Deciding Selective Declassification of Petri Nets

Eike Best[1] and Philippe Darondeau[2]

[1] Parallel Systems, Department of Computing Science
Carl von Ossietzky Universität Oldenburg, D-26111 Oldenburg, Germany
eike.best@informatik.uni-oldenburg.de
[2] INRIA, Centre Rennes - Bretagne Atlantique
Campus de Beaulieu, F-35042 Rennes Cedex
Philippe.Darondeau@inria.fr

Abstract. This paper considers declassification, as effected by *downgrading* actions D, in the context of *intransitive non-interference* encountered in systems that consist of *high-level* (secret) actions H and *low-level* (public) actions L. In a previous paper, we have shown the decidability of a strong form of declassification, by which D contains only a single action $d \in D$ declassifying all H actions at once. The present paper continues this study by considering *selective declassification*, where each transition $d \in D$ can declassify a subset $H(d)$ of H. The decidability of this more flexible, application-prone declassification framework is proved in the context of (possibly unbounded) Petri nets with possibly infinite state spaces.

1 Introduction

This work has been inspired by papers by Gorrieri et al., especially [2,5], which contain structurally defined security properties for Petri nets describing systems with high-level (secret) and low-level (public) transitions. In particular, the NDC property (non-distinguishability with respect to composition [2]) defines security in terms of parallel compositions with (almost) arbitrary other systems. While this is an intuitively appealing concept, it is desirable, for formally handling it, to obtain a characterisation based on transition systems. In [1], we provided such a characterisation and we used it in order to prove the decidability of NDC for (possibly unbounded) Petri nets. Moreover, we extended this investigation to the property INI (intransitive non-interference) defined in [5], which generalises the NDC property to systems exhibiting downgrading actions in addition to secret and public ones. The idea is that downgrading actions *declassify* (i.e., turn public) previously executed secret activity. In [1], the decidability of such a property was obtained as well.

The downgrading technique defined in [5,1] appears to be quite coarse, however, in the sense that a single downgrading action declassifies the entire set of secret actions. As suggested e.g. in [5], *selective* downgrading is more likely to be of practical interest. In this approach, declassification can be targeted selectively towards subsets of high-level actions (not necessarily the whole set). Thus, a subset of high-level actions may be associated with every downgrading action, and

P. Degano and J.D. Guttman (Eds.): POST 2012, LNCS 7215, pp. 290–308, 2012.

those subsets may differ from one downgrading action to the next. [1] contains neither a definition of such systems, nor any investigation of their decidability. These gaps are closed in the present paper.

2 Basic Definitions and Decidability Results

A *PT-net* is a triple $N = (P, T, F)$, where P and T are *finite* disjoint sets of vertices, called *places* and *transitions*, respectively, and $F\colon (P\times T)\cup(T\times P) \to \mathbb{N}$ is a set of directed edges with non-negative integer weights. A place p is said to be in *ordinary self-loop* with a transition t if $F(p,t) = 1$ and $F(t,p) = 1$.

A *marking* of N is a map $M\colon P \to \mathbb{N}$. A transition $t \in T$ is *enabled at* a marking M (notation: $M[t\rangle$) if $M(p) \geq F(p,t)$ for all places $p \in P$. If t is enabled at M, then it can *be fired*, leading to the new marking M' (notation: $M[t\rangle M'$) defined by $M'(p) = M(p) + F(t,p) - F(p,t)$ for all $p \in P$. These definitions are extended inductively to transition sequences $\sigma \in T^*$: for the empty sequence ε, $M[\varepsilon\rangle$ and $M[\varepsilon\rangle M$ are always true; for a non-empty sequence σt with $t \in T$, $M[\sigma t\rangle$ (or $M[\sigma t\rangle M'$) iff $M[\sigma\rangle M''$ and $M''[t\rangle$ (or $M''[t\rangle M'$, respectively) for some M''. A marking M' is *reachable* from a marking M if $M[\sigma\rangle M'$ for some $\sigma \in T^*$. The set of markings reachable from M is denoted by $[M\rangle$.

Let V be any alphabet. For words $w, v \in V^*$ and a set of letters $U \subseteq V$, let $w \sim_U v$ denote the fact that the projection of w onto letters of U equals the projection of v onto letters of U. A (transition-) labelling of a net (P, T, F) is a function $\lambda\colon T \to V \,\dot\cup\, \{\varepsilon\}$. Transitions t with $\lambda(t) \in V$ are called *visible*, while those with $\lambda(t) = \varepsilon$ are called *invisible*. The function λ can be extended to $\lambda\colon T^* \to V^*$ as follows: $\lambda(\varepsilon) = \varepsilon$ and $\lambda(\sigma t) = \lambda(\sigma)\lambda(t)$. The label ε here plays the role of the empty word in V^*.

In later parts of the paper, several behavioral notions will be investigated. Such notions usually depend both on some marking and on the labelling of a net. As it is normally clear from the context which marking and which labelling are meant, we will often take the liberty of denoting by N variously some net (P, T, F) or some marked net (P, T, F, M_0) or some marked labelled net (P, T, F, M_0, λ).

Let N denote a net with initial marking M_0 and labelling λ. The (prefix) language of N is the set of words

$$L(N) \;=\; \{w \in V^* \mid \exists \text{ firing sequence } M_0[\sigma\rangle \text{ such that } \lambda(\sigma) = w\}.$$

Two (initially marked and labelled) nets N_1, N_2 are called *language-equivalent* iff $L(N_1) = L(N_2)$. A labelling λ is called *plain* if $\lambda(t) \in V$ for all t, i.e., all transitions are visible. A labelling λ is called *injective* if $\lambda(t_1) = \lambda(t_2) \in V$ implies $t_1 = t_2$. In an injectively labelled net, the labels of the visible transitions can be identified with these transitions, and the rest of the transitions are invisible. In the main part of this paper, we will be concerned exclusively with injective labellings, so that it will be sufficient to designate the subset of transitions that are to be considered as visible. In particular, we can then just use the terminology "N_1 and N_2 are language-equivalent with respect to some set of transitions V".

We shall need the following known decidability results. Their proofs and original sources can be found in [11].

1. Given a net N and a place s. It is decidable whether there exists a reachable marking M with $M(s) > 0$.
2. Given two labelled nets N_1 and N_2 where N_2 is plainly and injectively labelled. It is decidable whether $L(N_1) \subseteq L(N_2)$. A stronger statement is the following:
3. Given two labelled nets N_1 and N_2 where N_2 is plainly and injectively labelled, and a regular language K. It is decidable whether $(L(N_1) \cap K) \subseteq L(N_2)$.

3 Noninterference Properties, and Previous Results

An (injectively labelled) net is called *HL* if its set of transitions T is partitioned as $T = H \dot{\cup} L$ where transitions in H are invisible (i.e., ε-labelled) and are called *high-level* transitions, while transitions in L are visible (i.e., every $t \in L$ is labelled by t, its own name) and are called *low-level* transitions. An *HL*-net is called H-net (L-net) if it only has high-level (low-level, respectively) transitions, i.e., if $T = H$ (respectively, $T = L$). Let $U \subseteq T$ be a set of transitions of N. Then $N \setminus U$ is the net obtained from N by erasing all transitions U and their surrounding arrows. Suppose that N and N' have disjoint sets of places, but not necessarily disjoint sets of transitions. Then $N|N'$ is the net obtained from N and N' by identifying (or "merging") their common transitions.

