On Intransitive Non-interference in Some Models of Concurrency

Roberto Gorrieri and Matteo Vernali

Dipartimento di Scienze dell'Informazione, Università di Bologna,
Mura A. Zamboni, 7, 40127 Bologna, Italy
email: gorrieri@cs.unibo.it

Abstract. Intransitive non-interference (*INI* for short) is a behavioural property extensively studied by Rushby over deterministic automata with outputs associated to transitions (Mealy machines) in order to discuss the security of systems where declassification of secret information is allowed. In this paper, we first propose a natural transposition of Rushby's definition on deterministic labelled transition systems, we call *INI* as well, and then an alternative, yet more easily checkable, formulation of *INI*, called *NI with downgraders* (*NID* for short). We show how *NID* can be naturally extended to the case of nondeterministic automata by using a variation of it based on bisimulation equivalence (*BNID*). The most novel contribution of this paper is the extension of this theory on the class of Petri nets called elementary net systems: we propose a semi-static technique, called *PBNID* and based on the inspection of the net structure, that is shown to be equivalent to *BNID*.

1 Introduction

Non-interference has been defined in the literature as an extensional property based on some observational semantics: the high part (usually considered the secret part) of a system does not interfere with the low part (i.e., the public one) if whatever is done at the high level produces *no visible effect* on the low part of the system.

The original notion of non-interference in [14] was defined, using trace semantics, for deterministic automata with outputs. Generalized notions of non-interference were then designed to include (nondeterministic) labeled transition systems and finer notions of observational semantics such as bisimulation (see, e.g., [24, 7, 25, 9]). The security properties in this class are based on the dynamics of systems; they are defined by means of one (or more) equivalence check(s); hence, non-interference checking is as difficult as equivalence checking, a well-studied hard problem in automata theory and concurrency theory.

When it is necessary to declassify information (e.g., when a secret plan has to be made public for realization), the two-level approach (H/secret – L/public) is usually extended with one intermediate level of downgrading (D), so that when an action in that class is performed, the previously performed high actions become observable to low users. This security policy is known under the name of

intransitive noninterference (*INI* for short) because the information flow relation is considered not transitive: even if information flows from level H to D and from D to L are allowed, direct flows from H to L are forbidden.

Rusby [23] defines *INI* only for deterministic finite automata with output associated to transitions (Mealy machines), based on earlier work of Haigh and Young [13], in turn influenced by [15]. The basic intuition is that a machine in this class is *INI* if for any trace σ, the states reached after σ and $ipurge(\sigma)$ (where all the action in H not justified by subsequent actions in D are removed) offer the same output. Here we redefine it on deterministic finite labeled transition systems, by interpreting output equivalence as low language equivalence of the reached states. This extensional definition is, however, rather cumbersome as it requires an equivalence check for any possible trace, hence infinitely many checks, despite the fact that finite automata have only finitely many states and transitions.

Therefore, we propose a different property, called *NID* (non-interference with downgraders) that essentially requires that for any high transition the source state and the target one are equivalent for low observers. *NID* is reminescent of the so-called *unwinding* condition in [23], even if ours better reveals the role of downgraders. We prove that *NID* is an equivalent, yet more easily checkable, characterization of *INI*.

When extending the approach to nondeterministic systems, we observe that *NID* is inadequate in some cases, as trace semantics is too weak to discriminate situations of possible danger. Nonetheless, by considering bisimulation semantics in place of trace semantics, we get *BNID* which seems to be fully satisfactory.

Intuitively, the many definitions of (transitive or intransitive) noninterference that have been proposed in the literature try to capture the essence of information flow as an extensional property. On the contrary, one may think that there are clear physical reasons for the occurrence of an information flow, that can be better understood if one exploits a computational model where causality of actions and conflict among actions can be modelled directly. Indeed, this is not the case of labeled transitions systems, a typical example of an *interleaving* model, where parallelism is not primitive.

For this reason, in [2–4] Busi and Gorrieri have shown that these extensional noninterference properties can be naturally defined also on Petri Nets, in particular on Elementary Nets [6], a well-known model of computation where causality and conflict are primitive concepts. More interestingly, they address the problem of defining statically non-interference for Elementary Nets, by looking at the structure of the net systems under investigation:

- in order to better understand the relationship between a flow of information and the causality (or conflict) relation between the activities originating such a flow, hence grounding more firmly the intuition about what is an interference, and
- in order to find more efficiently checkable noninterference properties that are sufficient (sometimes also necessary) conditions for those that have already received some support in the literature.

Structural noninterference was first proposed in [3, 4] on the basis of the absence of particular places in the net. Two special classes of places are of interest: *causal places*, i.e., places for which there are an incoming high transition and an outgoing low transition; and, *conflict places*, i.e. places for which there are both low and high outgoing transitions. Intuitively, causal places represent potential source of interference because the occurrence of the high transition is a prerequisite for the execution of the low transition. Similarly, conflict places represent potential source of interference because if the low event is not executable, then we can derive that a certain high transition has occurred. The absence of causal and conflict places is clearly a static property that can be easily checked (linear in the size of the net) by a simple inspection of the (finite) net structure. Interestingly enough, we show that absence of such places is a sufficient condition to ensure *BNID*.

In order to characterize more precisely *BNID*, the notion of causal place and conflict place is slightly refined, yielding the so-called *active* causal place and *active* conflict place. These new definitions are based also on a limited exploration of the state-space of the net (i.e. of its marking graph), hence, the absence of such places is not a purely structural property, rather a hybrid, semi-static property. When active causal and active conflict places are absent, we get a property, called *Positive Place–Based Non–Interference with Downgraders* (*PBNID* for short), which turns out to be equivalent to *BNID*.

1.1 Contribution of this paper

The first, minor contribution of the paper is a reformulation of Rushby's definition [23] of (basic) noninterference (*NI* for short) in the setting of deterministic labeled transition systems (LTSs for short). Apparently, this reformulation, called *NI* with abuse of notation, is new because it is the only one requiring low equivalence of the reached states. Then, an alternative characterization of *NI* called *SNDC*, is given in terms of a finite number (equal to the number of high transitions in the LTS) of equivalence checks. One further contribution is its extension to nondetermistic systems, yielding the bisimulation-based *SNDC*, called *SBNDC* for short. Both *SNDC* and *SBNDC* are not original [7, 9, 5], but new is the result that proves that *NI* is equivalent to *SNDC* for deterministic LTSs.

The same game is played for intransitive noninterference. First, we provide a reformulation of Rushby's definition [23] in the setting of deterministic LTSs, which is new. Then, the alternative characterization of *INI*, called *NID*, is given in terms of a finite number (equal to the number of high transitions in the LTS) of equivalence checks. Then, one further contribution is its extension to nondetermistic systems, yielding the bisimulation-based *NID*, called *BNID* for short. Also in this case, *NID* and *BNID* were defined already in [5] under the name of $DSNDC$ and $DSBNDC$, but new is the result of equivalence between *INI* and *NID*.

The main result of the paper is the extension of the structural approach to noninterference of [2–4] to the richer setting of intransitive noninterference. We

prove that an elementary net is *BNID* iff it is *PBNID*. Hence, we essentially prove that there is an illegal information flow directly from H to L iff there is a direct causality (or conflict) relation between a high transition and a low one.

From a complexity point of view, *BNID* over LTSs has complexity $O(n^3)$ (see [5] for a discussion on the complexity of $DSBNDC$) where n is the number of states. By following the same arguments reported in [10], we can conclude that *BNID* on elementary nets has complexity $O(pn2^{3p})$, where p is the number of places and n the number of transitions; hence, since the marking graph has $O(2^p)$ states, *BNID* is cubic in the number of the states also for elementary net systems. The complexity of checking for the absence of potential causal/conflict places is $O(f + p)$, where f is the number of arcs in the net. The complexity of *PBNID* is $O(pn2^{2p})$ in the worst case, hence *PBNID* is quadratic in the number of states.

These algorithms, for the case without downgrading actions (i.e., classic non-interference), have been already implemented in a software tool, called the Petri Net Security Checker (PNSC for short), which provides functionalities for creating, editing and executing Petri nets, as well as automatically detecting places that are potentially/actively causal/conflict [10].

1.2 Related work

Deterministic Systems

In [20] Pinsky studies intransitive noninterference (INI) over deterministic Mealy machine; his definition is very close to Rushby's original one; his main contribution is an algorithm for detecting this property, whose complexity is not examined, but seems exponential. This work has been extended in [21] to cover the case of nondeterministic machines.

In [11, 12] an algorithmic approach is proposed to solve the problem of verification of the property of INI, using tools and concepts of the theory of supervisory control of discrete event systems (DES). The algorithm is exponential in the number of states of the deterministic automaton.

In [17] Ron van der Meyden discusses some alternative variations of INI in the area of deterministic Moore machines. He suggests that Rushby's definition of *INI* is somehow inadequate in some practical cases and he proposes some stronger variations that better fit with Rushby's unwinding conditions.

Nondeterministic Systems

In [22] Roscoe and Goldsmith propose a rather discriminating definition of INI for nondeterministic systems which is based on strong assumptions about determinacy of low-level view. Their view is somehow related to the debate about what nondeterminism can be actually observed.

