\setminus H, E_6^{18}\setminus H) \in \mathcal{S}$.

- $E_6^{18} \setminus H \xrightarrow{l_3} E_6^{19} \setminus H$. In this case, $(E_6^6|\Pi) \setminus H \xrightarrow{l_3} (E_6^7|\Pi) \setminus H$ and, by definition of $\mathcal{S}$, $((E_6^7|\Pi) \setminus H, E_6^{19} \setminus H) \in \mathcal{S}$.

14. Consider $(E_6^7|\Pi) \setminus H, E_6^{13} \setminus H) \in \mathcal{S}$.

    - $(E_6^7|\Pi) \setminus H \xrightarrow{\tau} (E_6^7|\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^7|\Pi') \setminus H, E_6^{13} \setminus H) \in \mathcal{S}$.

15. Consider $(E_6^7|\Pi) \setminus H, E_6^{19} \setminus H) \in \mathcal{S}$.

    - $(E_6^7|\Pi) \setminus H \xrightarrow{\tau} (E_6^7|\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^7|\Pi') \setminus H, E_6^{19} \setminus H) \in \mathcal{S}$.

16. Consider $(E_6^{14}|\Pi) \setminus H, E_6^2 \setminus H) \in \mathcal{S}$.

    - $(E_6^{14}|\Pi) \setminus H \xrightarrow{goto\ s_2} (E_6^{15}|\Pi) \setminus H$. In this case, $E_6^2 \setminus H \xrightarrow{goto\ s_2} E_6^5 \setminus H$ and, by definition of $\mathcal{S}$, $((E_6^{15}|\Pi) \setminus H, E_6^5 \setminus H) \in \mathcal{S}$.
    - $(E_6^{14}|\Pi) \setminus H \xrightarrow{\tau} (E_6^{14}|\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^{14}|\Pi') \setminus H, E_6^2 \setminus H) \in \mathcal{S}$.
    - $E_6^2 \setminus H \xrightarrow{goto\ s_2} E_6^5 \setminus H$. In this case, $(E_6^{14}|\Pi) \setminus H \xrightarrow{goto\ s_2} (E_6^{15}|\Pi) \setminus H$ and, by definition of $\mathcal{S}$, $((E_6^{15}|\Pi) \setminus H, E_6^5 \setminus H) \in \mathcal{S}$.

17. Consider $(E_6^{15}|\Pi) \setminus H, E_6^5 \setminus H) \in \mathcal{S}$.

    - $(E_6^{15}|\Pi) \setminus H \xrightarrow{\tau} (E_6^{15}|\Pi') \setminus H$ where $\Pi \xrightarrow{\tau} \Pi'$. In this case, by definition of $\mathcal{S}$, we immediately have $((E_6^{15}|\Pi') \setminus H, E_6^5 \setminus H) \in \mathcal{S}$.

All the other cases are trivial. The property that E_6 is *BNDC* follows from the fact that $E_6 \equiv E_6^1$ and $\mathcal{S}$ contains all pairs of the form $((E_6|\Pi) \setminus H, E_6 \setminus H)$ with $\Pi \in \mathcal{E}_H$.

C Proof that process E_3 of Example 2.5 is *M_BNDC*

In this appendix we prove that process $E_3 \stackrel{\text{def}}{=} l_1.h.\bar{l}_2.\mathbf{0} + l_1.(\tau.\bar{l}_2.\mathbf{0} + \tau.\mathbf{0})$, introduced in Example 2.5, is *M_BNDC*.

In order to do it, given $l \in Loc$, we partition the set $\mathcal{E}_H^D$ of all high level MSPA systems into the following sets:

$$
\begin{array}{lcl}
\mathcal{E}_H^{D\tau,\bar{h},l} & = & \{M \in \mathcal{E}_H^D \mid M \stackrel{\tau}{\Longrightarrow} M' \wedge M' \stackrel{\bar{h}@k}{\not\Longrightarrow} \text{ for } k \in Loc \text{ and } M \stackrel{\bar{h}@l}{\Longrightarrow}\} \\
\mathcal{E}_H^{D\tau,l} & = & \{M \in \mathcal{E}_H^D \mid M \stackrel{\hat{\tau}}{\Longrightarrow} M' \wedge M' \stackrel{\bar{h}@k}{\not\Longrightarrow} \text{ for } k \in Loc \text{ and } M \notin \mathcal{E}_H^{D\tau,\bar{h},l}\} \\
\mathcal{E}_H^{D\bar{h},l} & = & \{M \in \mathcal{E}_H^D \mid M \stackrel{\bar{h}@l}{\Longrightarrow} \text{ and } \Pi \notin \mathcal{E}_H^{D\tau,\bar{h},l}\}
\end{array}
$$

Consider the labels of the states reachable from E_3 given in Figure 9. For a given $l \in Loc$, we construct the binary relation $\mathcal{S}_l$ as follows:

$$
\begin{aligned}
\mathcal{S}_l &= \{((l[\![E_3^1]\!]|M) \setminus H, l[\![E_3^1]\!] \setminus H) \mid M \in \mathcal{E}_H^D\} \\
&\cup \{((l[\![E_3^2]\!]|M) \setminus H, l[\![E_3^5]\!] \setminus H) \mid M \in \mathcal{E}_H^{D\tau,\bar{h},l}\} \\
&\cup \{((l[\![E_3^2]\!]|M) \setminus H, l[\![E_3^6]\!] \setminus H) \mid M \in \mathcal{E}_H^{D\bar{h},l}\} \\
&\cup \{((l[\![E_3^2]\!]|M) \setminus H, l[\![E_3^8]\!] \setminus H) \mid M \in \mathcal{E}_H^{D\tau,l}\} \\
&\cup \{((l[\![E_3^3]\!]|M) \setminus H, l[\![E_3^6]\!] \setminus H) \mid M \in \mathcal{E}_H^D\} \\
&\cup \{((l[\![E_3^4]\!]|M) \setminus H, l[\![E_3^7]\!] \setminus H) \mid M \in \mathcal{E}_H^D\} \\
&\cup \{((l[\![E_3^5]\!]|M) \setminus H, l[\![E_3^5]\!] \setminus H) \mid M \in \mathcal{E}_H^D\} \\
&\cup \{((l[\![E_3^6]\!]|M) \setminus H, l[\![E_3^6]\!] \setminus H) \mid M \in \mathcal{E}_H^D\} \\
&\cup \{((l[\![E_3^7]\!]|M) \setminus H, E_3^7[\![\setminus]\!]H) \mid M \in \mathcal{E}_H^D\} \\
&\cup \{((l[\![E_3^8]\!]|M) \setminus H, l[\![E_3^8]\!] \setminus H) \mid M \in \mathcal{E}_H^D\} \\
&\cup \{((l[\![E_3^8]\!]|M) \setminus H, l[\![E_3^2]\!] \setminus H) \mid M \in \mathcal{E}_H^D\}.
\end{aligned}
$$

It is easy to prove that $\mathcal{S}_l$ is a weak bisimulation. Indeed, since E_3 does not perform any *goto* action, the proof is analogous to the one in Appendix A.

Since $\mathcal{S}_l$ is a weak bisimulation for all $l \in Loc$, the binary relation $\mathcal{S} = \cup_{l \in Loc} \mathcal{S}_l$ is a weak bisimulation too. The property that E_3 is *M_BNDC* follows from the fact that $E_3 \equiv E_3^1$ and $\mathcal{S}$ contains all pairs of the form $((l[\![E_3]\!]|M) \setminus H, l[\![E_3]\!] \setminus H)$ with $M \in \mathcal{E}_H^D$.