The next definition originates from [2], and it was the starting point of our investigations.

Definition 1. Non-deducibility on compositions (NDC)

Let N be an (initially marked) *HL*-net with $T = H \dot{\cup} L$. N has property NDC iff $N \setminus H$ and $(N|N') \setminus (H \setminus H')$ are language-equivalent, for any H-net N' whose place set is disjoint from that of N and whose set of transitions $T' = H'$ is disjoint from L. □

In [1], the NDC property was characterised as follows:

Theorem 1. Characterisation of the NDC property

A net N is NDC if and only if N and $N \setminus H$ are language-equivalent (with respect to the set L of visible transitions). □

An (injectively labelled) net is called *HLD* if its set of transitions T is partitioned as $T = H \dot{\cup} L \dot{\cup} D$, where H and L denote the set of high-level and low-level transitions, respectively, and D denotes the set of downgrading (or declassifying) transitions. Transitions in H are considered invisible while transitions in $L \cup D$ are considered visible.

The next definition stems from [5].

Definition 2. Intransitive non-interference (INI)

Let N be an *HLD*-net with $T = H \uplus L \uplus D$ and with initial marking M_0. N has property INI iff $(N \setminus D, M)$ has the property NDC for $M = M_0$ and for any marking M such that $M_0[vd\rangle M$ in N for some sequence $v \in T^*$ and some downgrading transition $d \in D$. □

This captures the idea that as soon as some d occurs, all the previously invisible (secret) H actions become visible (i.e., no longer secret). In [1], the following facts were proved about Property INI. (Actually, Theorem 2 follows directly from Theorem 1.)

Theorem 2. Characterisation of the INI property

An HLD-net (N, M_0) is INI if and only if $(N \setminus D, M)$ and $((N \setminus (H \cup D), M)$ are language-equivalent for $M = M_0$ and for any marking M such that $M_0[vd\rangle M$ in N for some sequence $v \in T^*$ and some downgrading transition $d \in D$. □

Theorem 3. Decidability of the INI property

Given an HLD-net N, it is decidable whether N has Property INI. □

4 Selective Non-interference

Next, Definition 2 will be refined in order to account for selective declassification. Assume, to that end, that every transition $d \in D$ has some associated set of high-level transitions $H(d) \subseteq H$, and denote such nets as *sd* (for *selective declassification*). The idea is that an occurrence of d declassifies all previously executed transitions in $H(d)$, but no other high-level transitions.

Definition 3. INI with selective declassification (INISD)

An *sd-HLD*-net N with $T = H \uplus L \uplus D$ and with initial marking M_0 has property INISD iff for any firing sequence of the form:

$$\begin{array}{l} M_0[w_0 d_1 w_1 d_2 \ldots d_{n-1} w_{n-1} d_n w_n\rangle, \\ \text{where } d_1, \ldots, d_n \in D; \\ \text{for all } j \in \{1, \ldots, n\},\ d_j \text{ does not occur in } w_j d_{j+1} \ldots d_n w_n; \\ \text{and} \{d_1, \ldots, d_n\} \text{ is the set of all declassifying} \\ \quad \text{actions occurring in the sequence} \end{array} \tag{1}$$

there exists a corresponding firing sequence of the form

$$M_0[v_0 d_1 v_1 d_2 \ldots d_{n-1} v_{n-1} d_n v_n\rangle \tag{2}$$

with similar properties, such that for every $i \in \{0, \ldots, n\}$:

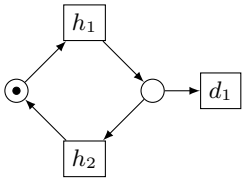
$$v_i \in L_i^* \text{ and } w_i \sim_{L_i} v_i\ , \quad \text{where } L_i = L \cup D \cup (\bigcup_{i<k\leq n} H(d_k)).$$

□

According to the conditions in (1), the *last* occurrence of any declassifying action d_k inside the sequence

$$w \;=\; w_0 d_1 w_1 d_2 \ldots d_{n-1} w_{n-1} d_n w_n$$

is singled out as being critical. Before that last occurrence, all occurrences of transitions in $H(d_k)$ are declassified by this d_k. After that last occurrence, they may either be declassified by some other d action, or still be secret. For this reason, $H(d_k)$ is included in L_i, for every $i < k$. The definition stipulates that the sequences v_i required to exist by (2) must be the projections of w_i onto L_i. Transitions left out of these projections are just those H-transitions which are *not* declassified later in w. In order to understand this definition better, it is perhaps helpful to scan w mentally from right to left; once d_k occurs in such a scan, no *previously* occurring $H(d_k)$-transition is to be considered as secret.



$$\begin{aligned} H &= \{h_1, h_2\} \\ L &= \emptyset \\ D &= \{d_1\} \\ H(d_1) &= \{h_1\} \end{aligned}$$

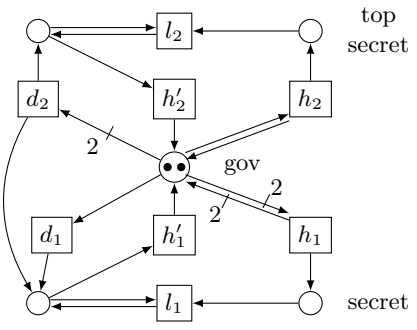
Fig. 1. An *sd-HLD*-Petri net which is not INISD

For the sake of illustration, consider the net shown in Figure 1 and the sequence

$$M_0[\underbrace{h_1 h_2 h_1}_{w_0} d_1\rangle.$$

We have $L_0 = \{d_1, h_1\}$ and $L_1 = \{d_1\}$. If we wanted to prove the INISD property, we would need to find a sequence $M_0[v_0 d_1\rangle$ such that $v_0 \in L_0^*$ and $w_0 \sim_{L_0} v_0$. As the projection of w_0 onto L_0 is $h_1 h_1$, we would thus need to check whether $h_1 h_1 d_1$ is firable. However, since this is not the case, the net does not have the INISD property. Informally, by "seeing $h_1 h_1 d_1$", an observer can sense something secret, namely that some non-declassified high-level action (in this case, h_2) must have occurred.

Consider Figure 2 for a slightly more involved example. Some government (place "gov") is engaged in two types of diplomatic activity, secret ones (transitions h_1, h_1') and top secret ones (transitions h_2, h_2'), producing an unknown (and unlimited) amount of documents on the corresponding places. At any point, it may be decided to declassify some information. This is done either by declassifying only secret information, without also declassifying top secret information (transition d_1), or by declassifying all informations, be they top secret or just



$H(d_2) = \{h_1, h'_1, h_2, h'_2\}$ and $H(d_1) = \{h_1, h'_1\}$

Fig. 2. Another non-INISD Petri net

secret (transition d_2). After d_1-declassification, visible secret-reading activity (transition l_1) may be started, or, alternatively, secret activity may be restarted by h'_1, which at the same time disables l_1. After d_2-declassification, both types of reading (l_1 and l_2) may be started, or, alternatively, disabled by their corresponding h'_1 or h'_2 actions (i.e.: restart of secret and top secret activity, respectively). Note that more h_1-secrets may only be produced if *both* restart actions h'_1 and h'_2 have occurred.