Mullins in [19] study the problem of intransitive non-interference for a variant of CCS [18], hence on (nondeterministic) LTSs. His proposed property, called *Admissible Interference* (*AI* for short), is based on trace semantics: E is *AI* if

for all reachable state E', $(E' \setminus D)/H$ (where downgrading actions are forbidden and high level actions are invisible) is (weak) trace equivalent to $E' \setminus H \cup D$ (where both downgrading actions and high level actions are forbidden). It is not difficult to prove that *AI* is strictly weaker than *BNID*. For instance, the system $E = h.l_1.\mathbf{0} + l_1.\mathbf{0} + l_2.\mathbf{0}$ is *AI*, while it is not even *NID* and indeed, E is not secure because a low level user that cannot perform l_2 is sure that h has been performed.

Lafrance and Mullins in [16] propose a (weak) bisimulation-based version of *AI*, called $BNAI$ (reminiscent of Focardi and Gorrieri's $SBSNNI$), which turns out to be strictly weaker than *BNID*. For instance, system $F = h.l_1.\mathbf{0} + \tau.l_1.\mathbf{0} + l_2.\mathbf{0}$, which is a slight variant of E above, is $BNAI$, but it is not even *NID* and indeed, F is not secure for the same reason as above.

In their thorough study [5] Bossi et al. consider various definitions of unwinding-based INI properties for a CCS-like language, with no precise indication of which of them is the right one. Among these properties, $DSBNDC$ is actually the same definition as our *BNID*. No attempt is made to compare their many variants with Rushby's original definition. They prove general composability properties (w.r.t. some operators of CCS) and provide conditions under which horizontal and vertical refinements are preserved.

Petri Nets

To the best of our knowledge, this is the first paper approaching INI over Petri nets. It builds over the results of [3, 4] for the case of (basic) two-level noninterference, where *SBNDC* (the property obtained from *BNID* in absence of downgrading actions) is proved to be equivalent to *PBNI+*(the property obtained from *PBNID* when downgrading actions are absent).

The paper is organised as follows. In Section 2 we recall the basic background definitions about Labeled Transition Systems and Elementary Net systems. In Section 3 we study basic noninterference, *NI* for short, starting from the definition proposed by Rushby for deterministic automata with outputs. We reformulate it over deterministic LTS and then propose our alternative, unwinding-like definition (*SNDC*) and prove the equivalence of the two. We discuss the extension to nondeterministic LTSs, yielding *SBNDC*. In Section 4 we present Rushby's definition of Intransitive Noninterference, *INI* for short, then we reformulate it over deterministic LTSs and provide an alternative characterization in terms of a local property, called *NID*. We then consider LTSs in general, and the property *BNID*, based on bisimulation. In Section 5 we define *BNID* over elementary net systems. We then formulate the semi-static property *PBNID* by looking at the presence of (active) causal/conflict places, and provide the proof that *BNID* is the same as *PBNID*. In Section 6 we draw some conclusions.

2 Background

2.1 Labeled Transition Systems

Here we recall some basic definitions over LTSs.

Definition 1. *A labeled transition system is a triple $TS = (St, E, \rightarrow)$ where*

- *St is the set of states*
- *E is the set of events*
- *$\rightarrow \subseteq St \times E \times St$ is the transition relation.*

In the following we use $s \xrightarrow{e} s'$ to denote $(s, e, s') \in \rightarrow$. Given a transition $s \xrightarrow{e} s'$, s is called the source*, s' the* target *and e the* label *of the transition. A* rooted *transition system is a pair (TS, s_0) where $TS = (St, E, \rightarrow)$ is a transition system and $s_0 \in St$ is the* initial state*. A transition system $TS = (St, E, \rightarrow)$ is* finite-state *if both St and E are finite sets.*

Definition 2. *A labeled transition system $TS = (St, E, \rightarrow)$ is deterministic iff the following holds: $\forall s \in St, \forall e \in E$ if $s \xrightarrow{e} s_1$ and $s \xrightarrow{e} s_2$ then $s_1 = s_2$.*

This means that $\forall s \in St, \forall e \in E$ there is at most one s' such that $s \xrightarrow{e} s'$, but such an s' may also not exist.

Definition 3. *Given a labeled rooted transition system $TS = (St, E, \rightarrow, s_0)$, a* path *from s_1 to s_{n+1} is a sequence of transitions $s_1 \xrightarrow{e_1} s_2 \ldots s_n \xrightarrow{e_n} s_{n+1}$. We say that s' is* reachable *from s if there exists a path from s to s'. TS is* reduced *when the set of states reachable from s_0 is exactly St, i.e., all the states are reachable from the initial state.*

In the following, we will usually consider only reduced labeled rooted transition systems.

Definition 4. *Let $TS = (St, E, \rightarrow, s_0)$ be a rooted transition system. A* trace *of TS is a (possibly empty) sequence of events $e_1 \ldots e_n$ such that there exists a path $s_1 \xrightarrow{e_1} \ldots s_n \xrightarrow{e_n} s_{n+1}$ with $s_1 = s_0$. The set of* traces *of TS is denoted by $Tr(s_0)$.*

Let $TS = (St, E, \rightarrow)$ be a transition system and let $s_1, s_2 \in St$. We say that s_1 and s_2 are trace equivalent *(denoted with $s_1 \sim s_2$) iff $Tr(s_1) = Tr(s_2)$.*

Definition 5. *A* bisimulation *between TS_1 and TS_2 is a relation $R \subseteq (St_1 \times St_2)$ such that if $(s_1, s_2) \in R$ then for all $e \in (E_1 \cup E_2)$*

- *$\forall s_1'$ such that $s_1 \xrightarrow{e} s_1'$, then $\exists s_2'$ such that $s_2 \xrightarrow{e} s_2'$ and $(s_1', s_2') \in R$*
- *$\forall s_2'$ such that $s_2 \xrightarrow{e} s_2'$, then $\exists s_1'$ such that $s_1 \xrightarrow{e} s_1'$ and $(s_1', s_2') \in R$.*

If $TS_1 = TS_2$ we say that R is a bisimulation on TS_1.

It is well-known that trace semantics and bisimulation semantics do coincide over deterministic LTSs, while in general bisimulation semantics is more discriminating (see, e.g., [18]).

2.2 Elementary Net Systems

Here we introduce basic definitions about the class of Petri Nets we use. Some familiarity with Petri net terminology is assumed. More details can be found in [6, 3].

Definition 6. *An* elementary net *is a tuple* $N = (S, T, F)$*, where*

- S *and* T *are the (finite) sets of* places *and* transitions*, such that* $S \cap T = \emptyset$
- $F \subseteq (S \times T) \cup (T \times S)$ *is the* flow relation*, usually represented as a set of* directed arcs *connecting places and transitions.*

A finite subset of S is called a *marking*. Given a marking m and a place s, if $s \in m$ then we say that the place s contains a token, otherwise we say that s is empty.

Let $x \in S \cup T$. The *preset* of x is the set ${}^\bullet x = \{y \mid F(y, x)\}$. The *postset* of x is the set $x^\bullet = \{y \mid F(x, y)\}$. The preset and postset functions are generalized in the obvious way to sets of elements: if $X \subseteq S \cup T$ then ${}^\bullet X = \bigcup_{x \in X} {}^\bullet x$ and $X^\bullet = \bigcup_{x \in X} x^\bullet$. A transition t is enabled at marking m if ${}^\bullet t \subseteq m$ and $t^\bullet \cap m = \emptyset$. The firing (i.e., execution) of a transition t enabled at m produces the marking $m' = (m \setminus {}^\bullet t) \cup t^\bullet$. This is usually written as $m[t\rangle m'$. With the notation $m[t\rangle$ we mean that there exists m' such that $m[t\rangle m'$.

An *elementary net system* is a pair (N, m_0), where N is an elementary net and m_0 is a marking of N, called *initial marking*. With abuse of notation, we use (S, T, F, m_0) to denote the net system $((S, T, F), m_0)$.

The set of *markings reachable from* m, denoted by $[m\rangle$, is defined as the least set of markings such that

- $m \in [m\rangle$
- if $m' \in [m\rangle$ and there exists a transition t such that $m'[t\rangle m''$ then $m'' \in [m\rangle$.

The set of *firing sequences* is defined inductively as follows:

- m_0 is a firing sequence;
- if $m_0[t_1\rangle m_1 \ldots [t_n\rangle m_n$ is a firing sequence and $m_n[t_{n+1}\rangle m_{n+1}$ then also $m_0[t_1\rangle m_1 \ldots [t_n\rangle m_n[t_{n+1}\rangle m_{n+1}$ is a firing sequence.

Given a firing sequence $m_0[t_1\rangle m_1 \ldots [t_n\rangle m_n$, we call $t_1 \ldots t_n$ a *transition sequence*. We use σ to range over transition sequences.

The *marking graph* of a net system N is the transition system

$$MG(N) = ([m_0\rangle, T, \{(m, t, m') \mid m \in [m_0\rangle \wedge t \in T \wedge m[t\rangle m'\}).$$

A net is *transition simple* if the following condition holds for all $x, y \in T$: if ${}^\bullet x = {}^\bullet y$ and $x^\bullet = y^\bullet$ then $x = y$. A marking m contains a *contact* if there exists a transition $t \in T$ such that ${}^\bullet t \subseteq m$ and $t^\bullet \cap m \neq \emptyset$. A net system is *contact–free* if no marking in $[m_0\rangle$ contains a contact. A net system is *reduced* if each transition can occur at least one time: for all $t \in T$ there exists $m \in [m_0\rangle$ such that $m[t\rangle$. In the following we consider contact-free elementary net systems that are transition simple and reduced.

3 Basic Noninterference on LTSs

3.1 Rushby's definition

The definition of Rushby [23], that elaborates over the original one of Goguen and Meseguer [14], is given over finite-state deterministic automata with outputs associated to transitions: these are essentially Mealy machines, where every action is allowed from each state.

