Non-interference by Unfolding*

Paolo Baldan[1] and Alberto Carraro[2]

[1] Università di Padova, Dipartimento di Matematica, Italy
baldan@math.unipd.it
[2] Università Ca' Foscari di Venezia, DAIS, Italy
and ANR Projet Récré, France
acarraro@pps.univ-paris-diderot.fr

Abstract. The concept of non-interference has been introduced to characterise the absence of undesired information flows in a computing system. Although it is often explained referring to an informal notion of causality - the activity involving the part of the system with higher level of confidentiality should not cause any observable effect at lower levels - it is almost invariably formalised in terms of interleaving semantics. Here we focus on Petri nets and on the BNDC property (Bisimilarity-based Non-Deducibility on Composition), a formalisation of non-interference widely studied in the literature. We show that BNDC admits natural characterisations based on the unfolding semantics - a classical true concurrent semantics for Petri nets - in terms of causalities and conflicts between high and low level activities. This leads to an algorithm for checking BNDC for safe Petri nets which relies on the construction of suitable complete prefixes of the unfolding. A prototype tool provides very promising results.

1 Introduction

The concept of non-interference has been introduced to characterise the absence of undesired information flows in a computing system. The underlying idea is simple: a system is viewed as consisting of components at different levels of confidentiality, in the simplest case a high part H, which intuitively should be secret, and a low part L, which is public. The absence of a flow of information from H to L is captured by asking that the activity of H does not determine visible effects, according to some selected observational semantics, at low level L. Originally the notion of non-interference [1] has been defined for deterministic programs, using a trace semantics. Since then, several variants have been studied, dealing with non-deterministic systems and finer observations, notably bisimilarity. In the setting of process calculi, a popular formulation is the so-called NDC (Non-Deducibility on Composition), which states that a process S is free of interferences whenever S running in isolation, seen from the low level, is behaviourally equivalent to S interacting with any parallel high level process. A

* Work supported by the project Récré (ANR) and the MIUR PRIN project CINA.

G. Ciardo and E. Kindler (Eds.): PETRI NETS 2014, LNCS 8489, pp. 190–209, 2014.

systematic presentation of non-interference notions for concurrent systems can be found, e.g., in [2,3,4].

Although the definition of non-interference often informally refers to some notion of causality – the activity at high level should not cause visible effects on the behaviour at low level – it has been almost invariably formalised in terms of interleaving semantics.

The value of a characterisation of non-interference in terms of a true concurrent semantics can be twofold. At a conceptual level, it can unveil the relation between causality and interferences. On the pragmatic side, the use of a true concurrent semantics, which represents concurrency of computational steps explicitly, rather than by their possible interleavings, can be helpful for facing the state explosion problem which affects the verification of concurrent systems.

A first step towards a causal characterisation of non-interference can be found in [5], where several notions of non-interference, in particular BNDC (Bisimilarity-based NDC), are formulated over Petri nets. More specifically, the authors consider contact-free elementary nets (or equivalently, safe P/T nets without self-loops), and they prove that a net satisfies the BNDC property when it does not include neither causal nor conflict places, i.e., places satisfying semi-structural conditions intended to identify undesired interactions between high and low transitions on those places. The result is also generalised to trace nets which extend elementary nets with arcs for testing the presence/absence of tokens in a place, and arcs for filling/emptying a place regardless of its previous content. The notions of causal and conflict place, however, are formulated by requiring the presence of suitable reachable markings and firing sequences, i.e., referring to an interleaving semantics rather than to a causal, true concurrent semantics. In [6] the authors face the question of decidability of BNDC over general P/T nets, providing a positive answer, but leaving aside the problem of providing a causal characterisation.

In this paper we study the notion of non-interference for Petri nets, formalised in terms of BNDC, aiming at a characterisation in terms of a causal semantics.

We first provide a characterisation of BNDC for P/T nets based on notions of causal and conflict places. Our definition slightly simplifies that given in [5] for elementary and safe nets, in a way that is later useful for the development of BNDC checking algorithms.

Focusing on safe nets, the above results are then exploited for obtaining a causal characterisation of BNDC in terms of a classical true concurrent semantics of Petri nets, the unfolding semantics [7]. More precisely, we prove that a safe net is BNDC if and only if its unfolding does not reveal a direct causality from a high level to a low level transition or a direct conflict between a high and a low level transitions. This enables the checking of the BNDC property on a suitably defined complete prefix of the unfolding of a safe net. In fact, we prove that it is possible to build a finite prefix of the unfolding which includes an interference witness if and only if the original net is not BNDC. As mentioned above, checking the property on an unfolding prefix rather than on the reachability graph can improve greatly the efficiency of the check for concurrent systems, since

unfolding-based partial order verification techniques have been proven effective in facing the state explosion problem (see, e.g., [8], and references therein).

We developed a prototype tool UBIC (Unfolding-Based Interference Checker) for checking BNDC on safe nets which implements the algorithm proposed in the paper. The prototype is based on CUNF [9], a toolset for carrying out unfolding-based verification on Petri nets. Some preliminary experimental results are extremely promising in comparison to those provided by existing tools for checking BNDC based on interleaving semantics, like ANICA [10].

The rest of the paper is organised as follows. In §2 we define Petri nets and BNDC, the non-interference notion we will work with. In §3 we give a characterisation of BNDC based on causal and conflict places. After recalling in §4 the basics of the unfolding semantics, in §5 we provide an unfolding-based characterisation of BNDC on safe nets which is exploited for devising a corresponding algorithm for checking BNDC. In §6 we report some experimental results obtained with the tool UBIC, implementing the proposed algorithm. Finally, in §7 we draw some conclusions.

2 Preliminaries

A *(Petri) net* is a tuple $N = (P, T, F)$ where P, T are two disjoint sets of *places* and *transitions*, respectively, and $F : (P \times T) \cup (T \times P) \to \mathbb{N}$ is the *flow function*, representing a set of directed edges with non-negative integer weights, such that for every $t \in T$ there exists $p \in P$ such that $F(p, t) > 0$.

A *marking* of N is a function $m : P \to \mathbb{N}$. A transition $t \in T$ is *enabled* at a marking m, denoted $m[t\rangle$, if $m(p) \geq F(p, t)$ for all $p \in P$. If $m[t\rangle$ then t can be *fired* leading to a new marking m', written $m[t\rangle m'$, defined by $m'(p) = m(p) + F(t, p) - F(p, t)$ for all places $p \in P$. The enabling and firing relations are extended to $\sigma \in T^*$ (set of all finite sequences of elements of T) by defining $m[\varepsilon\rangle m$ (where ε is the empty sequence) and $m[\sigma t\rangle m''$ when $m[\sigma\rangle m'$ and $m'[t\rangle m''$. For a marking m, we denote $m^\circ = \{t \in T : m[t\rangle\}$.

A *marked net* is a pair $\mathbf{N} = (N, m_0)$ where N is a net and m_0 is a marking of N. A marking m' is *reachable* if there exists $\sigma \in T^*$ such that $m_0[\sigma\rangle m'$. The set of reachable markings of $\mathbf{N}$ is denoted by $[m_0\rangle$. When $m[t\rangle m'$, the marking m', uniquely determined by m and t, is denoted by $\langle m[t\rangle$. Analogously, for $\sigma \in T^*$, if $m[\sigma\rangle$ we can define the marking $\langle m[\sigma\rangle$.

A *net system* is a net N where T is partitioned into two disjoint sets H and L. We sometimes write $N = (P, H, L, F)$ making the two sets of transitions explicit. The idea is to model a system with two distinguished classes of "users": high level users and low level users. Transitions in H, called *high transitions*, are thought of as visible only to the high level users, while the *low transitions* in L are globally visible. A net system N is called *high* when $L = \emptyset$.

The *low view* (see [5, Definition 3.1]) of a sequence $\sigma \in T^*$ represents the view of a low level user on the run σ w.r.t. the system N. It is defined inductively by $\nu_N(\varepsilon) = \varepsilon$ and $\nu_N(t\,\sigma) = t\,\nu_N(\sigma)$ if $t \in L$ and $\nu_N(t\,\sigma) = \nu_N(\sigma)$ if $t \in H$.