A sequence not disproving the INISD property. Consider the first line of (3):

$$\begin{aligned} w &= \underbrace{h_1 h_1 h_2 d_1 h_2 l_1 h'_1 h_1}_{w_0} d_2 \underbrace{l_2 h'_2 l_1 h'_1 h_2}_{w_1} d_1 \underbrace{l_1}_{w_2} \\ v &= \underbrace{h_1 h_1 h_2 d_1 h_2 l_1 h'_1 h_1}_{v_0} d_2 \underbrace{l_2 l_1 h'_1}_{v_1} d_1 \underbrace{l_1}_{v_2} \end{aligned} \tag{3}$$

Then w is of the form (1), and we compute $L_0 = \{l_1, l_2, d_1, d_2, h_1, h'_1, h_2, h'_2\}$, $L_1 = \{l_1, l_2, d_1, d_2, h_1, h'_1\}$, and $L_2 = \{l_1, l_2, d_1, d_2\}$. A corresponding v of the form (2), which is firable, is shown in the second line of (3). Note that the declassifying transition d_1 may be fired twice in a row, in which case the transition h'_1 needs to be fired also twice in a row to actually let secret activity be restarted.

A sequence disproving the INISD property. Consider $w' = d_2 h'_2 d_1$. We compute the same sets L_0, L_1, L_2 as for w. If the net were INISD, there would exist a firable sequence $v' = v'_0 d_2 v'_1 d_1 v'_2$ satisfying (2) of Definition 3, where v'_1 is the projection of h'_2 onto L_1, that is, $v'_1 = \varepsilon$. However, such a sequence does not exist. Hence, the net does not satisfy the INISD property. This problem hinges on the fact that some "visible" lower level activity (d_1-declassification) is made partially dependent on some top secret activity (the h'_2 event), and therefore, the latter is detectable when it should not be.

Definition 3 has been adopted as an unostentatious extension of Definition 2. In particular, there are the following special cases. If $D = \emptyset$, then in view of

Theorem 1, Definition 3 amounts to Definition 1. If $H(d) = H$ for every $d \in D$, then Definition 3 reduces to Definition 2. When comparing the INISD property to other properties of information control flow, we did not find any equivalent one, but several related ones. This will be discussed in more detail in Section 8.

5 Decidability of the INISD Property

As properties of systems described by *sd-HLD* Petri nets seem to be quite sensitive to design decisions, it would be nice if one could use an algorithm to check automatically whether a system satisfies the INISD property or not. Next, it will be shown that such an algorithm exists.

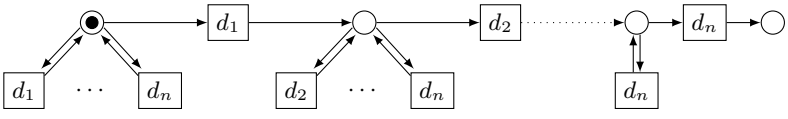


Fig. 3. The net $N[d_1 \dots d_n]$

Theorem 4. Property INISD is decidable

Given an sd-HDL-net N, it is decidable whether N has Property INISD.

Proof: First note that the $d_1, \dots, d_n$ in Definition 3 are mutually distinct, because every one of them is the *last* of its kind in the firing sequence considered in (1). Therefore, the set Σ of sequences $d_1 \dots d_n \in D^*$ such that some firing sequence $M_0[w_0 d_1 w_1 d_2 \dots d_n w_n\rangle$ exists and satisfies the conditions stated in (1) is finite, since D is finite and there are only finitely many repetition-free sequences over D. For repetition-free sequences $d_1 \dots d_n$, membership in Σ can be decided effectively by the following algorithm:

- Let $d_1 \dots d_n \in D^*$ with $\forall 1 \leq i, j \leq n\colon i \neq j \Rightarrow d_i \neq d_j$ be given.
- Consider the net $N[d_1 \dots d_n]$ depicted in Figure 3.
- For each $d_i \in \{d_1, \dots, d_n\}$, replicate the *unique* transition d_i of N as many times as needed to obtain the same number of transitions d_i in N and in $N[d_1 \dots d_n]$ (all replicas have similar flow relations).
- For each $d_i \in \{d_1, \dots, d_n\}$, glue the d_i-transitions of N and $N[d_1 \dots d_n]$ pairwise.
- If a marking can be reached such that the final place of $N[d_1 \dots d_n]$ carries a token, then $d_1 \dots d_n$ belongs to Σ, otherwise it does not.

This implies by the results recalled in section 2 that the finite set Σ can be effectively constructed. Note the special case $n = 0$ (hence $d_1 \dots d_n = \varepsilon$). The final place in $N[\varepsilon]$ equals the initial place, which is marked initially. Hence the empty sequence is trivially checked by the above algorithm as belonging to Σ (as indeed it should, by the definition of Σ).

Next, consider some fixed sequence $d_1 \dots d_n \in \Sigma$. We will define two Petri nets, $N_1[d_1 \dots d_n]$ and $N_2[d_1 \dots d_n]$. Both nets are derivatives from N, the given *HDL*-net. We shall use these pairs of nets (one pair for each sequence $d_1 \dots d_n \in \Sigma$) in order to reduce the INISD property to Petri net language inclusion.

The idea of this construction is as follows. L-actions, which are visible, will simply always be left intact. D-actions d will be duplicated into a tilde-adorned variant $\widetilde{d}$, which will denote the *last* occurrence of d, and plain d, which will denote all other occurrences. Also, H-actions h will be duplicated into $\widetilde{h}$ and plain h. The latter are invisible as before, while the former will denote declassified actions, which are made visible by being declassified.

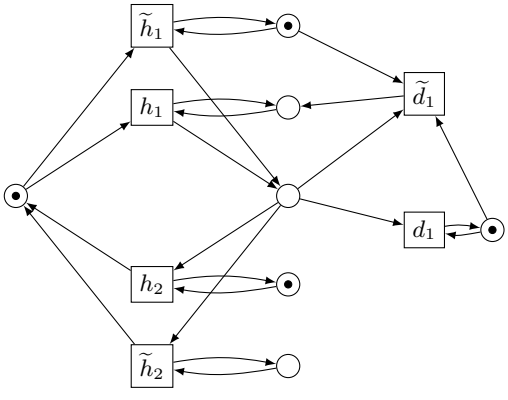
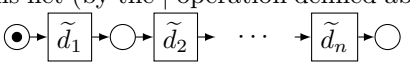


Fig. 4. $N_1[d_1]$, for the net shown in Figure 1

Definition of $N_1[d_1 \dots d_n]$:

1. Copy all the places of N into $N_1[d_1 \dots d_n]$.
2. Copy all L-transitions (with the same flow relations) from N to $N_1[d_1 \dots d_n]$.
3. Copy all H-transitions (with the same flow relations) from N to $N_1[d_1 \dots d_n]$. From each $h \in H$, make a copy $\widetilde{h}$ with the same flow relations as h.
4. Copy all transitions $\{d_1, \dots, d_n\} \subseteq D$ from N to $N_1[d_1 \dots d_n]$ and for every d_j $(1 \leq j \leq n)$, make a copy $\widetilde{d}_j$ with the same flow relations as d_j.
5. For every h and $\widetilde{h}$, add control places p_h and $p_{\widetilde{h}}$ in ordinary self-loop with h and $\widetilde{h}$, respectively. Put one token on p_h iff $h \notin \bigcup_{1 \leq j \leq n} H(d_j)$, and one token on $p_{\widetilde{h}}$ iff $h \in \bigcup_{1 \leq j \leq n} H(d_j)$.
6. For every d_j, add a one-token control place q_j in ordinary self-loop with d_j.
7. Add flow relations with the following effect: $\widetilde{d}_j$ disables d_j by emptying q_j; furthermore, if h is in the set $H(d_j) \setminus \bigcup_{j<k\leq n} H(d_k)$ then $\widetilde{d}_j$ disables transition $\widetilde{h}$ by emptying $p_{\widetilde{h}}$ and enables transition h by filling p_h.
8. Synchronise this net (by the $|$ operation defined above) with the net $\widetilde{N}[d_1 \dots d_n]$ as follows: 

The set of visible transitions of $N_1[d_1 \dots d_n]$ are defined as

$$X \;=\; L \cup \{d_1, \dots, d_n\} \cup \{\widetilde{d}_1, \dots, \widetilde{d}_n\} \cup \{\widetilde{h} \mid h \in H\},$$

i.e., all transitions except those in H. The result of this construction, applied to the net shown in Figure 1 and with the sequence d_1, is shown in Figure 4.

Definition of $N_2[d_1 \dots d_n]$:

1. Create $N_1[d_1 \dots d_n]$.
2. Delete all transitions h and all places p_h for $h \in H$, as well as their surrounding arcs.

All transitions of $N_2[d_1 \dots d_n]$ are defined to be visible. (This set of transitions happens to be the same set X as above.) The result of this construction, applied to the net shown in Figure 1 and with the sequence d_1, is shown in Figure 5.

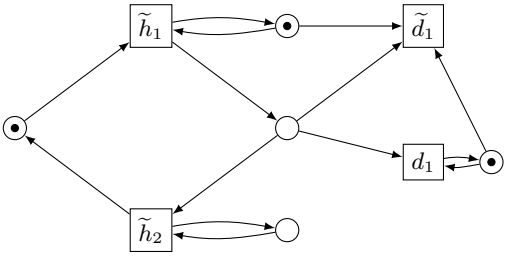


Fig. 5. $N_2[d_1]$, for the net shown in Figure 1

Note that because of the requirements formulated in Definition 3, in (1), there can be no D-action whatsoever in w_n, at most action d_n in w_{n-1}, at most actions d_n and d_{n-1} in w_{n-2}, and so on, until in w_0 there could be any number of actions from $\{d_1, \dots, d_n\}$ but no other D-actions. It is with a view to this that, in both nets $N_1[d_1 \dots d_n]$ and $N_2[d_1 \dots d_n]$, $\widetilde{d}_j$ disables d_j and actions in $D \setminus \{d_1, \dots, d_n\}$ do not appear.

To finish the proof, we show that the following two statements are equivalent:

A : N has the INISD property.

B : $L(N_1[\varepsilon]) \;\subseteq\; L(N_2[\varepsilon])$,

and for every $\sigma = d_1 \dots d_n \in \Sigma$ with $n \geq 1$:

$L(N_1[d_1 \dots d_n]) \cap (X^* \widetilde{d}_n L^*) \;\subseteq\; L(N_2[d_1 \dots d_n])$

where X is the set of visible transitions of $N_1[d_1 \dots d_n]$.

The claim of the theorem follows, because **B** can be checked effectively by the results recalled in section 2. Note, to this end, that $X^* \widetilde{d}_n L^*$ is a regular language.

(**A** $\Rightarrow$ **B**): Assume that N is INISD. Let $\sigma = d_1 \dots d_n$ be any sequence from Σ. We distinguish two cases, $n = 0$ and $n \geq 1$.

Case $n = 0$: We prove $L(N_1[\varepsilon]) \;\subseteq\; L(N_2[\varepsilon])$.

Let $x \in L(N_1[\varepsilon])$. Then there is some sequence τ, firable in $N_1[\varepsilon]$, such that x is the projection of τ on X^*, with $X = L \cup \{\widetilde{h} \mid h \in H\}$. In particular, x does not

contain any downgrading transitions. By the construction of $N_1[\varepsilon]$, τ does not contain any transition d or $\widetilde{d}$ with $d \in D$, nor any $\widetilde{h}$-transitions. So, the sequence τ is also firable in N, and moreover, $x \in L^*$. Setting $w = w_0 = w_n = \tau$, w is of the form (1). By the INISD property, there exists some $v = v_0$, firable in N, such that v_0 is the projection of w_0 onto $L \cup D$, hence onto L. By the construction of $N_2[\varepsilon]$, v_0 is firable in $N_2[\varepsilon]$ as well. But $v_0 = x$, because both x and v_0 are the projection on L^* of $w_0 = \tau \in (H \cup L)^*$. Since v_0 is firable in $N_2[\varepsilon]$, so is x. This implies that $x \in L(N_2[\varepsilon])$, and since x was arbitrary, we get a proof of $L(N_1[\varepsilon]) \subseteq L(N_2[\varepsilon])$.

Case $n \geq 1$: We prove $L(N_1[d_1 \dots d_n]) \cap (X^* \widetilde{d}_n L^*) \subseteq L(N_2[d_1 \dots d_n])$.

Let $x \in L(N_1[d_1 \dots d_n]) \cap (X^* \widetilde{d}_n L^*)$. Then there is some sequence τ, firable in $L(N_1[d_1 \dots d_n])$, such that x is the projection of τ on X^*, and moreover, x contains $\widetilde{d}_n$ at some point and only transitions from L after the last $\widetilde{d}_n$. By the construction of $N_1[d_1 \dots d_n]$ (more precisely, item 8. in that construction), τ contains all of $\widetilde{d}_1, \dots, \widetilde{d}_n$, in that order, and each $\widetilde{d}_i$ exactly once. Therefore, τ can be split as follows:

$$\tau \;=\; \tau_0 \widetilde{d}_1 \tau_1 \widetilde{d}_2 \dots \widetilde{d}_{n-1} \tau_{n-1} \widetilde{d}_n \tau_n$$

such that τ_n contains no d nor any $\widetilde{d}$ with $d \in D$, τ_{n-1} contains at most some d_n transitions, and so on, thus providing a sequence of the form (1).

In going from τ to x by projecting τ on X^*, at most some high-level transitions h *without tilde* are erased. By item 7. of the construction of $N_1[d_1 \dots d_n]$, such high-level transitions h may occur *only after* the last declassifying $\widetilde{d}_j$ for which $h \in H(d_j)$ holds. Before such a $\widetilde{d}_j$, high-level transitions $h \in H(d_j)$ can only occur (if at all) in the form $\widetilde{h}$. Let $w = \mathit{plain}(\tau)$ be the sequence obtained from τ by removing the tildes from all actions $\widetilde{h}$ and $\widetilde{d}$, but leaving the sequence unchanged otherwise. Then by the construction of $N_1[d_1 \dots d_n]$ (as it essentially - disregarding the tildes - does not add any behaviour to N), w is firable in N. Taking account of the splitting of τ, let $w_i = \mathit{plain}(\tau_i)$, for $0 \leq i \leq n$, and hence

$$w \;=\; w_0 d_1 w_1 d_2 \dots d_{n-1} w_{n-1} d_n w_n.$$

By the properties just explained, w conforms to the requirements associated with (1). Therefore, by the INISD property, a sequence v conforming to (2) exists. More precisely, there exists

$$v \;=\; v_0 d_1 v_1 d_2 \dots d_{n-1} v_{n-1} d_n v_n$$

such that v is firable in N and every v_i arises from w_i by erasing (only) those high-level transitions h that do not belong to $\bigcup_{i<k\leq n} H(d_k)$. Let τ' be the sequence obtained from v by putting back tildes on all remaining high-level transitions and on the last occurrence of each declassifying action $d \in D$. Then τ' is firable in $N_2[d_1 \dots d_n]$, because the firing in N of any high-level transition h occurring in v_i is faithfully simulated by the firing of the corresponding $\widetilde{h}$ in $N_2[d_1 \dots d_n]$.