Definition 7. *A system M is a tuple $(S, A, O, step, output)$ where:*

- *S is a finite set of states, with a distinguished initial state s_0;*
- *A is a finite set of actions (or inputs);*
- *O is a finite set of outputs;*
- *$step : S \times A \rightarrow S$ is the transition function, and*
- *$output : S \times A \rightarrow O$ is the function that returns the output associated to the transition.*

Function step can be extended to sequences of actions by means of function $run : S \times A^* \rightarrow S$ as follows:

$$run(s, \epsilon) = s$$
$$run(s, a\alpha) = run(step(s, a), \alpha)$$

where ϵ denotes the empty sequence and α a sequence of actions.

Rushby's definition is given in general for a set of security domains onto which a particular security policy is described. For simplicity's sake, we restrict our attention to the simple case of two levels only, usually called *high* – H (for classified or secret information) and *low* – L (for public information). Hence, we reformulate his definition in this simplified setting.[1] Some auxiliary definitions are necessary. Given a set $\mathcal{D} = \{L, H\}$ of security domains we consider a function $dom : A \rightarrow \mathcal{D}$ which associates a security domain to each action. A security policy (or *interference* relation) is a reflexive relation $\rightarrow_i \subseteq \mathcal{D} \times \mathcal{D}$, which, in our simplified setting, is given by the relation $\{(L, L), (L, H)(H, H)\}$, stating that information can flow from low to high. With $\nrightarrow_i$ we denote the complementary relation of *noninterference*, that in our case is just the relation $\{(H, L)\}$, meaning that information flows from high to low are forbidden. A security policy is *transitive* if its interference relation$\rightarrow_i$ is so, which is true in our case.

There is an *information flow* from domain u to domain v when the actions performed by domain u make the system, as observed by domain v, different from the case when such actions are not performed.

[1] This simplification is not against generality: given a n-level system, this can be verified by means of a set of two level checks, for any partitioning of the n levels in two groups.

Definition 8. *Given $v \in \mathcal{D}$ and $\alpha \in A^*$, we define $purge(\alpha, v)$ as the subsequence of α obtained by removing all the actions associated with the domains u such that $u \nrightarrow_i v$:*

$$purge(\epsilon, v) = \epsilon$$
$$purge(a\alpha, v) = \begin{cases} a\ purge(\alpha, v) & \text{if } dom(a) \rightarrow_i v \\ purge(\alpha, v) & \text{otherwise.} \end{cases}$$

In our specialized setting, as the forbidden information flow is only the one from H to L, $purge(\alpha, v)$ returns α when $v = H$, while it simply removes the high actions from α when $v = L$.

Definition 9. *A system M is NI (i.e., non-interferent) iff the following holds:*

$$\forall \alpha \in A^*\ \forall a \in A\ output(run(s_0, \alpha), a) = output(run(s_0, purge(\alpha, dom(a))), a).$$

This essentially amounts to say that, whenever a low action a is to be performed, the state s_1, reached by performing α, and the state s_2, reached by performing $purge(\alpha, L)$, offer the same output when performing a. Hence, *NI* holds if low outputs (i.e., the outputs performed in correspondence of low inputs) do not depend on high inputs.

3.2 How to define *NI* on deterministic LTSs?

Our aim is to analyse systems that can perform two kinds of actions: high level actions, representing the interaction of the system with high level users, and low level actions, representing the interaction with low level users. We want to verify if the interplay between the high user and the high part of the system can affect the view of the system as observed by a low user. We assume that the low user knows the structure of the system, and we check if, in spite of this, he is not able to infer the behavior of the high user by observing the low view of the execution of the system. Hence, we consider LTSs whose set of events E is partitioned into two subsets: the set E_H of high level events and the set E_L of low level events. To emphasize this partition we use the following notation: with $(St, E_L, E_H, \rightarrow)$ we denote the LTS $(St, E, \rightarrow)$ where $E = E_L \cup E_H$ and $E_L \cap E_H = \emptyset$.

We would like to redefine property *NI* over deterministic labeled transition systems in a way to preserve as much as possible the original intuition of the definition given by Rushby. We have to cope with some issues. First of all, the label of a transition in an LTS is either an input (high or low) or an output (high or low). Hence, the rigid synchrony of inputs and outputs is invalid in this more general model. Moreover, the equivalence on outputs required by the definition of *NI* above makes little sense on deterministic LTSs, where usually equivalence is expressed in terms of language equality (i.e., equality of the set of traces).

We first define the variant purge function in this context, that we call *hide*.

Definition 10. *Given a deterministic LTS $(St, E_L, E_H, \rightarrow)$, function* hide *(or low view) of a sequence of events $\alpha \in E^*$ is defined as follows:*

- $hide(\epsilon) = \epsilon$
- $hide(a\alpha) = \begin{cases} a\ hide(\alpha) & \textit{if } a \in E_L \\ hide(\alpha) & \textit{otherwise} \end{cases}$

Also function *run* should be adapted as follows, because deterministic LTSs do not ensure that in any state a transition is present for any event:

- $run(s, \epsilon) = s$
- $run(s, a\alpha) = \begin{cases} run(s', \alpha) \text{ if } s \xrightarrow{a} s' \\ \textit{undefined} \text{ otherwise} \end{cases}$

So the definition of *NI* in this setting should be something like this:

$$\forall \alpha \in E^*\ run(s_0, \alpha)\ \sim_L\ run(s_0, hide(\alpha)).$$

where $\sim_L$ is some notion of low-equivalence we have not yet identified and the equivalence is meant to hold whenever the left hand side of the equation is defined, i.e. when trace α is executable.

Now let us try to identify what is a sensible candidate for $\sim_L$. The idea is to have that the two reached states are indistinguishable for a low observer, at least until some high action takes place. Hence, we first define the initial low view $\Lambda(\alpha)$ of a trace $\alpha \in E^*$ as follows:

- $\Lambda(\epsilon) = \epsilon$
- $\Lambda(a\alpha) = \begin{cases} a\ \Lambda(\alpha) & \text{if } a \in E_L \\ \epsilon & \text{otherwise.} \end{cases}$

Λ is similar to function *hide*, however it differs because Λ truncates α to its first high level action.

Definition 11. *Given a deterministic LTS, we say that two states s_1 and s_2 are* initial low-view equivalent*, denoted with $s_1 \sim_L s_2$, iff $\Lambda(Tr(s_1)) = \Lambda(Tr(s_2))$.*

It is immediate to observe that $\sim_L$ is an equivalence relation. Moreover, it is decidable for finite-state LTS's (actually PSPACE-complete), as it can be reduced to the trace equivalence problem.

One may wonder why we need function Λ in the definition above and do not use instead function *hide*. The following example explains this point.

Example 1. The deterministic LTS in Figure 1 is clearly insecure because if a low user observes action l_2 then (s)he is sure that the high action h has been performed an odd number of times. Indeed, this system is not *NI*, but it would satisfy the variant definition where function *hide* is used in place of Λ.

So, we are now ready to define our notion of noninterference, we call *NI* with abuse of notation, as from now on this definition surpasses the previous one.

Definition 12. *Given a deterministic labeled rooted transition system $TS = (St, E_L, E_H, \rightarrow, s_0)$, we say that TS satisfies NI if the following holds: $\forall \alpha \in E^*$ if $run(s_0, \alpha)$ is defined, then so is $run(s_0, hide(\alpha))$ and*

$$run(s_0, \alpha)\ \sim_L\ run(s_0, hide(\alpha))$$

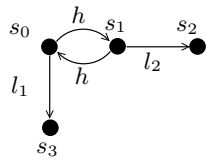

Fig. 1. An LTS not satisfying NI

3.3 Unwinding – *SNDC*

The definition of *NI* is rather cumbersome and difficult to check, because of the universal quantification over all possible traces in E^*. As a matter of fact, a direct, brute force algorithm checking *NI* would fail miserably because the number of equivalence checks is infinite in principle, even for finite-state LTS's.

However, in [15, 23] are reported some local conditions ensuring *NI* that allows for a better algorithmic verification of *NI*. These are called *unwinding* conditions in the jargon of information flow security.

In this section we want to propose one local property which is necessary and sufficient to prove *NI*. This property is called *SNDC* [7, 9, 5].

Definition 13. *A reduced deterministic LTS* $(St, E_L, E_H, \rightarrow, s_0)$ *satisfies SNDC iff* $\forall s \in St, \forall h \in E_H$ *whenever* $s \xrightarrow{h} s'$ *we have that* $s \sim_L s'$.

Hence, checking *SNDC* is decidable for finite-state LTS's, as we have to make as many equivalence checks as are the high transitions in the LTS, which are finitely many, and each equivalence check $s \sim_L s'$ is decidable.

We can then state the main theorem for this part of the paper, whose proof is postponed to Appendix A.

Theorem 1. *A reduced deterministic LTS* $(St, E_L, E_H, \rightarrow, s_0)$ *satisfies SNDC iff it satisfies NI.*

3.4 Extending the approach to nondeterminism

The definition of *SNDC* is rather satisfactory for deterministic LTSs, but when we move to general (i.e., possibly nondeterministic) LTSs, it is somehow inadequate, as the following example shows.

Example 2. The system in Figure 2 is *SNDC* because the states s_0 and s_1 are low-view equivalent. However, such a system is insecure, because a low level user willing to perform trace ll may be unable to do so and in such a case (s)he is sure that h has been performed.