An example of a net system, used also in [11], can be found in Fig. 1(a). It intuitively consists of a high and a low level subsystem, accessing in mutual

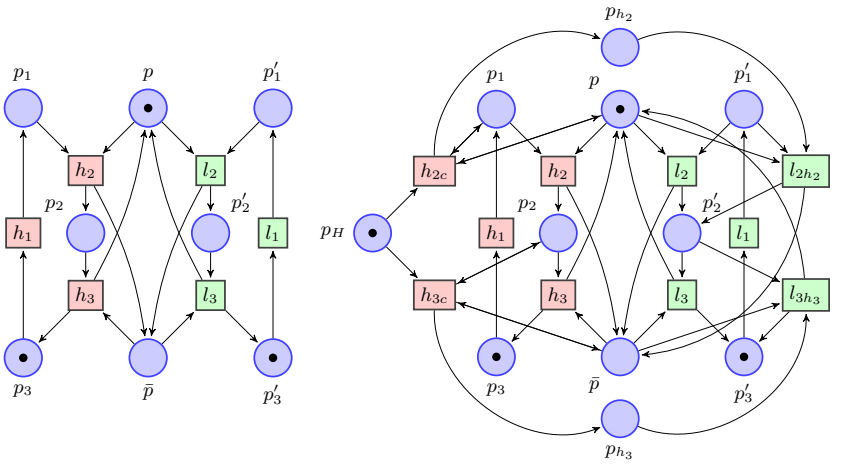

Fig. 1. (a) A net system $\mathbf{R}$ and (b) its causal reduct $\gamma(\mathbf{R})$

exclusion a common resource, represented by places p and $\bar{p}$. When place p is marked, the resource is free, when place $\bar{p}$ is marked, the resource is taken. Transitions h_1, l_1 represent the non-critical sections of the subsystems: they do not need the resource and they can run in parallel. Transitions h_2 and l_2 enter the critical section, hence they compete for the token in p. The resource is released and the critical section left by firing h_3, resp. l_3.

Following [5,6] we focus on a notion of non-interference known as BNDC (Bisimilarity-based NDC). The discriminating power of a low level user is formalised by means of a bisimulation equivalence based on the low views.

Definition 1 (low view bisimulation). *Let* $\mathbf{N}$ *and* $\mathbf{N}'$ *be two marked net systems. A* view simulation *of* $\mathbf{N}$ *by* $\mathbf{N}'$ *is a relation* $R \subseteq [m_0\rangle \times [m'_0\rangle$ *such that:*

- $(m_0, m'_0) \in R$;
- *if* $(m, m') \in R$ *and* $m[\sigma\rangle$ *then there exists* σ' *such that* $m'[\sigma'\rangle$, $(\langle m[\sigma\rangle, \langle m'[\sigma'\rangle) \in R$ *and* $\nu_N(\sigma) = \nu_{N'}(\sigma')$.

A low view bisimulation *between* $\mathbf{N}$ *and* $\mathbf{N}'$ *is a relation* $R \subseteq [m_0\rangle \times [m'_0\rangle$ *such that both* R *and* R^{-1} *are view simulations. If there exists a low view bisimulation between* $\mathbf{N}$ *and* $\mathbf{N}'$*, we say that they are* low view bisimilar *and we write* $\mathbf{N} \approx \mathbf{N}'$.

We remark that Definition 1 above is an instance of [6, Definition 2.5], which introduces a notion of weak bisimilarity for general partially observed labelled transition systems.

A net system $\mathbf{N}$ is deemed free of interferences when the behaviour of $\mathbf{N}$ interacting with any parallel high system is low view bisimilar to the behaviour of $\mathbf{N}$, restricted to the low transitions. In order to formalise this intuition we

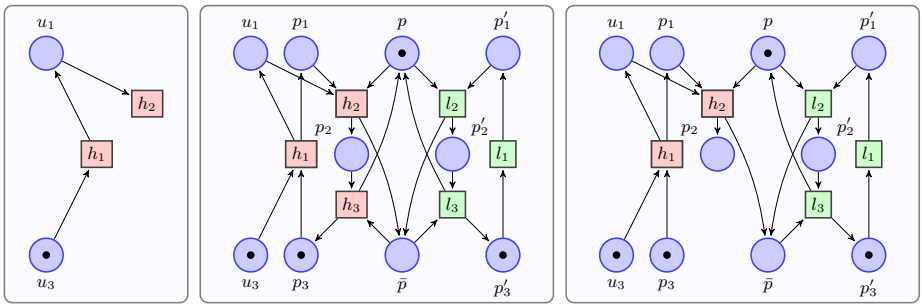

Fig. 2. (a) A high net system $\mathbf{D}$; (b) the parallel composition $\mathbf{R} \,|\, \mathbf{D}$, where $\mathbf{R}$ is the net system in Fig. 1(a); (c) the restriction $(\mathbf{R} \,|\, \mathbf{D}) \setminus (H - H_D)$

need to define (parallel) composition and restriction operators on net systems, as done in [6].

Definition 2 (composition). *Let N and N' be two net systems such that $P \cap P' = \emptyset$, $(H' \cap L) \cup (H \cap L') = \emptyset$, and $(P \cap T') \cup (P' \cap T) = \emptyset$. The* composition *of N and N' is the net system $N \,|\, N' = (P \cup P', H \cup H', L \cup L', F \,|\, F')$ where $F \,|\, F'$ is the least (w.r.t. inclusion) flow function on $P \cup P'$ and $T \cup T'$ that extends both F and F'. The composition of two marked net systems $\mathbf{N} = (N, m_0)$ and $\mathbf{N}' = (N', m'_0)$ is the marked net system $\mathbf{N} \,|\, \mathbf{N}' = (N \,|\, N', m_0 \,|\, m'_0)$ where $m_0 \,|\, m'_0$ is the marking on $N \,|\, N'$ that extends both m_0 and m'_0.*

Intuitively, $N \,|\, N'$ is the parallel composition of N and N', synchronised on the common transitions. Whenever we consider two net systems N and N' we shall implicitly assume that the disjointness requirements are satisfied so that $N \,|\, N'$ is always defined. Note that this may require some "renaming".

Definition 3 (restriction). *Given a net system N and a subset $T_1 \subseteq T$, the* restriction *of N by T_1 is the net system $N \setminus T_1 = (P, H - T_1, L - T_1, F \setminus T_1)$ where $F \setminus T_1$ is the restriction of F to $(P \times (T - T_1)) \cup ((T - T_1) \times P)$. For a marked net system $\mathbf{N}$, the restriction $\mathbf{N} \setminus T_1$ is $(N \setminus T_1, m_0)$.*

The restricted net system is thus obtained by simply removing the restricted transitions. For instance, the net system $\mathbf{R} \,|\, \mathbf{D}$ - where $\mathbf{R}$ is the net in Fig. 1(a) and $\mathbf{D}$ is the high net system in Fig. 2(a) - is depicted in Fig. 2(b). The restriction of $\mathbf{R} \,|\, \mathbf{D}$ with respect to $\{h_3\}$, namely $(\mathbf{R} \,|\, \mathbf{D}) \setminus \{h_3\}$ is depicted in Fig. 2(c).

We can now recall the definition of the BNDC property on net systems.

Definition 4 (BNDC [6]). *A marked net system $\mathbf{N}$ is BNDC if for every high marked net system $\mathbf{N}'$ we have that $\mathbf{N} \setminus H \approx (\mathbf{N} \,|\, \mathbf{N}') \setminus (H - H')$.*

Intuitively, a net system $\mathbf{N}$ is BNDC when no behavioural difference can be detected by a low level observer between $\mathbf{N} \setminus H$ (i.e. $\mathbf{N}$ running in isolation and without firing high level transitions) and $(\mathbf{N} \,|\, \mathbf{N}') \setminus (H - H')$ (i.e., $\mathbf{N}$ running in

parallel with an arbitrary high net $\mathbf{N}'$, thus possibly synchronising on high level transitions).