This is so because, for any such high-level transition h, necessarily, $h \in H(d_j)$ for some $j > i$, and by item 8. in the construction of $N_2[d_1 \dots d_n]$, the transition $\widetilde{d_j}$ cannot have been fired earlier.

Now by the above construction, τ' is the same as x. This shows that x is firable in $N_2[d_1 \dots d_n]$, ending the proof of $(\mathbf{A} \Rightarrow \mathbf{B})$.

$(\mathbf{B} \Rightarrow \mathbf{A})$: In order to prove the INISD property from $(\mathbf{B})$, let

$$w \;=\; w_0 d_1 w_1 d_2 \dots d_{n-1} w_{n-1} d_n w_n$$

be any sequence, firable in N and satisfying the conditions stated in (1). We show that a related sequence $v = v_0 d_1 v_1 d_2 \dots d_{n-1} v_{n-1} d_n v_n$ satisfying (2) exists such that $v_i \in L_i^*$ and $w_i \sim_{L_i} v_i$ for all i. Again, we distinguish the cases $n = 0$ and $n \geq 1$.

Case $n = 0$: Then $w = w_0 = w_n$, and no D action occurs in w. By construction of $N_1[\varepsilon]$, w is firable in $N_1[\varepsilon]$. (Note that in $N_1[\varepsilon]$, a token is on p_h but not on $p_{\widetilde{h}}$.) Let v be the projection of w onto X (which by the above, is the same as the projection of w onto L). By $L(N_1[\varepsilon]) \subseteq L(N_2[\varepsilon])$, coming from $(\mathbf{B})$, and since all transitions of $N_2[\varepsilon]$ are visible, v is firable in $N_2[\varepsilon]$. But since the L-firing sequences of $N_2[\varepsilon]$ agree with those of N, v is also firable in N. So, (2) is satisfied with $v_0 = v$.

Case $n \geq 1$: By the construction of $N_1[d_1 \dots d_n]$, and because w is firable in N, the sequence

$$\widetilde{w}_0 \widetilde{d_1} \widetilde{w}_1 \widetilde{d_2} \;\dots\; \widetilde{d_n} \widetilde{w}_n$$

is firable in $N_1[d_1 \dots d_n]$, where for $0 \leq i \leq n$, $\widetilde{w}_i$ is as w_i, except that every $h \in \bigcup_{i<k\leq n} H(d_k)$ (in w_i) is replaced by $\widetilde{h}$.

For every $0 \leq i \leq n$, let $\widetilde{v}_i$ be the projection of $\widetilde{w}_i$ onto X. By $(\mathbf{B})$,

$$\widetilde{v} \;=\; \widetilde{v}_0 \widetilde{d_1} \widetilde{v}_1 \widetilde{d_2} \;\dots\; \widetilde{d_n} \widetilde{v}_n$$

is firable in $N_2[d_1 \dots d_n]$. Let v_i be the same as $\widetilde{v}_i$, where each $\widetilde{h}$ is replaced by h. By the constructions of $N_1[d_1 \dots d_n]$, $\widetilde{v}_i$ and v_i, no $h \in H(d_i)$ occurs in the sequence $v_i d_{i+1} \dots d_n v_n$. Hence, by the construction of $N_2[d_1 \dots d_n]$,

$$v \;=\; v_0 d_1 v_1 d_2 \dots d_n v_n$$

which is a firing sequence of N, satisfies the requirements of (2). □

As an illustration of this theorem, let us apply $(\mathbf{B})$ to prove that the net shown in Figure 1 is not INISD. Consider

$$\tau \;=\; \widetilde{h}_1 h_2 \widetilde{h}_1 \widetilde{d}_1 \quad \text{in } N_1[d_1] \text{ (see Figure 4).}$$

The projection of τ on X^* is $\sigma = \widetilde{h}_1 \widetilde{h}_1 \widetilde{d}_1$. By comparing Figures 4 and 5, one can see that σ is in $L(N_1[d_1]) \cap (X^* \widetilde{d}_1 L^*)$ but not in $L(N_2[d_1])$. This captures net-theoretically the fact that for $w = h_1 h_2 h_1 d_1$, firable in N and satisfying (1), no corresponding $v = v_0 d_1$, firable in N and satisfying (2), exists such that $v_0 \in L_0^*$ and $v_0 \sim_{L_0} h_1 h_2 h_1$, where $L_0 = L \cup D \cup H(d_1) = \{d_1, h_1\}$.

6 Undecidability for Non-plain or Non-injective Labellings

One might consider extensions of this work in several directions. One possibility is to relax the assumption of injectivity of net labelling maps. A second possibility is to replace downgrading actions D by (completely) invisible actions I which are neutral with respect to non-interference, thus considering *HLI*-net with $T = H \dot{\cup} L \dot{\cup} I$. By neutral, we mean that on the one hand, such actions have no downgrading effect, and on the other hand, it is not required from low actions not to reveal the firing of these invisible actions. Alternatively, one may add neutral actions I as a fourth class of actions, i.e., consider *HLDI*-net with $T = H \dot{\cup} L \dot{\cup} D \dot{\cup} I$. The former extension would make a kind of bridge with language based security. The latter extensions would be ideal for considering non-interference in multi-agent systems, where the global alphabet of the system is partitioned to $T = H_a \dot{\cup} L_a \dot{\cup} D_a \dot{\cup} I_a$ in as many ways as there are agents a in the system (neutral actions I play essentially the same role as N-events in Mantel's taxonomies [7,8,9]).

Unfortunately, in all cases considered, we obtain undecidability instead of decidability results. The underlying net-theoretic reason is that a statement quoted in section 2, namely

> For two labelled nets N_1 and N_2 where N_2 is plainly *and* injectively labelled, it is decidable whether $L(N_1) \subseteq L(N_2)$,

changes into

> For two labelled nets N_1 and N_2, it is undecidable whether $L(N_1) \subseteq L(N_2)$ even if N_2 is assumed to be plainly *or* injectively labelled,

i.e., as soon as the precondition of N_2 being plainly and injectively labelled is omitted.

Before proceeding to prove these negative results, we propose an extended definition of NDC which, for the sake of simplicity, is based on Theorem 1 instead of Definition 1. Similar extensions could be proposed for INI or INISD, but this is not necessary since the undecidability of NDC for PT-nets with non-plain or non-injective labelling entails similar results for INI or INISD.