This example clarifies that we need a more discriminating notion of initial low-view equivalence. A natural way out could be to refine the definition by using the finer bisimulation equivalence in place of trace equivalence.

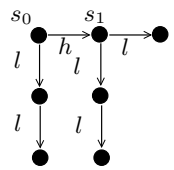

Fig. 2. An LTS that satisfies SNDC but that is not secure.

Definition 14. *Let $TS = (St, E_L, E_H, \rightarrow)$ be and LTS. An* initial low–view bisimulation *on TS is a relation $\mathcal{R} \subseteq St \times St$ such that if $(s_1, s_2) \in \mathcal{R}$ then for all $a \in E_L$:*

- *if $s_1 \xrightarrow{a} s_1'$ then there exists s_2' such that $s_2 \xrightarrow{a} s_2'$, with $(s_1', s_2') \in \mathcal{R}$*
- *if $s_2 \xrightarrow{a} s_2'$ then there exists s_1' such that $s_1 \xrightarrow{a} s_1'$, with $(s_1', s_2') \in \mathcal{R}$*

We say that two states s_1 and s_2 are initial low-view bisimulation equivalent*, denoted with $s_1 \approx_L s_2$, if there exists an initial low-view bisimulation $\mathcal{R}$ with $(s_1, s_2) \in \mathcal{R}$.*

It is immediate to observe that $\approx_L$ is an equivalence relation. Moreover, for finite-state LTS's, $\approx_L$ is decidable in polynomial time, because so is strong bisimulation.

We are now ready to define the improved information flow property.

Definition 15. *A reduced LTS $(St, E_L, E_H, \rightarrow, s_0)$ satisfies SBNDC if and only if $\forall s \in St, \forall h \in E_H$ whenever $s \xrightarrow{h} s'$ we have that $s \approx_L s'$.*

It is easy to observe that the system in Example 2 is not *SBNDC*. Moreover, *SBNDC* is decidable for finite-state LTS's because finite is the number of decidable equivalence checks $s \approx_L s'$. Finally, *SBNDC* is a conservative extension of *SNDC* as the following proposition states.

Theorem 2. *A reduced deterministic LTS $(St, E_L, E_H, \rightarrow, s_0)$ satisfies SBNDC iff it satisfies SNDC.*

By our experience, we conjecture that *SBNDC* is the right property in the setting of nondeterministic LTSs.

4 Intransitive Noninterference on LTSs

Basic (or transitive) noninterference is, from a practical point of view, too draconian: one often wants to make public some data that were secret previously. This operation is known as *declassification* and can be modeled naturally with an intransitive version of noninterference. Typically, we have three levels: H for secret actions, L for public actions and D for downgrading actions; we admit flows from H to L only if mediated by an action in D. Hence, in this three-level approach, the interference relation is $\{(L, L), (L, H)(H, H), (D, D), (L, D), (D, L), (H, D), (D, H)\}$ (where only (H, L) is forbidden), which is clearly not transitive.

4.1 Rushby's definition

In Section 3.1 we have presented Rushby's model, based on Mealy machines, for *NI*. Now we report his extension to intransitive non-interference, called *INI*, as described in [23]. We first need to identify which actions in a trace α are not to be deleted with the intransitive version of the *purge* function.

Definition 16. *Function sources* $: A^* \times \mathcal{D} \rightarrow \mathcal{P}(\mathcal{D})$ *is defined as follows:*

$$sources(\epsilon, u) = \{u\}$$
$$sources(a\alpha, u) = \begin{cases} sources(\alpha, u) \cup \{dom(a)\} & \text{if } \exists v : v \in sources(\alpha, u) \\ & \quad \wedge dom(a) \rightarrow_i v \\ sources(\alpha, u) & \text{otherwise.} \end{cases}$$

Essentially, $v \in sources(\alpha, u)$ means either that $v = u$ or that there exists a subsequence of α composed of actions in the domains $w_1, w_2, \ldots, w_n$ such that $w_1 \rightarrow_i w_2 \rightarrow_i \cdots \rightarrow_i w_n$, $v = w_1$ and $u = w_n$. Hence, when considering if a, performed before α, is allowed to influence domain u, we ask if there is any $v \in sources(\alpha, u)$ such that $dom(a) \rightarrow_i v$.

Definition 17. *Function ipurge* $: A^* \times \mathcal{D} \rightarrow A^*$ *(that is* intransitive-*purge) is defined as follows:*

$$ipurge(\epsilon, u) = \epsilon$$
$$ipurge(a\alpha, u) = \begin{cases} a\ ipurge(\alpha, u) & \text{if } dom(a) \in sources(a\alpha, u) \\ ipurge(\alpha, u) & \text{otherwise.} \end{cases}$$

Essentially, $ipurge(\alpha, u)$ is the subsequence of α where all the actions that cannot interfere with u are removed. It is interesting to observe that, in the simplified case of three levels only we consider in this paper, $ipurge(\alpha, H) = \alpha$, $ipurge(\alpha, D) = \alpha$, while $ipurge(\alpha, L)$ returns the subsequence of α where all the high level actions occurring before a low level action are removed, unless a downgrading action does occur in between the two.

Definition 18. *A system M is INI (i.e., intransitive non-interferent) iff the following holds:*

$$\forall \alpha \in A^*\ \forall a \in A\ output(run(s_0, \alpha), a) = output(run(s_0, ipurge(\alpha, dom(a))), a).$$

This essentially amounts to say that, whenever a low action a is to be performed, the state s_1, reached by performing α, and the state s_2, reached by performing $ipurge(\alpha, L)$, offer the same output when performing a. Hence, *INI* holds if low outputs (i.e., the outputs performed in correspondence of low inputs) do not depend on non-downgraded high inputs.

4.2 Defining *INI* for deterministic LTSs

As we have three levels of actions, the definition of LTS should reflect this partitioning, hence $TS = (St, E_L, E_D, E_H, \rightarrow)$. We start by defining an intransitive variant of function *hide*.

Definition 19. *Given a deterministic LTS* $(St, E_L, E_D, E_H, \rightarrow)$*, the function* ihide *(or intransitive low view) of a sequence of events* $\alpha \in E^*$ *is defined as follows:*

- $ihide(\epsilon) = \epsilon$
- $ihide(\alpha a) = \begin{cases} ihide(\alpha)a & \text{if } a \in E_L \\ ihide(\alpha) & \text{if } a \in E_H \\ \alpha a & \text{if } a \in E_D \end{cases}$

Function Λ extends naturally to the case of the presence of downgrading actions: it truncates traces at the point the first high action or downgrading action is met. Even if the definition of Λ works as expected also in the three level setting, we prefer to change its name to Δ to emphasize that we are working on this richer scenario. Hence, also the definition of initial low-view equivalence is syntactically slightly changed: $s_1 \sim_D s_2$ iff $\Delta(Tr(s_1)) = \Delta(Tr(s_2))$. It is immediate to observe that $\sim_D$ is an equivalence relation. Moreover, it is decidable for finite-state LTS's.

Definition 20. *Given a deterministic labelled rooted transition system* $TS = (St, E_L, E_D, E_H, \rightarrow, s_0)$*, we say that* TS *satisfies INI iff the following holds:* $\forall \alpha \in E^*$ *if* $run(s_0, \alpha)$ *is defined, then so is* $run(s_0, ihide(\alpha))$ *and then*

$$run(s_0, \alpha) \sim_D run(s_0, ihide(\alpha)).$$

The definition above is indeed a conservative extension of *NI* because, if we consider a deterministic labelled rooted transition system with empty set of downgrading actions, we get that *INI* and *NI* do coincide. This is due to the fact that $ihide(\alpha) = hide(\alpha)$ if $\alpha \in (E_L \cup E_H)^*$.

Theorem 3. *Given a deterministic labelled rooted transition system* $TS = (St, E_L, \emptyset, E_H, \rightarrow, s_0)$*, it holds that* TS *satisfies NI iff* TS *satisfies INI.*

4.3 Unwinding – *NID*

For basic (i.e., two-level) noninterference, we have shown that *SNDC* is a local property characterizing *NI*. Here we play the same game, by providing an obvious generalization of *SNDC*, we call *NID* (NonInterference with Downgraders) and then by showing that *NID* and *INI* are the same property (proof postponed to Appendix B).

Definition 21. *A reduced deterministic LTS* $(St, E_L, E_D, E_H, \rightarrow)$ *satisfies NID iff* $\forall s \in St, \forall h \in E_H$ *whenever* $s \xrightarrow{h} s'$ *we have that* $s \sim_D s'$.

Theorem 4. *A reduced deterministic LTS* $(St, E_L, E_D, E_H, \rightarrow, s_0)$ *satisfies NID iff it satisfies INI.*

NID is decidable for finite-state LTS's: we have to perform as many equivalence checks as are the transitions labeled with an high level action; moreover, each equivalence check of the form $s \sim_D s'$ is decidable, as it can be reduced to the trace equivalence problem.

4.4 Extending the approach to nondeterminism

The definition of *NID* seems rather satisfactory for deterministic LTSs, but when we move to nondeterministic systems, it may be inadequate, as the system in Figure 2 shows: the states s_0 and s_1 are low-view equivalent, but a low level user willing to perform trace ll may be unable to do so and in such a case (s)he is sure that h has been performed. A similar example is reported in Figure 3.

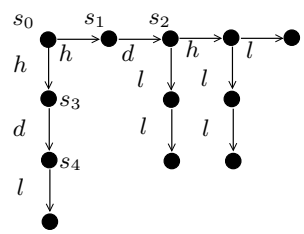

Fig. 3. An insecure nondeterministic system satisfying NID.