For instance, the net system $\mathbf{R}$ in Fig. 1(a) is not BNDC. In fact, $\mathbf{R} \setminus H$ is clearly not bisimilar to $(\mathbf{R}|\mathbf{D}) \setminus (H - H_D)$, since the latter has a deadlock (reached, e.g., by firing $h_1 h_2$). The intuition is that if the high level component deadlocks in the critical zone (after firing h_2) then this is detectable by the low level component because it will no longer be able to enter the critical section and access the resource.

3 BNDC through Causal and Conflict Places

In this section we start describing the contributions of this paper, showing how the BNDC property can be characterised in terms of the absence of suitably defined causal and conflict places. Roughly speaking, a net system is BNDC when there is no causal flow between high and low transitions, and high and low transitions are never in conflict, competing for a token. An analogous characterisation for elementary Petri nets without self-loops has been originally provided in [5]. Our characterisation works for P/T nets, possibly with self-loops, and it is based on a simpler notion of causal and conflict place.

For $x \in P \cup T$ we define ${}^\bullet x = \{y \in P \cup T : F(y,x) > 0\}$ (the *pre-set* of x) and $x^\bullet = \{y \in P \cup T : F(x,y) > 0\}$ (the *post-set* of x).

Definition 5 (causal place). *A place p of a net system $\mathbf{N}$ is* causal *if there exist transitions $l \in L$, $h \in H$ and a sequence $\tau \in L^*$ such that (1) $p \in {}^\bullet l \cap h^\bullet$; (2) there is $m \in [m_0\rangle$ satisfying $m[h\tau l\rangle$, $m[\tau\rangle$ and $\langle m[\tau\rangle(p) < F(p,l)$.*

Intuitively, in Definition 5, the fact that $p \in {}^\bullet l \cap h^\bullet$ means that in the place p a token can be potentially produced by a high transition and consumed by a low one, thus determining a flow of information from high to low level. Condition (2) ensures that this indeed happens in some computation starting from a reachable marking. In fact, transition l is enabled by $m[h\tau\rangle$ but not by $m[\tau\rangle$, since $\langle m[\tau\rangle(p) < F(p,l)$. This means that h puts at least one token in p which can be later used by the firing of l. In the spirit of [5] a place satisfying the conditions of Definition 5 should be called *active causal* but we preferred a more concise terminology.

Similarly, a *conflict place* is defined as a place where a low and a high transitions compete for a token in some computation.

Definition 6 (conflict place). *A place p of a net system $\mathbf{N}$ is* conflict *if there exist transitions $l \in L$, $h \in H$ and a sequence $\tau \in L^*$ such that (1) $p \in {}^\bullet l \cap {}^\bullet h$; (2) there is $m \in [m_0\rangle$ satisfying $m[h\tau\rangle$, $m[\tau l\rangle$ and $\langle m[h\tau\rangle(p) < F(p,l)$.*

We can now prove that for a net system the absence of causal or conflict places is a necessary and sufficient condition for being BNDC.

Theorem 1 (BNDC by causal/conflict places). *A marked net system $\mathbf{N}$ is not BNDC iff $\mathbf{N}$ has either a causal place or a conflict place.*

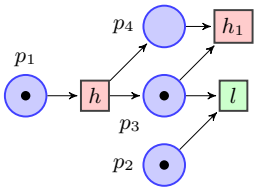

Fig. 3. Weak causal places are not always causal places

3.1 Safe Nets

Recall that a net $\mathbf{N}$ is *safe* if for every $p \in P$ and $m \in [m_0\rangle$ we have $m(p) \leq 1$. In a safe net, any marking m can be seen a subset of places and we write $p \in m$ instead of $m(p) = 1$. For safe net systems we can characterise BNDC in terms of a weaker notion of conflict and causal place (Definitions 7 and 8); these notions are novel and are motivated by the fact that images of these places are easily recognisable in the *unfolding* of the net (see §4), which is the main object of study in this paper.

Notation. Given a net system $\mathbf{N}$, a place $p \in P$ and a transition $t \in T$, we set $\delta_t(p) = F(t,p) - F(p,t)$, $t^- = \{p \in P : \delta_t(p) < 0\}$ and $t^+ = \{p \in P : \delta_t(p) > 0\}$.

Definition 7 (weak causal place). *A place p of a net system $\mathbf{N}$ is* weak causal *if there exist transitions $l \in L$, $h \in H$ and a sequence $\tau \in (L \cup H)^*$ such that (1) $p \in {}^\bullet l \cap h^+$; (2) there is $m \in [m_0\rangle$ such that $m[h\tau l\rangle$.*

In words, we require the existence of a computation in which h precedes l, and of a place $p \in {}^\bullet l \cap h^+$, namely in the pre-set of l and where h contributes positively. Intuitively, in this way the firing of l could depend on the firing of h. A notion of weak conflict place can be defined along the same lines.

Definition 8 (weak conflict place). *A place p of a net system $\mathbf{N}$ is* weak conflict *if there exist transitions $l \in L$, $h \in H$ and a sequence $\tau \in (L \cup H)^*$ such that (1) $p \in {}^\bullet l \cap h^-$; (2) there is $m \in [m_0\rangle$ satisfying $m[h\rangle$ and $m[\tau l\rangle$.*

Every causal/conflict place is also a weak causal/conflict place. The converse implication does not hold in general, i.e., the presence of a weak causal/conflict place does not imply the existence of a causal/conflict place. For instance, consider the net system in Fig. 3: p_3 is a weak causal place, as witnessed by the firing sequence $h\, h_1\, l$, but p_3 is not a causal place and, actually, the net is BNDC.

Remarkably, in safe net systems, the existence of a weak causal (resp. conflict) place, implies the existence of a causal (resp. conflict) place. This fact is at the basis of the following theorem.

Theorem 2 (BNDC in safe nets). *Let $\mathbf{N}$ be a safe marked net system. The following statements are equivalent :*

(i) $\mathbf{N}$ contains either a causal place or a conflict place

(ii) $\mathbf{N}$ *contains either a weak causal place or a weak conflict place*
(iii) $\mathbf{N}$ *is not BNDC*

Moreover for every safe net $\mathbf{N}$ all interferences can be reduced to causal ones in the following sense. We can build a safe net $\gamma(\mathbf{N})$ such that $\mathbf{N}$ contains a weak causal or conflict place iff $\gamma(\mathbf{N})$ contains a weak causal place.

Definition 9 (causal reduct). *Given a net system* $\mathbf{N}$*, we define its* causal reduct $\gamma(\mathbf{N})$ *as the net system* $(\gamma(N), \gamma(m_0))$ *obtained from* $\mathbf{N}$ *as follows. Let* $H_{\#} = \{h \in H : {}^\bullet l \cap h^- \neq \emptyset \text{ for some } l \in L\}$*. Then:*

- *add a new place* p_H*;*
- *add a new place* p_h *for each transition* $h \in H_{\#}$*;*
- *for each transition* $h \in H_{\#}$*, add a new high transition* h_c*, setting* $F(p, h_c) = F(h_c, p) = F(p, h)$ *for every* $p \in P$*; moreover set* $F(p_H, h_c) = 1$ *and* $F(h_c, p_h) = 1$*;*
- *for each pair of transitions* $h \in H$*,* $l \in L$ *such that* ${}^\bullet l \cap h^- \neq \emptyset$*, insert a new low transition* l_h *setting* $F(p, l_h) = F(p, l)$ *and* $F(l_h, p) = F(l, p)$ *for every* $p \in P$*; moreover set* $F(p_h, l_h) = 1$*;*
- *finally* $\gamma(m_0)$ *is the unique extension of* m_0 *that assigns* 1 *to* p_H *and* 0 *to all other new places.*

In words, each h_c is a copy of the corresponding high transition h. It is enabled when h is enabled but its firing does not change the marking (it puts all consumed tokens back), apart from generating a token in the new place p_h which can later be used by a new transition l_h. Transition l_h is in turn a copy of l whose pre-set is extended with the addition of place p_h. The fact that each h_c inputs a token from p_H implies than every firing sequence will include at most one newly added high transition and at most one newly added low transition.