Definition 4. Non-deducibility on compositions for labelled nets

Let $N = (P, T, F)$ be a PT-net with initial marking M_0 and labelling map $\lambda\colon T \to H \dot{\cup} L \dot{\cup} \{\varepsilon\}$. N has property NDC iff N and $N \setminus \lambda^{-1}(H)$ are language-equivalent with respect to the set of labels L. □

In the above definition, $I = \lambda^{-1}(\{\varepsilon\})$ is the set of completely invisible transitions. We will prove first the undecidability of NDC for PT-nets with plain but non-injective labelling, i.e., such that $\lambda(t) \neq \varepsilon$ for all $t \in T$ but possibly $\lambda(t) = \lambda(t')$ for $t \neq t'$.

Theorem 5

Given a PT-net N with a plain but non-injective labelling $\lambda\colon T \to H \dot{\cup} L \dot{\cup} \{\varepsilon\}$, it is undecidable whether N has Property NDC.

Proof: Let $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$ be two plainly labelled PT-nets with initial markings $M_{0,1}, M_{0,2}$ and labelling maps $\lambda_1 : T_1 \to L$ and $\lambda_2 : T_2 \to L$. Without loss of generality, assume that P_1, T_1, P_2, T_2 are pairwise disjoint. Let $\{M_{0,2}[t_i\rangle M_{i,2} \mid i = 1 \ldots n\}$ be the set of possible firings from the initial marking of N_2. Embed N_1 and N_2 in a larger net N with two new places p_0, p_0' and $n+1$ new transitions t_0 and $t_1', \ldots, t_n'$ as follows. Initially, p_0 contains one token, places $p \in P_1$ contain $M_{0,1}(p)$ tokens, and all other places are empty. All transitions of N_1 are set in ordinary self-loops with the place p_0', hence they cannot be fired from the initial marking of N. The transition t_0, labelled with $\lambda(t_0) = h \in H$, transfers the missing token from p_0 to p_0', thus enabling N_1 to execute. On the other hand, for each transition $M_{0,2}[t_i\rangle M_{i,2}$ of N_2, N has a corresponding transition t_i', labelled with $\lambda(t_i') = \lambda_2(t_i)$, that takes the token from p_0 and loads the marking $M_{i,2}$ in the places of N_2. Clearly, the initial transition in a run of N determines whether this run simulates a run of N_1 or a run of N_2, and such simulations cannot interfere. Now let $\lambda(t) = \lambda_1(t)$ for $t \in T_1$, and $\lambda(t) = \lambda_2(t)$ for $t \in T_2$. Then, by definition, N has the property NDC if and only if $L(N_1) \subseteq L(N_2)$, and this is undecidable. □

We next show that NDC is undecidable for HLI-nets.

Theorem 6

Given a PT-net N with an injective but non-plain labelling map $\lambda : T \to H \uplus L \uplus \{\varepsilon\}$, it is undecidable whether N has Property NDC.

Proof: Let $N_1 = (P_1, T_1, F_1)$ and $N_2 = (P_2, T_2, F_2)$ be two PT-nets with initial markings M_1, M_2 and injective labelling maps $\lambda_1 : T_1 \to L \uplus \{\varepsilon\}$ and $\lambda_2 : T_2 \to L \uplus \{\varepsilon\}$. W.l.o.g., assume that L is included in the ranges of λ_1 and λ_2, and let $I_1 = \lambda_1^{-1}(\varepsilon)$ and $I_2 = \lambda_2^{-1}(\varepsilon)$. W.l.o.g., assume also that P_1 and P_2 are disjoint.

Construct from N_1 and N_2 a new PT-net N as follows. First, one makes the fusion, for each $l \in L$ of the two transitions of N_1 and N_2 labelled with l, respectively. Let L denote the resulting set of transitions of N. Second, one adds four places p_{00}, p_0, p_1, p_2 and four transitions i_0, h, i_1, i_2, yielding a global set of transitions $H \cup I \cup L$ with $H = \{h\}$ and $I = I_1 \cup I_2 \cup \{i_0, i_1, i_2\}$. The initial marking of N is the joint extension M of M_1 and M_2 defined with $M(p_{00}) = 1$ and $M(p_0), M(p_1), M(p_2) = 0$. The net N inherits all flow relations from N_1 and N_2. The other flow relations are as follows (see Figure 6). First, one sets $F(p_{00}, i_0) = F(i_0, p_0) = F(i_0, p_1) = 1$ and $F(p_{00}, h) = F(h, p_0) = F(h, p_2) = 1$. Second, one sets $F(p_1, i_1) = F(i_1, p_1) = 1$, $F(p_2, i_2) = F(i_2, p_2) = 1$, and $F(p_0, t) = F(t, p_0) = 1$ for every transition $t \in L \cup I_1 \cup I_2$. Finally, one sets $F(i_1, p) = 1$ for every place p originated from N_1 (but not from N_2) and $F(i_2, p) = 1$ for every place p originated from N_2 (but not from N_1).

At the start, only i_0 or h can be fired. After i_0 has fired, i_1 may be fired at any time and as often as desired, hence the language generated by N after firing i_0 is equal to the language of N_2 (recall that only the transitions in L are visible). After h has fired, i_2 may be fired at any time and as often as desired, hence the language generated by N after firing h is equal to the language of N_1.

Now, the language of $N \setminus H$ (or $N \setminus \{h\}$) is clearly equal to the language of N_2. Therefore, N is NDC if and only if $L(N_1) \cup L(N_2) = L(N_2)$, i.e., $L(N_1) \subseteq L(N_2)$, and this cannot be decided. □

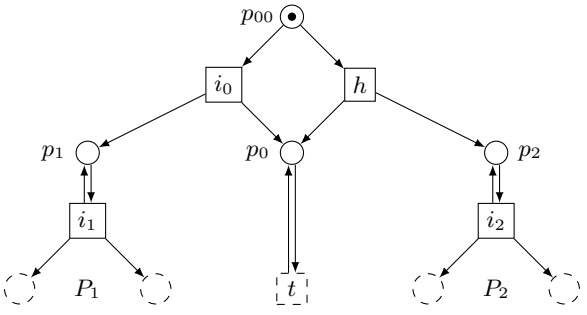


Fig. 6. The construction of N in the proof of Theorem 6

7 Another Example

Consider the HDI-net shown in Fig. 7, where $H(d_i) = \{h_i\}$ for $i = 1 \ldots 3$. The outer circuit comprised of places p_1, p_2, p_3 and low transitions l_1, l_2, l_3 is followed clockwise by sheep, that cannot move from p_i to $p_{(i+1) \mod 3}$ (low transition $l_{(i+1) \mod 3}$) unless the place $r_{(i+1) \mod 3}$ is marked (this place controls a gate and it is marked when the gate is open). Initially, there are two sheep in place p_2, and the gates r_3 and r_1 are open. Sheep may reproduce in place p_2. The central place is occupied by a number of wolves (in the system shown in the figure, there is only one of them). Each wolf may use one of high transitions h_i to hide in the corresponding place q_i and wait there for catching sheep in place p_i. In order to (perhaps) increase chances of success, when using transition h_i, a wolf opens the gate r_i if not already open (so that prey can come in), and tries to close the gate $r_{(i+1) \mod 3}$ (so that prey cannot escape). When a wolf hides in q_i and there is sheep in p_i, the wolf can catch prey and come back to the central place.