As discussed in Section 3.4, we should replace the trace based notion of initial low-view equivalence with a finer one based on bisimulation equivalence. The definition of *low-view observational equivalence* of Definition 14 is unchanged for this richer scenario, but to emphasize the difference (presence also of the intermediate level D), we denote it with $\approx_D$ instead of $\approx_L$.

We are now ready to define the improved information flow property, we call *BNID* (Bisimulation-based *NID*).

Definition 22. *A reduced LTS* $(St, E_L, E_D, E_H, \rightarrow)$ *satisfies BNID if and only if* $\forall s \in St, \forall h \in E_H$ *whenever* $s \xrightarrow{h} s'$ *we have that* $s \approx_D s'$.

It is easy to observe that the system in Figure 3 is not *BNID*. Moreover, *BNID* is decidable for finite-state LTS's because finite is the number of decidable equivalence checks $s \approx_D s'$. Finally, based on our experience, we conjecture that *BNID* is the right property in the setting of nondeterministic LTSs.

5 Noninterference on Elementary Nets

We first briefly survey the work in [3, 4], presenting the adaptation of the property *SBNDC* [7, 9] in the case of elementary net systems, and then the idea of *structural* noninterference. Then, Section 5.3 reports the original extension to the case of intransitive noninterference. We remind that we consider elementary net systems that are contact-free, reduced and transition simple.

5.1 A Dynamic Non-interference Property: *SBNDC*

We consider nets whose set of transitions is partitioned into two subsets: the set H of high level transitions and the set L of low level transitions. To emphasize this partition we use the following notation. Let L and H be two disjoint sets: with (S, L, H, F, m_0) we denote the net system $(S, L \cup H, F, m_0)$.

Among the many non-interference properties defined by Focardi and Gorrieri in [7–9], here we consider *SBNDC* (Strong Bisimulation Non-Deducibility on Composition). To properly define it over Petri nets, we need the auxiliary definition of initial low-view bisimulation over elementary net systems.

Definition 23. *Let $N = (S, L, H, F, m_0)$ be a net system. An* initial low–view bisimulation *is a relation $\mathcal{R} \subseteq \mathcal{P}_{fin}(S) \times \mathcal{P}_{fin}(S)$ such that if $(m_1, m_2) \in R$ then for all $t \in L$:*

- *if $m_1[t\rangle m_1'$ then there exists m_2' such that $m_2[t\rangle m_2'$, and $(m_1', m_2') \in \mathcal{R}$;*
- *if $m_2[t\rangle m_2'$ then there exists m_1' such that $m_1[t\rangle m_1'$, and $(m_1', m_2') \in \mathcal{R}$.*

We say that m_1 is initial low–view bisimilar *to m_2, denoted by $m_1 \approx_L m_2$, if there exists an initial low–view bisimulation $\mathcal{R}$ such that $(m_1, m_2) \in \mathcal{R}$.*

Now we are ready to define *SBNDC*.

Definition 24. *Let $N = (S, L, H, F, m_0)$ be a net system. N is SBNDC iff for all markings $m \in [m_0\rangle$ and for all $h \in H$ the following holds: $m[h\rangle m'$ implies $m \approx_L m'$.*

The intuition behind *SBNDC* is that, whenever a high transition h is performed, the markings before h and after h are observationally indistinguishable for a low observer. Note that *SBNDC* is clearly decidable for finite elementary net systems because the number of reachable markings is finite, as well as the set H of high transitions; hence, the number of equivalence checks is finite; moreover, each equivalence check $m \approx_L m'$ is decidable, because so is bisimulation equivalence over finite elementary net systems.

Example 3. As a simple case study, consider the net in Figure 4, which represents a mutually exclusive access to a shared resource (represented by the token in s) by a high-user (left part of the net) and a low-user (right part of the net). Even if it might appear, at first sight, that the system is secure (and indeed, it is *BSNNI* (Bisimulation Strong Nondeterministic Non-Interference) [7, 9]), actually

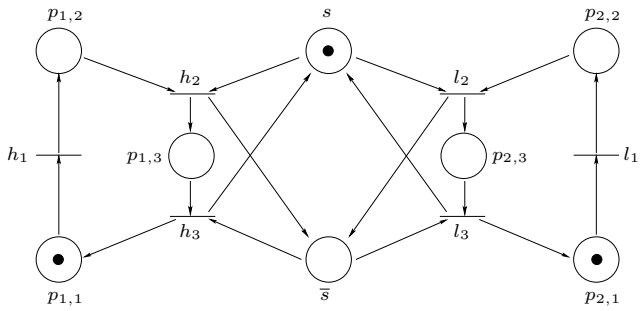

Fig. 4. The net system for a mutually exclusive access to a shared resource.

it is not *SBNDC*: consider the reachable marking $m = \{p_{1,2}, s, p_{2,2}\}$; it is easy to observe that $m[h_2\rangle m' = \{p_{1,3}, \overline{s}, p_{2,2}\}$ and $m[l_2\rangle$, but l_2 is not fireable from m'. Indeed, the system is not secure, because an unsuccesful attempt of a low level user to perform l_2 just after having performed l_1 will give him the information that the high user has performed h_2 but not yet h_3.

5.2 Structural Non-interference

Consider a net system $N = (S, L, H, F, m_0)$. Consider a low level transition l of the net: if l can fire, then we know that the places in the preset of l are marked before the firing of l; moreover, we know that such places become unmarked after the firing of l. If there exists a high level transition h that produces a token in a place s in the preset of l (see the system N_1 in Figure 5), then the low level user can infer that h has occurred if he can perform the low level transition l. We note that there exists a causal dependency between the transitions h and l, because the firing of h produces a token that is consumed by l. In this case we will say that s is a potential causal place.

Consider now the situation illustrated in the system N_2 of Figure 5: in this case, place s is in the preset of both l and h, i.e., l and h are competing for the use of the resource represented by the token in s. Aware of the existence of such a place, a low user knows that the high-level action h has been performed, if he is not able to perform the low-level action l. Place s represents a conflict between transitions l and h, because the firing of h prevents l from firing. In this case we will call s a potential conflict place.

In order to avoid the definition of a security notion that is too strong, and that rules out systems that do not reveal information on the high-level actions that have been performed, we need to refine the concepts illustrated above. In particular the potential causal place is an active causal place if there is an execution where the token produced by the high level transition is eventually consumed by the low level transition. Similarly, a potential conflict place is active if the token that could be consumed immediately by a high level transition can be later on also consumed by a low level transition. The formal definitions follow.

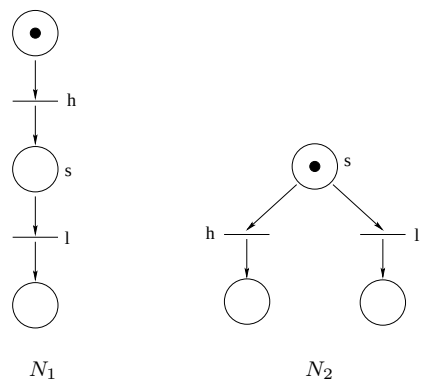

Fig. 5. Examples of net systems containing causal and conflict places.

Definition 25. *Let $N = (S, L, H, F, m_0)$ be an elementary net system. Let s be a place of N such that $s^\bullet \cap L \neq \emptyset$ (i.e., a token in s can be consumed by a low transition).*

The place $s \in S$ is a potentially causal place *if ${}^\bullet s \cap H \neq \emptyset$ (i.e., a token in s can be produced by a high transition). A potentially causal place s is an* active causal place *if the following condition holds: there exist $l \in s^\bullet \cap L$, $h \in {}^\bullet s \cap H$, $m \in [m_0\rangle$ and a transition sequence σ such that $m[h\sigma l\rangle$ and $s \notin t^\bullet$ for all $t \in \sigma$.*

The place $s \in S$ is a potentially conflict place *if $s^\bullet \cap H \neq \emptyset$ (i.e., the token in s can be consumed also by a high transition). A potentially conflict place is an* active conflict place *if the following condition holds: there exist $l \in s^\bullet \cap L$, $h \in s^\bullet \cap H$, $m \in [m_0\rangle$ and a transition sequence σ such that $m[h\rangle$, $m[\sigma l\rangle$ and $s \notin t^\bullet$ for all $t \in \sigma$.*

Definition 26. *Let $N = (S, L, H, F, m_0)$ be an elementary net system. We say that N is PBNI+ (*positive Place Based Non-Interference*) if, for all $s \in S$, s is neither an active causal place nor an active conflict place.*

The following non-trivial result, proved in [4], states that the behavioural non-interference property *SBNDC* is equivalent to the semi-static, structural property *PBNI+*.

Theorem 5. *Let $N = (S, L, H, F, m_0)$ be an elementary net system. Then N is PBNI+ iff N is SBNDC.*

An obvious consequence is that if N has no *potentially causal* and *potentially conflict* places, then N is *SBNDC*. Hence, a simple strategy to check if N is *SBNDC* is to first identify potential causal/conflict places, a procedure that has complexity $O(f + p)$ in the size of the net (p is the number of places and f of arcs). If no place of these sorts is found, then N is *PBNI+*, hence *SBNDC*. Otherwise, any such a candidate place should be better studied to check if it is actually an *active* causal/conflict place, a procedure that requires a limited

exploration of the marking graph. The complexity of *PBNI+* in the worst case is $O(pn2^{2p})$ (see [10] for details).