As an example, the causal reduct $\gamma(\mathbf{R})$ of the net $\mathbf{R}$ can be found in Fig. 1(b). E.g., the fact that $p \in {h_2}^- \cap {}^\bullet l_2 \neq \emptyset$ in $\mathbf{R}$ leads to the creation of a copy h_{2c} of h_2, place p_{h_2} and a copy l_{2h_2} of transition l_2.

Proposition 1 (BNDC via the causal reduct). *Let* $\mathbf{N}$ *be a safe net system. Then* $\mathbf{N}$ *is not BNDC iff* $\gamma(\mathbf{N})$ *contains a weak causal place.*

Roughly, the core of the proof relies on the fact that every firing sequence $m_1[\tau\rangle m_2$ in $\mathbf{N}$ is also a firing sequence in $\gamma(\mathbf{N})$. A weak causal place in $\mathbf{N}$ is also a weak causal place in $\gamma(\mathbf{N})$. Given a weak conflict place p in $\mathbf{N}$, with $p \in {}^\bullet l \cap h^-$, and $m[h\rangle$, $m[\tau l\rangle$ for some reachable marking $m \in [m_0\rangle$, we can construct in the net $\gamma(\mathbf{N})$ the firing sequence $m'[h_c \tau l_h\rangle$ with $p \in {}^\bullet l_h \cap {h_c}^+$, witnessing that p_h is a weak causal place in $\gamma(\mathbf{N})$. For an example of this phenomenon see Fig. 4(b), where the weak causal place p_{h_2} witnesses the presence of a weak conflict place p. Vice versa the presence of a weak causal place p in $\gamma(\mathbf{N})$ implies that either p itself is a weak causal place in $\mathbf{N}$ or that there is some weak conflict place in $\mathbf{N}$ (none of the newly added places can be a weak conflict place).

4 Unfolding Semantics and Related Notions

We recall the definition of the unfolding semantics for safe P/T nets, which is used to develop the results of the next section §5. The expert reader can quickly browse the current section for notation and jump to the next one.

The dynamics of a net system $\mathbf{N}$ can be represented by its *marking graph*, a directed graph whose nodes are the reachable markings of $\mathbf{N}$. When $\mathbf{N}$ is safe, its marking graph is finite but its size can still be exponential in the size of $\mathbf{N}$. An implicit representation of the marking graph is given by the *unfolding* of $\mathbf{N}$, a possibly infinite net which, when $\mathbf{N}$ is safe, admits a finite fragment containing a representation of all the reachable markings. Moreover the size of such a fragment is, in presence of many concurrent transitions, considerably smaller than the marking graph [12,13,8].

The following relations can be defined on arbitrary nets, but they assume particular significance on net unfoldings.

Definition 10 (causality/conflict/concurrency). *Let N be a net. The* causality *relation $<$ is the least transitive binary relation on $P \cup T$ such that $x < y$ if $x \in {}^\bullet y$ (or, equivalently, if $y \in x^\bullet$). By $\leq$ we denote the reflexive closure of $<$. The* conflict *relation $\sharp$ is the least symmetric binary relation on $P \cup T$ such that:*
(1) if $t, t' \in T$, $t \neq t'$ and ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$ then $t \sharp t'$;
(2) if $x < x'$ and $x \sharp y$ then $x' \sharp y$.
Finally the concurrency *relation $\parallel$ is the complement of $(\sharp \cup \leq \cup >)$.*

The relation $\parallel$ is symmetric and irreflexive. Given a subset $X \subseteq P \cup T$, we write $\mathsf{co}(X)$ when X is pairwise concurrent, i.e., $x \parallel y$ for all $x, y \in X$, $x \neq y$.

Net unfoldings fall into the class of *occurrence nets* [7]: their main properties are that causality is acyclic and well-founded, conflict is irreflexive and the arcs of the flow relation have weight at most 1 (i.e. F is a relation). Moreover their initial marking is often left implicit as it is identified as the set of minimal places.

Given a net $\mathbf{N}$, an unfolding construction produces a net $\mathcal{U}(\mathbf{N})$ whose places are a reification of the tokens that circulate in $\mathbf{N}$ and whose transitions are copies of the transitions of $\mathbf{N}$, representing their possible firings. The unfolding is constructed inductively starting from the initial marking of $\mathbf{N}$ and then adding, at each step, an occurrence of each transition of $\mathbf{N}$ which is enabled by (the image of) a concurrent subset of the places already generated.

For the sake of the presentation, hereafter we restrict to safe net systems $\mathbf{N}$, where all reachable markings are (equivalent to) subsets of P and $F(x, y) \leq 1$ for $x, y \in P \cup T$, so that the flow function F is completely determined by the pre- and post-sets of transitions. In what follows we indicate by π_1 and π_2, respectively, the standard projection functions over the first and second component of pairs.

Definition 11 (unfolding). *Let $\mathbf{N} = ((P, T, F), m_0)$ be a safe marked net and let $\perp$ be an element not in P, T, F. Define the net $U^{(0)} = (T^{(0)}, P^{(0)}, F^{(0)})$ as:*

$$T^{(0)} = \emptyset \ ; \qquad P^{(0)} = \{(p, \perp) : p \in m_0\} \ ; \qquad F^{(0)} = \emptyset \ .$$

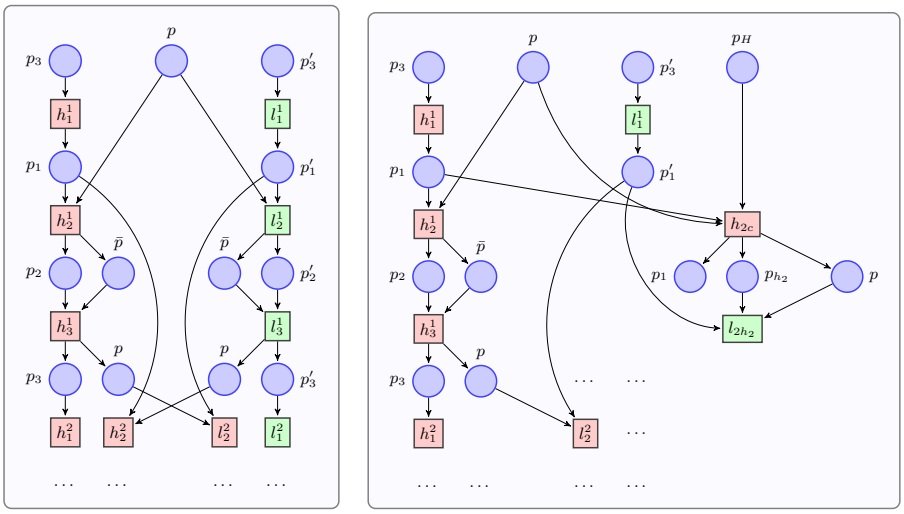

Fig. 4. (a) A fragment of the unfolding $\mathcal{U}(R)$ of the net system in Fig. 1(a); (b) a fragment of the unfolding $\mathcal{U}(\gamma(R))$ of its causal reduct $\gamma(R)$ in Fig. 1(b)

Then the unfolding is the least net $\mathcal{U}(\mathbf{N}) = (P^{(\omega)}, T^{(\omega)}, F^{(\omega)})$ *containing* $U^{(0)}$ *and such that*

- *if* $t \in T$ *and* $X \subseteq P^{(\omega)}$ *is such that* $\mathsf{co}(X)$ *and* $\pi_1(X) = {}^\bullet t$ *then* $(t, X) \in T^{(\omega)}$*;*
- *if* $e \in T^{(\omega)}$*,* $p \in \pi_1(e)^\bullet$ *then* $(p, e) \in P^{(\omega)}$*;*
- $F^{(\omega)}(x,y) = \begin{cases} 1 & \text{if } x \in T^{(\omega)},\ y \in P^{(\omega)} \text{ and } x = \pi_2(y) \\ 1 & \text{if } x \in P^{(\omega)},\ y \in T^{(\omega)} \text{ and } x \in \pi_2(y) \\ 0 & \text{otherwise} \end{cases}$

The unfolding of a marked net system is an occurrence net system. We set $H^{(\omega)} = \{e \in T^{(\omega)} : \pi_1(e) \in H\}$ and $L^{(\omega)} = \{e \in T^{(\omega)} : \pi_1(e) \in L\}$. Places and transitions in the unfolding represent tokens and firing of transitions, respectively, of the original net. Each place in the unfolding is a tuple recording the place in the original net and the "history" of the token. By historical reasons the transitions in the unfolding are also called *events*. The projection π_1 over the first component maps places and transitions of the unfolding to the corresponding items of the original net $\mathbf{N}$.