The question is to decide whether this net has the INISD property, meaning that the sheep can oppose no strategy to the wolves and cannot ever know that they will be caught until this actually happens. The answer is that the net is not INISD, as $l_3 l_1 h_2 l_2$ is friable but $l_3 l_1 l_2$ is not.

The example may be modified in various ways as follows. If there are two or more wolves, then the net is still not INISD, for the same reason. If there are three open gates initially, then the net is not INISD, since $l_3 l_1 h_1 d_1 l_3 l_1 h_2 l_2$ can be executed, but $l_3 l_1 h_1 d_1 l_3 l_1 l_2$ cannot. If only one gate is open initially, three cases can be distinguished. With gate 3 open, $l_3 h_2 d_2 h_1 l_1$ can be executed and $l_3 h_2 d_2 l_1$ cannot be executed; hence the net is not INISD. With gate 2 open, gate 3 gets closed forever and no low action can take place (only h_1 can fire);

hence the net is INISD. With gate 1 open, $h_3 l_3$ can be executed but l_3 cannot be executed; hence the net is not INISD. If all gates are deconstructed (i.e.: places r_1, r_2, r_3 and all their surrounding arrows are deleted), then the net becomes INISD. Finally, if transition + is deleted, the same reasonings continue to apply.

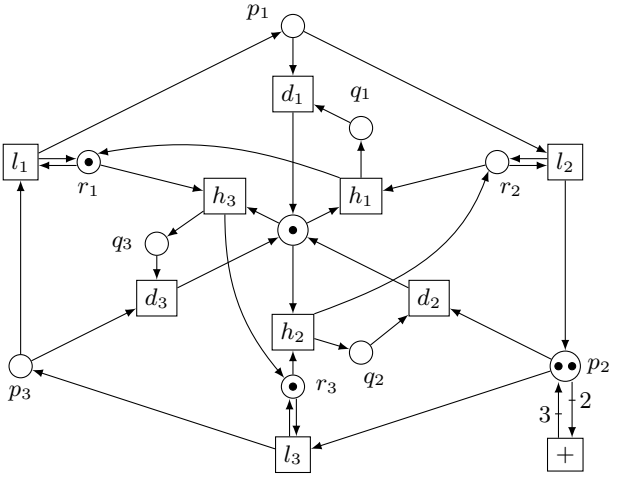


Fig. 7. An HDI-net

8 INISD and Information Flow Security

In this section, the connection between INISD (Definition 3 in section 4) and information flow control will be examined. In particular, Definition 3 will be compared with the definitions of intransitive noninterference studied in [9] and in [10], respectively. First, we show that Definition 3 may be explained alternatively in terms of security domains and the intransitive purge function introduced by Haigh and Young [6] and reformulated by van der Meyden [10].

8.1 Equivalent Reformulation of Definition 3

Let N be an *sd-HLD*-net with $T = H \dot\cup L \dot\cup D$ as in section 4. For $d \in D$, let $H(d)$ denote the set of high-level actions declassified by d. Let $\equiv_D$ and $\equiv_H$ be the equivalence relations on D and H defined by

$$\begin{aligned} & d \equiv_D d' \quad \text{iff} \quad H(d) = H(d') \\ \text{and} \quad & h \equiv_H h' \quad \text{iff} \quad \forall d \in D\colon h \in H(d) \iff h' \in H(d). \end{aligned}$$

For $t \in T$, let the security domain $dom(t)$ of t be defined by $dom(t) = L$ if $t \in L$, $dom(t) = [t]_{\equiv_D}$ if $t \in D$, and $dom(t) = [t]_{\equiv_H}$ if $t \in H$. Let $\mathcal{D}$ be the set of

security domains L or $[d]_{\equiv_D}$ or $[h]_{\equiv_H}$. Consider the intransitive security policy $\rightarrowtail \subseteq \mathcal{D} \times \mathcal{D}$ defined by

$$\begin{array}{ll} x \rightarrowtail x & \text{for all } x \in \mathcal{D} \\ [d]_{\equiv_D} \rightarrowtail L & \text{for all } d \in D \\ [h]_{\equiv_H} \rightarrowtail [d]_{\equiv_D} & \text{for all } d \in D \text{ and } h \in H(d). \end{array}$$

For two security domains x and x', $x \rightarrowtail x'$ means that information may legally flow from x to x'. Finally, consider van der Meyden's *ipurge* function defined as follows for $w \in T^*$, $t \in T$, and for any subset X of security domains:

$$\begin{array}{rcl} ipurge_X(w\,t) & = & \textbf{if } (\exists x \in X : dom(t) \rightarrowtail x) \textbf{ then } (ipurge_{X \cup \{dom(t)\}}(w))\,t \\ & & \qquad\qquad\qquad\qquad\quad \textbf{else } ipurge_X(w) \end{array}$$

Letting $Tr = \{w \in T^* \mid M_0[w\rangle\}$ be the set of firing sequences of N, condition INISD may be reformulated equivalently as follows:

$$\begin{array}{l} \forall w \in Tr, v \in (L \cup H)^* : wv \in Tr \Rightarrow \\ \quad \exists w' \in Tr, v' \in L^* : w'v' \in Tr \wedge (ipurge_{\{L\}}(w) {=} ipurge_{\{L\}}(w')) \wedge v \sim_L v'. \end{array}$$

8.2 Mantel's Framework [9]

According to Mantel's definition of (possibly intransitive) flow policies given in [9], the above defined structure $(\mathcal{D}, \rightarrowtail)$ may be interpreted as $(\mathcal{D}, \rightsquigarrow_V, \rightsquigarrow_N, \not\rightsquigarrow)$ where for all $x, x' \in \mathcal{D}$:

$$\begin{array}{ll} x \rightsquigarrow_V x' & \text{if } x \rightarrowtail x' \\ x \not\rightsquigarrow x' & \text{if } x = [h]_{\equiv_H} \text{ and } (x' = L \text{ or } (x' = [d]_{\equiv_D} \text{ and } h \notin H(d))) \\ x \rightsquigarrow_N x' & \text{if neither } x \rightsquigarrow_V x' \text{ nor } x \not\rightsquigarrow x'. \end{array}$$

The relation $x \rightsquigarrow_N x'$ means that it is not considered important whether information flows or does not flow from x to x'.

In this alternative framework, Mantel proposes two basic security properties: IBSD (Intransitive Backwards Strict Deletion of confidential events) and IBSIA (Intransitive Backwards Strict Insertion of confidential events), parametric on a subset of security domains $\mathcal{D}' \subseteq \mathcal{D}$, and applicable to a set of traces Tr. (In our case, Tr would simply again be the set of firing sequences.) We sketch below (without completely defining IBSD) an explanation why the property INISD studied in this paper has only a relatively loose relationship with IBSD, or more precisely, with the conjunction of $\text{IBSD}_{\mathcal{D}'}$ for all subsets $\mathcal{D}'$ of $\mathcal{D}$ containing L and not containing $[h]_{\equiv_H}$ for any $h \in H$. The reader is referred to [9] for the full definition of IBSD.