Observe that the net in Figure 4 of our running example is not *PBNI+* because place s is an active conflict (and also active causal) place.

5.3 Extending the approach for Intransitive Noninterference

Not too surprisingly, the theory presented in the two previous subsections can be adapted to the more general scenario when also downgrading actions are present. First of all, observe that the notion of initial low view bisimulation equivalence $\approx_L$ of Definition 23 works as expected also for the three level scenario; however, to better reflect the fact that also actions in D are possible, we rename it with $\approx_D$. Hence, the definition of *BNID* over elementary net systems is essentially the same as *SBNDC*.

Definition 27. *Let $N = (S, L, D, H, F, m_0)$ be a net system. N is BNID iff for all markings $m \in [m_0\rangle$ and for all $h \in H$ the following holds: $m[h\rangle m'$ implies $m \approx_D m'$.*

Then, we adapt the structural noninterference approach to the three level scenario.

Definition 28. *Let $N = (S, L, D, H, F, m_0)$ be an elementary net system. A place $s \in S$ is a* potentially causal *place if ${}^\bullet s \cap H \neq \emptyset$ and $s^\bullet \cap L \neq \emptyset$. Place $s \in S$ is an* active causal *place if it is potentially causal and there exist $h \in {}^\bullet s \cap H$, $l \in s^\bullet \cap L$, $m \in [m_0\rangle$ and a transition sequence $\sigma \in (H \cup L)^*$ such that $m[h\sigma l\rangle$ and for all $t \in \sigma$, $s \notin t^\bullet$.*

Observe that the only difference w.r.t. Definition 25 is that the transition sequence σ is constrained not to contain downgrading actions. The idea is that if a downgrading action d is performed inevitably in between h and l, then the flow is mediated by d and so the information flow becomes legal.

Definition 29. *Let $N = (S, L, D, H, F, m_0)$ be an elementary net system. A place $s \in S$ is a* potentially conflict *place if $s^\bullet \cap H \neq \emptyset$ and $s^\bullet \cap L \neq \emptyset$. Place $s \in S$ is an* active conflict *place if it is potentially conflict and there exist $h \in s^\bullet \cap H$, $l \in s^\bullet \cap L$, $m \in [m_0\rangle$ and a transition sequence $\sigma \in (H \cup L)^*$ such that $m[h\rangle$, $m[\sigma l\rangle$ and for all $t \in \sigma$, $s \notin t^\bullet$.*

Also for active conflict places, the only difference is the constraint on σ about downgrading actions. Indeed, if a downgrading action d is to be performed before l, then h and l are not really conflicting, because that conflict is made public by action d.

Definition 30. *Let $N = (S, L, D, H, F, m_0)$ be an elementary net system. We say that N is PBNID (*Positive Place Based Non Interference with Downgraders*) if for all $s \in S$, s is neither a active causal place, nor an active conflict place.*

The main result is that the dynamic property *BNID* and the semi-static property *PBNID* are actually equivalent on elementary net systems. The proof of the following theorem, based on the analogous proof in [4] of Theorem 5, is reported in Appendix C.

Theorem 6. *Let $N = (S, L, D, H, F, m_0)$ be an elementary net system. N is BNID if and only if N is PBNID.*

The complexity of checking *BNID* is the same as checking *SBNDC* and, as reported in [10], it is $O(pn2^{3p})$ where p is the number of places and n the number of transitions. Similarly, the complexity of checking *PBNID* is the same as checking *PBNI+* and, as reported in [10], varies from a minimum of $O(f + p)$ (where f is the number of arcs) to a maximum of $O(pn2^{2p})$.

Example 4. Consider again the insecure net in Figure 4. A possible way to make the net secure is to include additional downgrading transitions as reported in Figure 6.

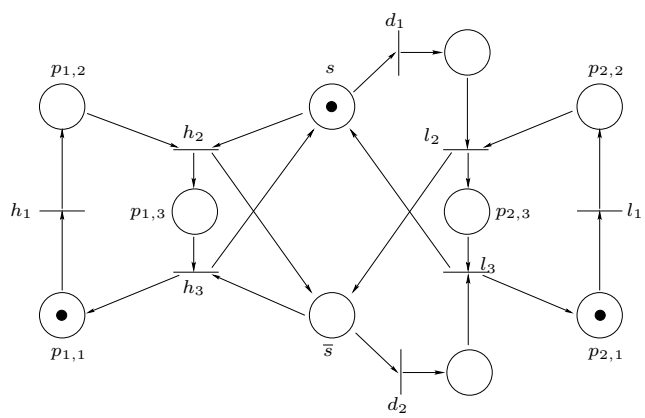

Fig. 6. Secure mutually exclusive access to a shared resource.

6 Conclusion

In this paper we presented a study about the noninterference properties in the basic case of two levels of confidentiality (H and L) and in the richer case of three levels, where the level D of downgrading actions is considered additionally. In both settings, we start by first recalling the definitions by Rushby in [23] for deterministic finite state automata with outputs (Mealy machines); we try to provide similar definitions for the more popular model of deterministic labeled transition system, following as much as possible the original intuition,

yielding our definition of *NI* and *INI*. Then, in both scenarios, we investigate local information flow properties, called *SNDC* and *NID*, respectively, that are necessary and sufficient conditions to ensure *NI* and *INI*, respectively. In both cases, we discuss the need of bisimulation-based definition of information flow security properties as soon as one considers nondeterministic LTSs. Finally, we extend the work in [3, 4] over the model of elementary net systems to the case of intransitive noninterference.

Future work will be devoted to study systematic techniques for adapting an insecure system to make it secure. This is a rather challenging problem because there is a large range of possible modifications of a system that can be considered; e.g., enlarging the behaviour of the low part of the system, inserting suitable declassifiers whenever possible (as we have done in Example 4) or even cutting some possible high level behaviour. Moreover, we plan to study more selective forms on intransitive noninterference where the occurrence of a downgrading action does not reveal all the previously performed high level actions, as it is now prescribed in Rushby's definition (*cf.* Definition 19).

Furthermore, it is interesting to see if the approach presented here can be extended to the richer setting of P/T Petri nets. Some results are reported in a recent paper [1] (actually written after this), where it is shown that *SBNDC* and *BNID* are decidable for finite P/T nets (hence for a class of infinite-state systems), even if it is left open the problem of understanding if (transitive as well as intransitive) noninterference can be equivalently characterized as a relation of causality or conflict among particular high actions and low ones.

Acknowledgements

The authors would like to thank the anonymous referees for helpful comments.

References

1. E. Best, Ph. Darondeau, R. Gorrieri, On the Decidability of Non Interference over Unbounded Petri Nets, in Procs. SecCo 2010, EPTCS 51:16-33, 2010.
2. N. Busi and R. Gorrieri. A Survey on Non-Interference with Petri Nets. *Advanced Course on Petri Nets 2003*, Springer LNCS 3098:328-344, 2004.
3. N. Busi and R. Gorrieri. Positive Non-Interference in Elementary and Trace Nets. Proc. 25th Int.l Conf. on Application and Theory of Petri Nets, Springer LNCS 3099:1-16, 2004.
4. N. Busi and R. Gorrieri. Structural Non-Interference in Elementary and Trace Nets. *Mathematical Structures in Computer Science* 19(6):1065-1090, 2009.
5. A. Bossi, C. Piazza, S. Rossi. Modelling Downgrading in Information Flow Security. Procs. 17th IEEE Computer Security Foundations Workshop (CSFW'04), IEEE Press, 187-201, 2004.
6. J.Engelfriet and G. Rozenberg. Elementary Net Systems, *Lectures on Petri Nets I: Basic Models*, Springer LNCS 1491:12-121, 1998.
7. R. Focardi, R. Gorrieri. A Classification of Security Properties. *Journal of Computer Security* 3(1):5-33, 1995.

8. R. Focardi, 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.
9. R. Focardi, R. Gorrieri. Classification of Security Properties (Part I: Information Flow), *Foundations of Security Analysis and Design* - Tutorial Lectures (R. Focardi and R. Gorrieri, Eds.), Springer LNCS 2171:331-396, 2001.
10. S. Frau, R. Gorrieri, C. Ferigato. Structural Noninterference at Work: the Petri Net Security Checker, in Procs. *Formal Aspects of Security and Trust* (FAST'08), Springer LNCS 5491:210-225, 2009.
11. Nejib Ben Hadj-Alouane, Stphane Lafrance, Feng Lin, John Mullins, Mohamed Moez Yeddes. Characterizing Intransitive Noninterference for 3-Domain Security Policies With Observability. *IEEE Transactions on Automatic Control* 50(6):920-925, 2005.
12. N.B. Hadj-Alouane, S. Lafrance, Feng Lin, J. Mullins, M.M. Yeddes. On the verification of intransitive noninterference in multilevel security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 35(5): 948-958, 2005.
13. J.T. Haigh and W.D. Young. Extending the noninterference version of MLS for SAT. *IEEE Trans. on Software Engineering*, SE-13(2):141150, 1987.
14. J.A. Goguen, J. Meseguer. Security Policy and Security Models. Proc. of Symposium on Security and Privacy (SSP'82), IEEE CS Press, pp. 11-20, 1982.
15. J.A. Goguen, J. Meseguer. Unwinding and Inference Control. Proc. of Symposium on Security and Privacy (SSP'84), IEEE CS Press, pp. 75-86, 1984.
16. S. Lafrance, J. Mullins. Bisimulation-based Nondeterministic Admissible Interference and ita Applications to the Analysis of Cryptographic Protocols. *Information and Software Technology* 45(11): 779-790 (2003) Preliminary version in *Electronic Notes in Theoretical Computer Science* 61:1-24, 2002.
17. Ron van der Meyden. What, Indeed, Is Intransitive Noninterference? Proc. of ESORICS'07, LNCS 4734:235-250, Springer, 2007.
18. R.Milner. *Communication and Concurrency*, Prentice-Hall, 1989.
19. J. Mullins. Nondeterministic Admissible Interference. *Journal of Universal Computer Science* 11:1054-1070, 2000.
20. S. Pinsky. Absorbing Covers and Intransitive Noninterference. Proc. of IEEE Symposium on Security and Privacy (SSP'95), IEEE CS press, pp. 102-113, 1995.
21. S. Pinsky, E. Zieglar. Noninterference Equations for Nondeterministic Systems. Proc. of IEEE Computer Security Foundations Workshop (CSFW'01), IEEE CS press, pp. 3-14, 2001.
22. A.W. Roscoe, M-H. Goldsmith. What is Intransitive Noninterference? Proc. of 12th Computer Security Foundations Workshop (CSFW), IEEE CS Press, pp. 228-238, 1999.
23. J. Rushby. Noninterference, Transitivity, and Channel-control Security Policies. Technical Report CSL-92-02, SRI International, 1992.
24. P.Y.A. Ryan. Mathematical Models of Computer Security, *Foundations of Security Analysis and Design* - Tutorial Lectures (R. Focardi and R. Gorrieri, Eds.), Springer LNCS 2171:1-62, 2001.
25. P.Y.A. Ryan, S. Schneider. Process Algebra and Noninterference, Proc. of 12th Computer Security Foundations Workshop (CSFW), IEEE CS Press, pp. 214-227, 1999.