For instance, a fragment of the unfolding of the net system in Fig. 1(a) can be found in Fig. 4(a). Different occurrences of a transition are distinguished using a superscript. E.g., h_2^1 and h_2^2 are two different occurrences of transition h_2.

For a transition $e \in T^{(\omega)}$ we define its *causal closure* $[e] = \{e' \in T^{(\omega)} : e' \leq e\}$. We write $[e)$ for the set of strict causes, i.e., $[e) = [e] - \{e\}$. The runs of $\mathbf{N}$ are represented by the configurations of $\mathcal{U}(\mathbf{N})$, i.e., subsets of $T^{(\omega)}$ that are causally closed and conflict-free.

Definition 12 (configuration). *A* configuration *of* $\mathcal{U}(\mathbf{N})$ *is a finite subset* $C \subseteq T^{(\omega)}$ *such that* $(C \times C) \cap \sharp = \emptyset$ *and* $[e] \subseteq C$ *for all* $e \in C$. *The set of all configurations of* $\mathcal{U}(\mathbf{N})$ *is denoted by* $\mathcal{C}(\mathcal{U}(\mathbf{N}))$.

A configuration of $\mathcal{U}(\mathbf{N})$ can be associated with a reachable marking of $\mathbf{N}$, obtained by firing all its transitions in any order compatible with causality. The *frontier* of a configuration C is the set $C^\circ = (P^{(0)} \cup \bigcup_{e \in C} e^\bullet) - (\bigcup_{e \in C} {}^\bullet e)$; it induces a marking $\mathsf{M}(C)$ on $\mathbf{N}$ defined by $\mathsf{M}(C)(p) = |\{b \in C^\circ : \pi_1(b) = p\}|$, for every place p in $\mathbf{N}$. The unfolding has been shown to be marking complete in the sense that $m \in [m_0\rangle$ iff there exists $C \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$ such that $\mathsf{M}(C) = m$ (see [7,14,12]).

5 Non-interference in the Unfolding

We show that the BNDC property for safe net systems can be characterised by quite effective conditions on the unfolding. This is later used to devise unfolding-based algorithms for checking BNDC on this class of net systems.

5.1 Images of Weak Causal and Weak Conflict Places in the Unfolding

Notation. For a place b and a transition e in $\mathcal{U}(\mathbf{N})$ we set $e^+ = \{b \in P^{(\omega)} : \pi_1(b) \in \pi_1(e)^+\}$ and $e^- = \{b \in P^{(\omega)} : \pi_1(b) \in \pi_1(e)^-\}$. Moreover transitions denoted by h and l, possibly with subscripts, will be implicitly assumed to be high and low, respectively.

Theorem 3. *Let* $\mathbf{N}$ *be a net system.*

(i) If p *is a weak causal place in* $\mathbf{N}$, *then there are transitions* h', l' *and a place* b *in* $\mathcal{U}(\mathbf{N})$ *such that* $b \in {}^\bullet l' \cap h'^+$ *and* $\pi_1(b) = p$;
(ii) if p *is a weak conflict place in* $\mathbf{N}$, *then there are transitions* h', l' *and a place* b *in* $\mathcal{U}(\mathbf{N})$ *such that* $b \in {}^\bullet l' \cap h'^-$, $[h'] \cup [l'] \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$ *and* $\pi_1(b) = p$.

Theorem 3 gives structural conditions that the unfolding $\mathcal{U}(\mathbf{N})$ necessarily satisfies if $\mathbf{N}$ contains either a weak causal or a weak conflict place. Conversely, the next theorem gives structural conditions on $\mathcal{U}(\mathbf{N})$ which are sufficient for $\mathbf{N}$ to contain either a weak causal or a weak conflict place.

Theorem 4. *Let* $\mathbf{N}$ *be a net system and let* b *a place in its unfolding* $\mathcal{U}(\mathbf{N})$.

(i) If there are transitions h', l' *in* $\mathcal{U}(\mathbf{N})$ *such that* $b \in {}^\bullet l' \cap h'^+$, *then* $\pi_1(b)$ *is a weak causal place in* $\mathbf{N}$;
(ii) if there are transitions h', l' *in* $\mathcal{U}(\mathbf{N})$ *such that* $b \in {}^\bullet l' \cap h'^-$ *and* $[h'] \cup [l'] \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$, *then* $\pi_1(b)$ *is a weak conflict place in* $\mathbf{N}$.

Putting Theorem 3 and Theorem 4 together we get a characterisation of BNDC for safe nets in terms of the absence, in the unfolding, of places satisfying certain structural conditions.

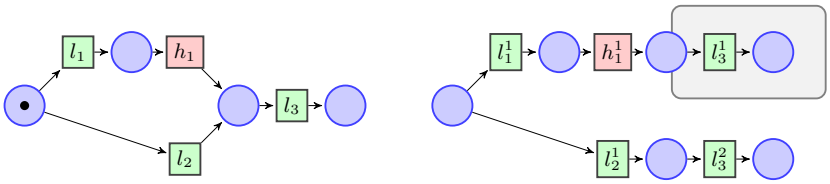

Fig. 5. A net system and a marking complete prefix not including an interference

Theorem 5 (BNDC in the unfolding of safe nets). *A safe net system* $\mathbf{N}$ *is not BNDC iff there are transitions* h', l' *and a place* b *in* $\mathcal{U}(\mathbf{N})$ *such that either (1)* $b \in {}^\bullet l' \cap h'^+$ *or (2)* $b \in {}^\bullet l' \cap h'^-$ *and* $[h') \cup [l'] \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$.

Remark 1. From the algorithmic point of view condition (1) above is very local and easy to check, while condition (2) involves the exploration of event histories. Notably, if we also assume that $\mathbf{N}$ does not contain self-loops on high transitions, it is no longer necessary to check that $[h') \cup [l']$ is a conflict-free set so that condition (2) becomes local. Further insights will be given at the end of the section, but a thorough treatment of this case cannot be included here for space limitations and is deferred to an extended version of this paper.

5.2 Complete Prefixes for Interferences

A *prefix* of an occurrence net N is a causally closed subnet of N. In McMillan's seminal work [12] the focus is on reachability and a prefix U of the unfolding is deemed *complete* when any marking reachable in $\mathbf{N}$ is in the image of the marking produced by some configuration of U. This has been later extended to more general properties [13].

A prefix which is complete for reachability (and also for executability of transitions), could not include relevant information for interferences. Consider for instance the net in Fig. 5 (left) and its unfolding on the right. The unfolding prefix not including the boxed part is marking complete, but it excludes a direct causality between h_1^1 and l_3^1, which witnesses the presence of a weak causal place and thus the fact that the net system is not BNDC.