Given a fixed $\mathcal{D}' \subseteq \mathcal{D}$, for any transition $t \in T$, let $t \in V$ ("t is visible") if $\exists x' \in \mathcal{D}' : dom(t) \rightsquigarrow_V x'$, and $t \in C$ ("t is confidential") if $\forall x' \in \mathcal{D}' : dom(t) \not\rightsquigarrow x'$. The property $\text{IBSD}_{\mathcal{D}'}(Tr)$ may then be expressed in the following form (where the crucial predicate φ is left unspecified):

$$\forall wtv \in Tr : \quad t \in C \wedge \varphi(v) \quad \Rightarrow \quad \exists v' \in T^* : \; wv' \in Tr \wedge \varphi(v') \wedge v \sim_V v'.$$

IBSD serves here to check that there is no illegal information flow inside the net N, even though such flows of information cannot be detected by any external observer. Indeed, for a fixed $\mathcal{D}' \subseteq \mathcal{D}$, the prefix w of wtv in the expression of IBSD is in general different from $ipurge_{\mathcal{D}'}(w)$, i.e., it cannot be observed from outside. In our definition of INISD, we have taken the opposite stance which, for IBSD, would consist of requiring instead:

$$\forall wtv \in Tr\colon \;\; t \in C \wedge \varphi(v) \;\; \Rightarrow$$
$$\exists w'v' \in T^*\colon \;\; w'v' \in Tr \wedge (ipurge_{\mathcal{D}'}(w) = ipurge_{\mathcal{D}'}(w')) \wedge \varphi(v') \wedge v \sim_V v'.$$

8.3 Van der Meyden's Framework [10]

IP-security (short for *intransitive purge security*) [6,10] is closer to the intransitive noninterference (INISD) model which we have presented in this paper. IP-security may be expressed as the conjunction for all security domains $x \in \mathcal{D}$ of the property

$$\forall w, w' \in Tr\colon \;\; ipurge_{\{x\}}(w) = ipurge_{\{x\}}(w') \;\; \Rightarrow \;\; obs_x(w) = obs_x(w')$$

where obs_x is a fixed family of observation functions parametric on security domains. However, INISD does not coincide with IP-security. Even though IP-security stipulates that observing an existing trace w cannot afford more information than $ipurge_{\{x\}}(w)$, which is the maximal legal information, more information can be inferred without effective information flow. Indeed, IP-security does not stipulate, for a given trace w, that there exist other traces w' such that $ipurge_{\{x\}}(w) = ipurge_{\{x\}}(w')$. In the extreme case where w is alone in its equivalence class with respect to the purge function, $obs_x(w)$ reveals all of w. In our definition of INISD, like in IBSD, we have taken a different stance which, instead of IP-security, would consist of requiring:

$$\forall w \in Tr \;:\;\; ipurge_{\{L\}}(w) \in Tr\,,$$

Like IP-security, INISD suffers from a limitation pointed out and overcome by van der Meyden who proposed for this purpose two other security properties called TA-security and TO-security in [10]. The considered limitation lays in that, in case $h_1 \in H(d_1)$ and $h_2 \in H(d_2)$, if h_1 and d_1 are concurrent with h_2 and d_2 in the net N, the order in which the transitions h_1 and h_2 have been executed may nevertheless be revealed by a firing sequence of the form $\ldots h_1 \ldots h_2 \ldots d_1 \ldots d_2$. The proof techniques that we have developed for deciding INISD rely on the use of the sequential firing rule of Petri nets, and they do not address this limitation. Different techniques should be discovered for modifying INISD accordingly in a truly concurrent framework. We feel that Petri nets, being a privileged model for true concurrency, are well equipped for this future (and possibly quite exciting) task.

9 Conclusion

The only other decidability results of trace-based security properties for infinite-state systems we know about are described in [1,3,4]. Of these papers, [1] is a

direct predecessor of the present one. The result of [3] and its relation to [1] (and therefore also to the present paper) has already been discussed in [1]. The main results of the paper [4], which came to our attention only after [1] had been published, concern the *un*decidability of several of the security properties described in [8] for pushdown systems. These sets of results are not directly comparable, since pushdown languages and Petri net languages are not comparable either. At the end of [4], we also find a decidability result. This result pertains to a very restricted system model. More precisely, it is shown there that a property called "weak non-inference", which falls neither into Mantel's framework [7,8] nor into the NDC/INI/INISD framework of the present paper, and which is undecidable even for finite-state systems, becomes decidable for pushdown systems if one limits both the set of visible actions and the set of secret actions to cardinality 1. Nevertheless, the approaches in the present paper and in [4] lead to the question (which is so far open, to our knowledge), as to which – if any – of the many trace-based transitive security properties of [7,8] are actually decidable for unbounded Petri nets, and to a similar question for the intransitive security policies discussed in section 8.

Acknowledgements. The authors would like to thank several anonymous reviewers for their comments, and in particular, one of them for detecting a technical mistake in Figure 2. A remark of this reviewer also prompted the discussion contained in section 8.

References

1. Best, E., Darondeau, P., Gorrieri, R.: On the Decidability of Non Interference over Unbounded Petri Nets. In: Chatzikokolakis, K., Cortier, V. (eds.) Proceedings 8th International Workshop on Security Issues in Concurrency, SecCo. EPTCS, vol. 51, pp. 16–33 (2010), http://dx.doi.org/10.4204/EPTCS.51.2
2. Busi, N., Gorrieri, R.: Structural Non-Interference in Elementary and Trace Nets. Mathematical Structures in Computer Science 19(6), 1065–1090 (2009), doi:10.1017/S0960129509990120
3. Dam, M.: Decidability and Proof Systems for Language-based Noninterference Relations. In: Proc. POPL 2006, pp. 67–78 (2006), doi:10.1145/1111037.1111044
4. D'Souza, D., Holla, R., Kulkarni, J., Ramesh, R.K., Sprick, B.: On the Decidability of Model-Checking Information Flow Properties. In: Sekar, R., Pujari, A.K. (eds.) ICISS 2008. LNCS, vol. 5352, pp. 26–40. Springer, Heidelberg (2008)
5. Gorrieri, R., Vernali, M.: On Intransitive Non-interference in Some Models of Concurrency. In: Aldini, A., Gorrieri, R. (eds.) FOSAD 2011. LNCS, vol. 6858, pp. 125–151. Springer, Heidelberg (2011), http://dx.doi.org/10.1007/978-3-642-23082-0_5
6. Haigh, T.J., Young, W.D.: Extending the noninterference versions of MLS for SAT. IEEE Trans. on Software Engineering SE-13(2), 141–150 (1987)
7. Mantel, H.: Possibilistic Definitions of Security - an Assembly Kit. In: Proc. of the 13th IEEE Computer Security Foundations Workshop, Cambridge, UK, July 3-5, pp. 185–199 (2000)

8. Mantel, H.: A Uniform Framework for the Formal Specification and Verification of Information Flow Security. PhD Thesis, Universität des Saarlandes (2003)
9. Mantel, H.: Information Flow Control and Applications - Bridging a Gap. In: Oliveira, J.N., Zave, P. (eds.) FME 2001. LNCS, vol. 2021, pp. 153–172. Springer, Heidelberg (2001)
10. van der Meyden, R.: What, Indeed, Is Intransitive Noninterference? In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 235–250. Springer, Heidelberg (2007)
11. Wimmel, H.: Entscheidbarkeit bei Petri Netzen - Überblick und Kompendium, p. 239. Springer, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-85471-5