Appendix

A Proofs of Section 3

Proposition 1. *If a deterministic reduced LTS satisfies NI, then it satisfies SNDC.*

Proof. By hypothesis, we have that for all $\alpha \in E^*$ such that $run(s_0, \alpha)$ is defined, $run(s_0, \alpha) \sim_L run(s_0, hide(\alpha))$. Let $s = run(s_0, \alpha')$ and $t = run(s_0, hide(\alpha'))$; hence, $s \sim_L t$. Take $\alpha = \alpha' b$. If $b \in E_H$ then $s \xrightarrow{b} s'$ and $run(s_0, hide(\alpha' b)) = t$, hence $s' \sim_L t$. Hence, by transitivity of $\sim_L$, we conclude that $s \sim_L s'$, i.e., *SNDC* holds. ■

In order to prove the reverse implication (i.e., *SNDC* implies *NI*), it is useful to prove first the following lemma.

Lemma 1. *If a deterministic reduced LTS satisfies SNDC then, for all $s, t \in St$ and for all $a \in E_L$, $s \sim_L t$ and $s \xrightarrow{a} s'$ imply $t \xrightarrow{a} t'$ and $s' \sim_L t'$.*

Proof. As $s \sim_L t$, for all $\alpha = a\beta$ we have that $\alpha \in \Lambda(Tr(s))$ iff $\alpha \in \Lambda(Tr(t))$. If $a\beta \in \Lambda(Tr(s))$, there must exist a state s' such that $s \xrightarrow{a} s'$ with $\beta \in \Lambda(Tr(s'))$. Similarly, If $a\beta \in \Lambda(Tr(t))$ there must exist a state t' such that $t \xrightarrow{a} t'$ with $\beta \in \Lambda(Tr(t'))$. Hence for all β we have that $\beta \in \Lambda(Tr(s'))$ iff $\beta \in \Lambda(Tr(t'))$, i.e., $s' \sim_L t'$. ■

Proposition 2. *If a deterministic reduced LTS satisfies SNDC, then it satisfies NI.*

Proof. We prove by induction on the length of traces α that

$$s \sim_L t \;\; \Rightarrow \;\; run(s, \alpha) \sim_L run(t, hide(\alpha)) \tag{1}$$

for all α such that $run(s, \alpha)$ is defined. When in the equation (1) above we set $s = t = s_0$, we get the thesis.

The base case when $\alpha = \epsilon$ is trivial. For the inductive case, assume that the thesis holds for all α of length n for which $run(s, \alpha)$ is defined, and consider trace $a\alpha$. If $run(s, a)$ is undefined, then trace $a\alpha$ is to be ignored; otherwise let $s' = run(s, a)$. We have that $run(s, a\alpha) = run(s', \alpha)$. On the other hand for $run(t, hide(a\alpha))$ we have to distinguish two cases.

(1. $a \in E_L$**)** In this case, the definition of *hide* justifies that $run(t, hide(a\alpha)) = run(t, a\, hide(\alpha))$. By hypothesis that $s \sim_L t$ and since $s' = run(s, a)$, there must exist a $t' = run(t, a)$. Hence, $run(t, a\ hide(\alpha)) = run(t', hide(\alpha))$. As $s \sim_L t$ and *SNDC* holds, we get by Lemma 1 that $s' \sim_L t'$. By applying the inductive hypothesis, we get $run(s', \alpha) \sim_L run(t', hide(\alpha))$, and so the implication in (1) holds for $a\alpha$.

(2. $a \in E_H$**)** In this case, the definition of *hide* justifies that $run(t, hide(a\alpha)) = run(t, hide(\alpha))$. On the other hand, as $a \in E_H$ and *SNDC* holds, we get $s \sim_L s'$, hence (since by hypothesis $s \sim_L t$) also $s' \sim_L t$. By applying the inductive hypothesis, we get $run(s', \alpha) \sim_L run(t, hide(\alpha))$, and so the implication in (1) holds for $a\alpha$. ■

Corollary 1. *A deterministic reduced LTS is NI if and only if it is SNDC.*

Proof. It follows directly from Proposition 1 and Proposition 2. ■

B Proofs of Section 4

Proposition 3. *If a deterministic reduced LTS satisfies INI, then it satisfies NID.*

Proof. By hypothesis, we have that for all $\alpha \in E^*$ such that $run(s_0, \alpha)$ is defined, $run(s_0, \alpha) \sim_D run(s_0, ihide(\alpha))$. Let $s = run(s_0, \alpha')$ and $t = run(s_0, hide(\alpha'))$; hence we have that $s \sim_D t$. Take $\alpha = \alpha' b$. If $b \in E_H$ then $s \xrightarrow{b} s'$ and $run(s_0, ihide(\alpha' b)) = t$, hence $s' \sim_D t$. Hence, by transitivity of $\sim_D$, we conclude that $s \sim_D s'$, i.e., *NID* holds. ■

In order to prove the reverse implication (i.e., *NID* implies *INI*), it is useful to prove first the following lemma.

Lemma 2. *If a deterministic reduced LTS satisfies NID then, for all $s, t \in St$ and for all $a \in E_L$, $s \sim_D t$ and $s \xrightarrow{a} s'$ imply $t \xrightarrow{a} t'$ and $s' \sim_D t'$.*

Proof. As $s \sim_D t$, for all $\alpha = a\beta$ we have that $\alpha \in \Delta(Tr(s))$ iff $\alpha \in \Delta(Tr(t))$. If $a\beta \in \Delta(Tr(s))$, there must exist a state s' such that $s \xrightarrow{a} s'$ with $\beta \in \Delta(Tr(s'))$. Similarly, If $a\beta \in \Delta(Tr(t))$ there must exist a state t' such that $t \xrightarrow{a} t'$ with $\beta \in \Delta(Tr(t'))$. Hence for all β we have that $\beta \in \Delta(Tr(s'))$ iff $\beta \in \Delta(Tr(t'))$, i.e., $s' \sim_L t'$. ■

Proposition 4. *If a deterministic reduced LTS satisfies NID, then it satisfies INI.*

Proof. We prove, by induction on the length of α, that

$$\forall \alpha \in E^* \text{ if } run(s_0, \alpha) \text{ is defined, then } run(s_0, \alpha) \sim_D run(s_0, ihide(\alpha)) \quad (2)$$

The base case when $\alpha = \epsilon$ is trivial. For the inductive case, assume that the thesis holds for all α of length n for which $run(s_0, \alpha) = s$ is defined, and consider trace αa. If $run(s, a)$ is undefined, then trace αa is to be ignored; otherwise let $s' = run(s, a)$. We have that $run(s_0, \alpha a) = run(s, a) = s'$. On the other hand for $run(s_0, ihide(\alpha a))$ we have to distinguish three cases.
(1. $a \in E_L$**)** In this case, the definition of *ihide* justifies that $run(s_0, ihide(\alpha a)) = run(s_0, ihide(\alpha)a) = run(t, a)$. The inductive hypothesis ensures that $s \sim_D t$; hence, by Lemma 2 we have that $s' \sim_D t'$, where $t' = run(t, a)$, and so the implication in (2) holds for $a\alpha$.
(2. $a \in E_H$**)** In this case, the definition of *ihide* justifies that $run(s_0, ihide(\alpha a)) = run(s_0, ihide(\alpha)) = t$. On the other hand, $run(s_0, \alpha) = s$ and $run(s, a) = s'$. By inductive hypothesis, $s \sim_D t$. On the other hand, as $a \in E_H$ and *NID* holds, we get $s \sim_D s'$, hence also $s' \sim_D t$, and so the implication in (2) holds for $a\alpha$.
(3. $a \in E_D$**)** In this case, by definition of *ihide*, we get $run(s_0, ihide(\alpha a)) = run(s_0, \alpha a)$. Since $\sim_D$ is reflexive, the thesis follows. ■

Corollary 2. *A deterministic reduced LTS is INI if and only if it is NID.*

Proof. It follows directly from Proposition 3 and Proposition 4. ■

C Proofs of Section 5

Theorem 7. *Let $N = (S, L, D, H, F, m_0)$ be an elementary net system. If N satisfies $PBNID$, then N satisfies $BNID$.*

Proof. Let N be *PBNID*. We will show that N is *BNID*.
Take $m \in [m_0\rangle$ such that $m[h\rangle m'$ for $h \in H$. We have to prove that there exists an initial low-view bisimulation R on N such that $(m, m') \in R$.
Let $R = \{(m_1, m_2) \mid \forall l \in L\ \forall s \in {}^\bullet l : m_1(s) \neq m_2(s) \Rightarrow (\forall \sigma \forall i \in \{1, 2\} : m_i[\sigma l\rangle \Rightarrow \exists l_1 \in \sigma : s \in {l_1}^\bullet)\}$ be the candidate relation.