In order to capture all interferences, our idea is to build a prefix of the unfolding which contains a variety of configurations sufficient to witness all markings of the original net, taking into account also the confidentiality level of transitions producing the tokens. More precisely, we associate with each configuration C a so-called *extended marking*, $\mathsf{M}^*(C) = \langle \mathsf{M}(C), \mathsf{H}(C) \rangle$, where $\mathsf{H}(C) = \{\pi_1(b) \in P : \exists h' \in H^{(\omega)}.\ b \in \pi_1(h')^+ \cap C^\circ\}$. In words $\mathsf{H}(C)$ is the set of places in the frontier of C that have been produced by some high transition that contributes positively. This extra information is needed to avoid the phenomenon illustrated in Fig. 5.

Definition 13 (e-complete prefix). *A prefix* U *of* $\mathcal{U}(\mathbf{N})$ *is* complete for extended marking reachability*, or simply* e-complete*, when for any configuration* $C \in \mathcal{C}(\mathcal{U}(\mathbf{N}))$ *there exists* $C' \in \mathcal{C}(U)$ *such that* $\mathsf{M}^*(C) = \mathsf{M}^*(C')$.

Given a safe net system $\mathbf{N}$, an e-complete prefix U of $\mathcal{U}(\mathbf{N})$ contains sufficient information for deciding whether or not $\mathbf{N}$ contains a weak causal place.

Theorem 6 (completeness for weak causal places). *Let* $\mathbf{N}$ *be a safe net system and let* U *be an e-complete prefix of* $\mathcal{U}(\mathbf{N})$*. Then* p *is a weak causal place in* $\mathbf{N}$ *iff there exist high and low transitions* h', l' *and a place* b *in* U *such that* $b \in {}^\bullet l' \cap h'^+$ *and* $\pi_1(b) = p$*.*

As a consequence of Proposition 1 and of the above theorem the BNDC property for safe net systems can be checked on an e-complete prefix of the unfolding of the causal reduct.

Corollary 1 (BNDC on e-complete prefixes). *Let* $\mathbf{N}$ *be a safe net system and let* U *be an e-complete prefix of* $\mathcal{U}(\gamma(\mathbf{N}))$*. Then* $\mathbf{N}$ *is not BNDC iff there exist high and low transitions* h', l' *in* U *and a place* $b \in {}^\bullet l' \cap h'^+$*.*

5.3 Unfolding-based Algorithms

Corollary 1 naturally leads to an algorithm for checking the BNDC property for safe net systems: it consists in generating an e-complete prefix of the unfolding of the causal reduct while checking for the presence of a place produced by a high transition and consumed by a low one.

A key notion for constructing an e-complete prefix is that of cut-off, which is roughly a transition in the unfolding that produces an extended marking already produced by other transitions with smaller history. The idea of cut-off events goes back to [12]. The word "smaller" is formalised by fixing an order $\prec$ on configurations. For example we can take $C \prec C'$ is $|C| < |C'|$, as in [12], but finer adequate orders (see [13]) can be considered for reducing the size of complete prefixes and improving the efficiency.

Definition 14 (cut-off). *An event* e *of the prefix* U *is called a* cut-off *when there exists another event* e' *such that* $\mathsf{M}^*([e']) = \mathsf{M}^*([e])$ *and* $[e'] \prec [e]$*.*

A procedure that unfolds a net system by adding events with $\prec$-minimal history and stopping at cut-offs produces an e-complete prefix of the unfolding. This follows from the general theory in [13].

On these bases we developed the algorithm in Fig. 6. It first computes the causal reduct $\gamma(\mathbf{N})$ of $\mathbf{N}$. Then it builds an e-complete prefix of the unfolding of $\gamma(\mathbf{N})$, looking, at each step, for the presence of direct causalities between high and low transitions satisfying the conditions in Theorem 6. The initial part of the unfolding is simply the initial marking of $\gamma(\mathbf{N})$. If U is an unfolding prefix, a *possible extension* of U is any transition $t' \in \gamma(T)^{(\omega)}$ such that ${}^\bullet t' \subseteq C^\circ$ for some configuration C of U. The set of possible extensions of U is denoted $PE(U)$.

Theorem 7 (correctness of the algorithm for safe nets). *Let* $\mathbf{N}$ *be a safe net system. Then the algorithm in Fig. 6 always terminates and provides the answer 'yes' iff* $\mathbf{N}$ *is BNDC.*

```
Data: A safe net system N.
compute γ(N)
U = γ(m0)
pe = PE(U)
while pe ≠ ∅ do
    take t ∈ pe such that [t] is ≺-minimal;
    if π1(t) ∈ L then
        if ∃h, b ∈ U. π1(h) ∈ H.
        b ∈ •t ∩ h+
        then
            return 'no'
        end
    end
    add t to U
    if t is not a cut-off then
        add t• to U
    end
    pe = PE(U)
end
return 'yes';
```

Fig. 6. Algorithm to decide BNDC on safe net systems

For example, when applied to the safe net system $\mathbf{R}$ in Fig. 1(a), the algorithm first computes the causal reduct $\gamma(\mathbf{R})$ of Fig. 1(b). Then it starts generating the unfolding $\mathcal{U}(\gamma(\mathbf{R}))$ of Fig. 4(b) until the weak causal place p_{h_2} and the transitions l_{2h_2}, h_{2c} are discovered, corresponding to the fact that $p \in {}^\bullet l_2 \cap h_2{}^-$ is a (weak) conflict place. At this point, the algorithm returns the answer 'no'.

The size of the e-complete prefix of $\mathcal{U}(\gamma(\mathbf{N}))$ produced by the algorithm in Fig. 6 and thus the efficiency of the algorithm are highly influenced by the "level of concurrency" in $\mathbf{N}$. In absence of concurrency the size of the prefix can equate that of the marking graph, which is exponential in the number of places. It has been shown in [13] that, when dealing with standard marking completeness, the use of a total adequate order ensures that the prefix will be never larger than the marking graph. Here the situation is similar, but the reference marking graph is that of the causal reduct of $\gamma(\mathbf{N})$, whose size is at most $|H|$ times the size of the marking graph of $\mathbf{N}$.

Consider for example a generalisation of $\mathbf{R}$ with k low and high components competing for the resource: the marking graph would be of size exponential in k, thus impacting on the performance of algorithms based on the interleaving semantics, while the size of an e-complete prefix would be of the order of k^2. We will consider these kind of examples in § 6, where experiments corroborate our hypothesis about the efficiency of our approach.

We conclude this section with a further explanation of Remark 1. If $\mathbf{N}$ is safe and without self-loops on high transitions we can prove that a place b in $\mathcal{U}(\mathbf{N})$ is the image of a conflict place of $\mathbf{N}$ iff there are a high and a low transition h', l' such that $b \in {}^\bullet l' \cap h'{}^-$ and $\mathsf{co}({}^\bullet l' \cup {}^\bullet h')$. Moreover Theorem 6 can be extended

saying that if such a place b occurs in $\mathcal{U}(\mathbf{N})$, then another place b', instance of the same conflict place, already occurs in an e-complete prefix U of $\mathcal{U}(\mathbf{N})$. For example, adding the check for conflict places, the modified algorithm would directly compute $\mathcal{U}(\mathbf{R})$ in Fig. 4(a) and then stop (returning the answer 'no') after generating transitions $h_1^1, l_1^1, h_2^1, l_2^1$ in the unfolding, since it identifies the place $p \in {}^\bullet l_2^1 \cap h_2^{1-}$, with the union of pre-sets ${}^\bullet l_2^1 \cup {}^\bullet h_2^1 = \{p_1, p, p_1'\}$ pairwise concurrent.

6 Experimental Results

We developed a prototype tool UBIC (Unfolding-Based Interference Checker) for checking the BNDC property for safe Petri nets according to the algorithm devised in this paper. The tool is based on CUNF [9], a toolset for unfolding and verifying low level nets with read arcs.

The tool UBIC takes as input a safe net $\mathbf{N}$, in the PEP ll_net format [15]. The confidentiality level of transitions is determined by the transition name: transitions whose name starts by "h" are of high level, and all the others are low level transitions. The tool constructs the causal reduct $\gamma(\mathbf{N})$ of the input net and then it starts the unfolding procedure. It can stop once the first weak causal place is found (if one is interested only in checking whether a net is or not BNDC) or it can produce a prefix of the unfolding of $\gamma(\mathbf{N})$ (when one is interested in complete information about the interferences in the net $\mathbf{N}$). A preliminary version of the tool UBIC is available at the page `http://www.math.unipd.it/~baldan/UBIC`.

Other algorithms have been proposed for checking BNDC on Petri nets and we are aware of two other tools that actually implement them. The tool Petri Net Security Checker (PNSC) [16] (written in Java), based on the work [5], basically generates the marking graph of the input net and then performs path searches in order to find interference situations.

In contrast, the tool ANICA [10] (Automated Non-Interference Check Assistant, written in C++) does not create the whole marking graph. Rather, it reduces each potential interfering place to a single reachability problem, which is then handled with the tool LOLA [17] (Low Level Petri Net Analyzer). According to the authors, this approach leads to a gain of efficiency with respect to PNSC, which allows ANICA to handle larger inputs. ANICA works on safe Petri nets.

Our tool UBIC (Unfolding-Based Interference Checker) provides very promising performances, especially in terms of speed. Here we illustrate some preliminary experimental results, in which we compare the performances of UBIC and ANICA on the same set of nets. All tests have been run on a laptop mounting an Intel®Core™i3 CPU (2.27GHz) and 3GB of RAM, in a Linux 13.10 environment. In each test we collect the execution time required by the two tools.

The tests have been conducted on three families of net systems:

(1) generalisations of the mutual exclusion net system $\mathbf{R}$ (see Fig. 1(a)), obtained by considering variants $\mathbf{R}(m, n)$, parametric with respect to the numbers m and n of high and low level components, respectively, competing for

Table 1. Experimental results on the parametric variants of the net system **R** in Fig. 1(a)

			UBIC	ANICA
Input net	**Places**	**Transitions**	**Time(sec)**	**Time(sec)**
$\mathbf{R}(5,5)$	34	32	0.005	0.060
$\mathbf{R}(10,10)$	64	62	0.016	0.490
$\mathbf{R}(15,15)$	94	92	0.018	1.950
$\mathbf{R}(20,20)$	124	122	0.043	4.880
$\mathbf{R}(25,25)$	154	152	0.086	10.650
$\mathbf{R}(30,30)$	184	182	0.177	16.220
$\mathbf{R}(35,35)$	214	212	0.337	24.380
$\mathbf{R}(50,50)$	304	302	1.708	76.080
$\mathbf{R}(75,75)$	454	452	9.251	324.190
$\mathbf{R}(100,100)$	604	602	29.618	929.330

the resource. Net $\mathbf{R}(m,n)$ has $4+3(n+m)$ places and $2+3(n+m)$ transitions. Table 1 contains the results concerning these nets.

(2) "line-shaped" nets consisting of a sequence of low transitions ending with a high transition (see Fig. 7). The net $\mathbf{L}(n)$ contains $n-1$ low transitions and one high transition. Table 2 contains the results concerning these nets.

(3) nets $\mathbf{K}(m)$ implementing Dijkstra's mutual exclusion algorithm [18], parametric with respect to the number m of processes competing for the resource. Net $\mathbf{K}(m)$ has $12m$ places and $3m+3m^2$ transitions. In each example we inserted exactly one high transition. This example is taken from the set of experiments provided with the toolset Cunf. In each net exactly one transition has been made high. Table 3 contains the results concerning these nets.

In all the examples we required the tools to stop as soon as the answer "yes BNDC" or "no BNDC" is found. These example nets are provided with the UBIC package. For example the net $\mathbf{R}(5,5)$ is encoded in file `netR5-5.ll_net` for UBIC and in file `netR5-5.ifn` for ANICA. The net $\mathbf{L}(5)$ is in file `netLine-5.ll_net` for UBIC, and in file `netLine-5.ifn` for ANICA. Similarly, net $\mathbf{K}(5)$ is in file `netDijkstra-5.ll_net` for UBIC and in file `netDijkstra-5.ifn` for ANICA.

Let us first focus on Table 1, which reports the experimental results for deciding BNDC on $\mathbf{R}(m,n)$, with UBIC and ANICA. All these nets are not BNDC. Note that UBIC is uniformly more efficient than ANICA and, although this is not reported, the difference becomes even more evident when full information about interferences (complete prefix for UBIC and all causal/conflict places for ANICA) are looked for.

The second class of examples, namely the nets $\mathbf{L}(n)$ are those for which the performances of the two tools differ the least, because there is no concurrency at all, so the unfolding of these nets is isomorphic to the marking graph. All these nets, but $\mathbf{L}(1)$, are not BNDC.

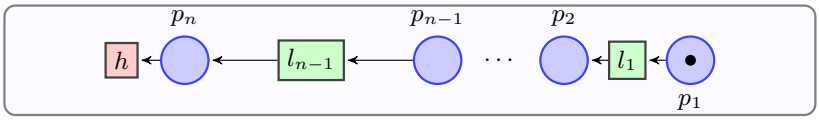

Fig. 7. Line-shaped net system $\mathbf{L}(n)$

Table 2. Experimental results on line-shaped net in Fig. 7 with variable size

			UBIC	ANICA
Input net	**Places**	**Transitions**	**Time(sec)**	**Time(sec)**
$\mathbf{L}(50)$	53	52	0.004	0.000
$\mathbf{L}(75)$	78	77	0.004	0.010
$\mathbf{L}(100)$	103	102	0.005	0.010
$\mathbf{L}(200)$	203	202	0.009	0.030
$\mathbf{L}(300)$	303	302	0.015	0.050
$\mathbf{L}(400)$	403	402	0.016	0.080
$\mathbf{L}(1000)$	1003	1002	0.104	0.430
$\mathbf{L}(2000)$	2003	2002	0.424	2.340
$\mathbf{L}(3000)$	3003	3002	1.057	5.390
$\mathbf{L}(4000)$	4003	4002	2.224	10.020
$\mathbf{L}(5000)$	5003	5002	4.261	15.660
$\mathbf{L}(10000)$	10003	10002	31.642	78.410

Finally, the last set of tests on the nets $\mathbf{K}(m)$ (implementing Dijkstra's mutual exclusion algorithm with m processes) produced the results in Table 3. All these nets, but $\mathbf{K}(1)$, are not BNDC. These experiments confirm that the efficiency of unfolding-based algorithms tend to increase when the system is highly concurrent. In the case of the nets $\mathbf{K}(m)$ the performance gap is so large that, for $m \geq 8$, such nets could not be processed within one entire night by ANICA.

We also performed a preliminary comparison concerning memory usage for the tests described above. The comparison seems to reveal that on "small" input nets UBIC requires less memory than ANICA, but as the size of input nets grows, the increase of the memory required by ANICA is less than proportional. The memory required by UBIC instead is essentially proportional to the number of transitions. As a consequence for large inputs ANICA requires less memory than UBIC. A proper analysis of memory usage is not trivial and some further investigation is needed (e.g., ANICA throws subprocesses and we did not yet manage to analyse the use of memory of both tools in a uniform way). Indeed ANICA first isolates all potential interfering places and then for each one creates a reachability problem which is then passed over to LOLA. The memory used for processing a reachability problem can then be reused for the next one, until all potential interfering places have been analysed. In contrast, UBIC has a "global" approach: it checks all places while they are being added to the growing unfolding of the input net. Hence the memory is released only when the entire process stops.