1. We show that R is an initial low-view bisimulation on N.
   Let $(m_1, m_2) \in R$. Suppose $m_1[l\rangle m_1'$. We show that also $m_2[l\rangle$. Suppose that there exists $s \in {}^\bullet l$ such that $m_2(s) = 0$, hence $m_1(s) \neq m_2(s)$. As $(m_1, m_2) \in R$ and $m_1[l\rangle$, by definition of R (with $\sigma = \epsilon$), there must exists $t \in \epsilon$, reaching a contradiction. Hence $\forall s \in {}^\bullet l\ m_2(s) \geq 1$, and so there exists m_2' such that $m_2[l\rangle m_2'$ (by contact-freeness).
   Now we show that $(m_1', m_2') \in R$. Suppose that $(m_1', m_2') \notin R$. Then there exist l', $s' \in {}^\bullet l'$ such that $m_1'(s') \neq m_2'(s')$ and there exist σ and i such that $m_i'[\sigma l'\rangle$ and $s' \in {l_1}^\bullet$ for no $l_1 \in \sigma$. As $m_i[l\rangle m_i'$ for $i = 1, 2$, $m_1(s') \neq m_2(s')$ and there exists $j \in \{1, 2\}$ such that $m_j[l\sigma l'\rangle$ and $s' \in {l_1}^\bullet$ for no $l_1 \in \sigma$. Seeing that $m_1'(s') \neq m_2'(s')$ necessarily $s' \notin l^\bullet$, hence $s' \in {l_1}^\bullet$ for no $l_1 \in l\sigma$. Thus we obtain $(m_1, m_2) \notin R$, reaching a contradiction. Hence, we have that $(m_1', m_2') \in R$.
   The symmetric case can be proved in the same way, hence we obtain that R is an initial low-view bisimulation on N.
2. We show that $(m, m') \in R$. If there does not exist s and $l \in s^\bullet$ such that $m(s) \neq m'(s)$, we are done: $(m, m') \in R$. Hence, suppose that there exist s and $l \in s^\bullet$ such that $m(s) \neq m'(s)$. We show that $\forall \sigma : m[\sigma l\rangle \Rightarrow \exists t \in \sigma : s \in t^\bullet$ and $\forall \sigma : m'[\sigma l\rangle \Rightarrow \exists t \in \sigma : s \in t^\bullet$. As $m[h\rangle m'$, from $m(s) \neq m'(s)$ we deduce that one of the following holds:
   - $s \in h^\bullet$. Hence s is a potentially causal place.
     Take a sequence σ such $m[\sigma l\rangle$. We show that there exists $t \in \sigma$ such that $s \in t^\bullet$. Two subcases can happen:
     - $\sigma = h\sigma'$. As *PBNID* holds, s is not an active causal place. Hence, for all $\bar{m} \in [m_0\rangle$ and for all $\bar{\sigma}$: if $\bar{m}[h\bar{\sigma} l\rangle$ then there exists $t \in \bar{\sigma}$ such that $s \in t^\bullet$. As $m[\sigma l\rangle$ and $\sigma = h\sigma'$, then there exists $t \in \bar{\sigma}$ such that $s \in t^\bullet$.
     - $\sigma = \epsilon$ or $\sigma = t'\sigma'$ with $t' \neq h$. As $s \in h^\bullet$ we obtain $m(s) = 0$. As $m[\sigma l\rangle$ and $s \in {}^\bullet l$, there must exist a transition $t \in \sigma$ that produces one token in s, i.e., such that $s \in t^\bullet$. In particular, $\sigma \neq \epsilon$.

Consider now a sequence σ such that $m'[\sigma l\rangle$. We show that there exists $t \in \sigma$ such that $s \in t^\bullet$. As $m[h\rangle m'$, we have that $m[h\sigma l\rangle$, hence, because *PBNID* holds, there exists $t \in \sigma$ such that $s \in t^\bullet$.

- $s \in {}^\bullet h$. Hence, s is a potentially conflict place.
  Take a sequence σ such that $m[\sigma l\rangle$. We show that there exists $t \in \sigma$ such that $s \in t^\bullet$.
  As *PBNID* holds s cannot be an active conflict place. Hence, for all $\bar{m} \in [m_0\rangle$ and for all $\bar{\sigma}$: if $\bar{m}[h\rangle$ and $\bar{m}[\bar{\sigma} l\rangle$ then there exists $t \in \bar{\sigma}$ such that $s \in t^\bullet$.
  As $m[h\rangle m'$ and $m[\sigma l\rangle$, there exists $t \in \sigma$ such that $s \in t^\bullet$.
  Take now a sequence σ such that $m'[\sigma l\rangle$. As $s \in {}^\bullet h$, we have $m'(s) = 0$. As $s \in {}^\bullet l$ from $m'[\sigma l\rangle$ we obtain that there must exists a transition $t \in \sigma$ producing one token in s, i.e., $s \in t^\bullet$.

Theorem 8. *Let $N = (S, L, D, H, F, m_0)$ be an elementary net system. If N is $BNID$ then N is $PBNID$.*

Proof. Suppose that N is $BNID$. We show that no place in N can be an active causal place or an active conflict place.

- Suppose that s is an active causal place. Then, there exist $h \in {}^\bullet s$, $l \in s^\bullet$, $m \in [m_0\rangle$ and $\sigma \in (H \cup L)^*$ such that $m[h\sigma l\rangle$ and $\forall t \in \sigma$, $s \notin t^\bullet$.
  Among the markings and the transition sequences that satisfy the conditions above, take m and σ such that σ contains the minimun number of transitions in H.
  Two cases can happen:
  1. All transitions in σ belong to L. We have that $m[h\rangle m'$. By *BNID* there exists an initial low-view bisimulation on N containing the pair (m, m'). As $m'[\sigma l\rangle$, also $m[\sigma l\rangle$. But from $h \in {}^\bullet s$ and $m[h\rangle$ we deduce that $s \notin m$; we also know that $\forall t \in \sigma$, $s \notin t^\bullet$; hence, after the firing of σ place s is still empty, contradicting the fact that $m[\sigma l\rangle$.
  2. There exists a high level transition in σ. Let h' be the last high transition in σ. Hence, there exist σ_1, σ_2, such that $\sigma = \sigma_1 h' \sigma_2$ and all transitions in σ_2 belong to L. Thus, there exist m_1, m_2 such that $m[h\sigma_1\rangle m_1[h'\rangle m_2[\sigma_2 l\rangle$. From $m_1[h'\rangle m_2$, by *BNID* there exists an initial low-view bisimulation on N containing the pair (m_1, m_2). From $m_2[\sigma_2 l\rangle$, we obtain that also $m_1[\sigma_2 l\rangle$, thus obtaining the firing sequence $m[h\sigma_1\sigma_2 l\rangle$, contradicting the fact that the chosen transition sequence was one with the least number of high transitions.
- Suppose that s is an active conflict place. There there exist $h \in s^\bullet$, $l \in s^\bullet$, $m \in [m_0\rangle$ and $\sigma \in (H \cup L)^*$ such that $m[h\rangle$, $m[\sigma l\rangle$ and $\forall t \in \sigma$, $s \notin t^\bullet$.
  Among the markings and the transition sequences that satisfy the conditions above, take m and σ such that σ contains the minimun number of transitions in H.
  Two cases can happen:
  1. All transitions in σ belong to L. We have that $m[h\rangle m'$. By *BNID* there exists an initial low-view bisimulation on N containing the pair (m, m').

As $m[\sigma l\rangle$, then $m'[\sigma l\rangle$. But from $h \in s^\bullet$ and $m[h\rangle$ we deduce that $s \notin m'$, we also know that $\forall t \in \sigma$, $s \notin t^\bullet$; hence, after the firing of σ place s is still empty, contradicting the fact that $m'[\sigma l\rangle$.

2. There exists a high level transition in σ. Let h' be the last high transition in σ. Hence, there exist σ_1, σ_2, such that $\sigma = \sigma_1 h' \sigma_2$ and all transitions in σ_2 belong to L. Thus, there exist m_1, m_2 such that $m[\sigma_1\rangle m_1[h'\rangle m_2[\sigma_2 l\rangle$. From $m_1[h'\rangle m_2$, by *BNID* there exists an initial low-view bisimulation on N containing the pair (m_1, m_2). From $m_2[\sigma_2 l\rangle$, we obtain that also $m_1[\sigma_2 l\rangle$, thus obtaining the firing sequence $m[\sigma_1\sigma_2 l\rangle$, contradicting the fact that the chosen transition sequence was one with the least number of high transitions.