Table 3. Experimental results on the net systems $\mathbf{K}(m)$

			UBIC	ANICA
Input net	**Places**	**Transitions**	**Time(sec)**	**Time(sec)**
$\mathbf{K}(1)$	12	6	0.000	0.000
$\mathbf{K}(2)$	24	18	0.000	0.004
$\mathbf{K}(3)$	36	36	0.003	0.016
$\mathbf{K}(4)$	48	60	0.003	0.028
$\mathbf{K}(5)$	60	90	0.006	2.608
$\mathbf{K}(6)$	72	126	0.010	42.136
$\mathbf{K}(7)$	84	168	0.019	608.856
$\mathbf{K}(8)$	96	216	0.020	$\perp$
$\mathbf{K}(9)$	108	270	0.025	$\perp$
$\mathbf{K}(10)$	120	330	0.040	$\perp$
$\mathbf{K}(20)$	240	1260	1.292	$\perp$
$\mathbf{K}(30)$	360	2790	16.257	$\perp$
$\mathbf{K}(40)$	480	4920	156.210	$\perp$
$\mathbf{K}(50)$	600	7650	728.522	$\perp$

7 Conclusions and Future Work

Non-interference is a strong property, and often proper weakenings of it are studied (e.g., based on declassification or downgrading). Still, it plays a basic role as a reference, conceptual notion.

We provided a "causal" characterisation of non-interference by focusing on Petri nets and on Bisimilarity-based Non-Deducibility on Composition (BNDC), a classical formalisation of non-interference. For the class of safe nets, we characterised BNDC on the unfolding, in terms of causalities and conflicts between high and low level activities. More general results, which apply also to non-safe nets, have not been included for space limitations and are deferred to an extended version of this paper.

Our work led to an algorithm for checking BNDC based on the generation of a suitable finite prefix of the unfolding. Other algorithms exist for checking BNDC on Petri nets and we are aware of two tools that actually implement them. The tool Petri Net Security Checker [16] (written in Java), based on the work [5], basically generates the marking graph of the input net and then performs path searches in order to find interference situations. In contrast, the tool ANICA [10] (Automated Non-Interference Check Assistant, written in C++) does not create the whole marking graph. Rather, it reduces each potential interfering place to a single reachability problem, which is then handled with the tool LOLA [17] (Low Level Petri Net Analyzer). According to the authors, their technique allows not only to handle larger inputs, but also to parallelize the checks. ANICA works on safe Petri nets.

We developed a prototype tool UBIC (Unfolding-Based Interference Checker) based on CUNF [9], that provides very promising results, in terms of time efficiency, when compared to ANICA. Some preliminary experiments reveal a gain which can reach orders of magnitude. This fact, in some cases, allows us to treat much larger examples, on which ANICA does not respond at all within hours. We believe that resorting to true concurrent semantics of Petri nets really helps in devising efficient algorithms, alleviating the state explosion problem.

The idea of using a true concurrent semantics to check the possibility of deducing the occurrence of non-observable transitions from the occurrence of observable ones has been proposed in [19], in the context of asynchronous diagnosis of discrete event event systems. The *diagnosability* properties studied in [19] - or better, their negations - are similar in spirit to noninterference properties. Although non-diagnosability and BNDC properties surely differ (e.g., the former are trace-based while the latter is bisimulation-based), the relation between diagnosability and noninterference properties deserves to be deepened.

As future lines of research, we intend to explore the possibility of providing causal characterisations of known notions of non-interference, e.g., including downgrading, possibly defined on other formalisms. E.g., for imperative languages, by considering Petri net encodings, where the control and data flow is suitably captured by causality. Or for process calculi, for which there are several Petri net encodings (see, e.g., [20,21,22]).

Another venue of research consists in considering formalisations of noninterference obtained from the classical ones, by replacing interleaving observational semantics with true-concurrent ones [23]. The higher distinguishing power of such semantics could allow to identify new forms of interference which cannot be captured in an interleaving setting.

Acknowledgements. We are grateful to César Rodríguez for developing the tool CUNF and for his suggestions on its use for producing UBIC. We also thank the anonymous reviewers who contributed to the improvement of this paper with their valuable suggestions.

References

1. Goguen, J.A., Meseguer, J.: Security policies and security models. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 11–20. IEEE Computer Society (1982)
2. Focardi, R., Gorrieri, R.: Classification of security properties (Part I: Information flow). In: Focardi, R., Gorrieri, R. (eds.) FOSAD 2000. LNCS, vol. 2171, pp. 331–396. Springer, Heidelberg (2001)
3. Ryan, P., Schneider, Y.: Process algebra and non-interference. Journal of Computer Security 9, 75–103 (2001)
4. Mantel, H.: Possibilistic definitions of security - an assembly kit. In: CSFW, pp. 185–199. IEEE Computer Society (2000)
5. Busi, N., Gorrieri, R.: Structural non-interference in elementary and trace nets. Mathematical Structures in Computer Science 19, 1065–1090 (2009)

6. Best, E., Darondeau, P., Gorrieri, R.: On the decidability of non interference over unbounded Petri nets. In: Proceedings of SecCo 2010. EPTCS, vol. 51, pp. 16–33 (2010)
7. Nielsen, M., Plotkin, G., Winskel, G.: Petri Nets, Event Structures and Domains, Part 1. TCS 13, 85–108 (1981)
8. Esparza, J., Heljanko, K.: Unfoldings - A Partial order Approach to Model Checking. EACTS Monographs in Theoretical Computer Science. Springer (2008)
9. Rodríguez, C., Schwoon, S.: Cunf: A tool for unfolding and verifying Petri nets with read arcs. In: Van Hung, D., Ogawa, M. (eds.) ATVA 2013. LNCS, vol. 8172, pp. 492–495. Springer, Heidelberg (2013)
10. Accorsi, R., Lehmann, A.: Automatic information flow analysis of business process models. In: Barros, A., Gal, A., Kindler, E. (eds.) BPM 2012. LNCS, vol. 7481, pp. 172–187. Springer, Heidelberg (2012)
11. Frau, S., Gorrieri, R., Ferigato, C.: Petri net security checker: Structural non-interference at work. In: Degano, P., Guttman, J., Martinelli, F. (eds.) FAST 2008. LNCS, vol. 5491, pp. 210–225. Springer, Heidelberg (2009)
12. McMillan, K.L.: A technique of state space search based on unfolding. Form. Methods Syst. Des. 6, 45–65 (1995)
13. Khomenko, V., Koutny, M., Vogler, W.: Canonical prefixes of Petri net unfoldings. Acta Informatica 40, 95–118 (2003)
14. Meseguer, J., Montanari, U., Sassone, V.: Representation theorems for petri nets. In: Freksa, C., Jantzen, M., Valk, R. (eds.) Foundations of Computer Science. LNCS, vol. 1337, pp. 239–249. Springer, Heidelberg (1997)
15. Best, E., Grahlmann, B.: PEP Documentation and User Guide 1.8 (1998)
16. Frau, S., Gorrieri, R., Ferigato, C.: Petri net security checker: Structural non-interference at work. In: Degano, P., Guttman, J., Martinelli, F. (eds.) FAST 2008. LNCS, vol. 5491, pp. 210–225. Springer, Heidelberg (2009)
17. Wolf, K.: Generating Petri net state spaces. In: Kleijn, J., Yakovlev, A. (eds.) ICATPN 2007. LNCS, vol. 4546, pp. 29–42. Springer, Heidelberg (2007)
18. Dijkstra, E.: Solution of a problem in concurrent programming control. Communication of the ACM 8, 569 (1965)
19. Haar, S.: Types of asynchronous diagnosability and the reveals-relation in occurrence nets. IEEE Transactions on Automatic Control 55, 2310–2320 (2010)
20. Gorrieri, R., Montanari, U.: SCONE: A simple calculus of nets. In: Baeten, J.C.M., Klop, J.W. (eds.) CONCUR 1990. LNCS, vol. 458, pp. 2–30. Springer, Heidelberg (1990)
21. Devillers, R., Klaudel, H., Koutny, M.: A compositional Petri net translation of general pi-calculus terms. Formal Asp. Comput. 20, 429–450 (2008)
22. Meyer, R., Khomenko, V., Hüchting, R.: A polynomial translation of π-calculus (FCP) to safe Petri nets. In: Koutny, M., Ulidowski, I. (eds.) CONCUR 2012. LNCS, vol. 7454, pp. 440–455. Springer, Heidelberg (2012)
23. van Glabbeek, R., Goltz, U.: Refinement of actions and equivalence notions for concurrent systems. Acta Informatica 37, 229–327 (2001)