Model-checking bisimulation-based information flow properties for pushdown systems

Deepak D'Souza, K. R. Raghavendra

Department of Computer Sc. & Automation, Indian Institute of Science, India
{deepakd, raghavendrakr}@csa.iisc.ernet.in

Abstract. Bisimulation-based information flow properties were introduced by Focardi and Gorrieri [5] as a way of specifying security properties for transition system models. These properties were shown to be decidable for finite-state systems. In this paper, we study the problem of verifying these properties for some well-known classes of infinite state systems. We show that all the properties are undecidable for each of these classes of systems.

Keywords: model-checking, pushdown systems, bisimulation, information flow

1 Introduction

Information flow properties are a way of specifying security properties of systems, dating back to the work of Goguen and Meseguer [7] in the eighties. In this framework, a system is modelled as having high-level (or confidential) events as well as low-level (or public) events, and a typical property requires that the high-level events should not *influence* the outcome of low-level events. In other words, the sequence of low-level events observed from a system execution, should not reveal "too much" information about the high-level events that may have taken place during the execution.

There is a great variety of information flow properties proposed in the literature and can be broadly classified into the following categories. The original formulation of *non-interference* by Goguen and Meseguer was *state-based* in the sense that it spoke about the state of the system after a sequence of events: the state reached by the system after executing a sequence of low and high-level events, must be the same (from the low-level observer's point of view) as the state reached after executing only the low-level events in the sequence. As non-interference is often too strong a requirement (for example a typical password checking program is interferent), many relaxations to non-interference have been proposed in the literature. Some information flow properties are *trace-based* in that they specify information flow security as a property of the set of traces or executions produced by the system and its variants. For example, the *strong non-deterministic non-interference* (SNNI) property [5] states that the set of traces after hiding high-level events (replacing them with ϵ-transitions) should be the same as the set of traces with high-level events deleted. This corresponds to

non-inference of the occurrence of high-level events, as every low-level observation of a trace is itself a possible trace in the system. Finally there are properties based on the *structure* of the system model. For example, the property *Bisimulation-based Strong Non-deterministic Non-interference (BSNNI)* is the same as SNNI except that the check is on bisimulation equivalence rather than trace equivalence. These properties are termed bisimulation based information flow properties and are studied by Focardi and Gorrieri in [5].

We motivate bisimulation-based information flow properties with an example adapted from Focardi and Gorrieri [5]. Consider the component of an access-control system implementing the *no read up* policy as described by the state transition system in Fig. 1. The transition lRl represents a low user requesting to read a low object. Similarly lRh, hRl and hRh represent low reading high, high reading low and high reading high requests respectively. The acc_grant_l and acc_grant_h transitions grant the read access request originating from low and high users respectively. The acc_deny_l transition denies the read access request from a low user on a high object. Here the events lRl, lRh, acc_grant_l and acc_deny_l are low events and hRl, hRh and acc_grant_h are high events. The attacker observes only the low-level events in any execution of the system. We want the semantic property of *non-inference*: the attacker should not be able to infer the occurrence or non-occurrence of high-level events in any system execution.

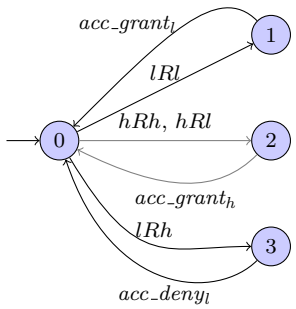

Fig. 1. Implementation of *no read up* without high interrupts

It is easy to see that the system satisfies the property of non-inference. This system satisfies both SNNI and BSNNI.

Consider a slight modification of the example with high-level interrupts as shown in Fig. 2. High-level interrupts h_stop_1, h_stop_2, h_stop_3 and h_stop_4 when fired halts the system by taking it to a trap-state. This system satisfies SNNI but not non-inference. A low user can never conclude that a high-level interrupt has been executed; however when he asks to reads a low object and if he sees acc_grant_l then he knows that the h_stop_1 event did not happen. This subtle

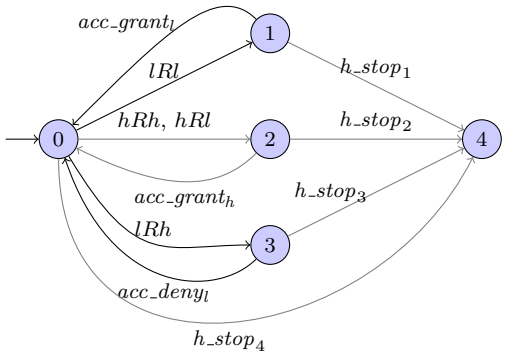

Fig. 2. Implementation of *no read up* with high interrupts

information flow can be exploited in order to construct an information channel from high level to low level. In order to detect these kind of flows, bisimulation-based information flow properties are used. As we show in Section 2, the system in Fig. 2 does not satisfy BSNNI.

In general bisimulation-based equivalence is a finer equivalence than trace-based equivalence and detects possible high level deadlocks that can compromise the security of the system [5]. The problem of checking bisimulation-based properties has been shown to be decidable for finite-state systems and has been implemented in a tool called *CoSec* [6, 1].

The problem of model-checking most of the known trace-based information flow properties is shown to be decidable [3]. However the problem of model-checking these trace-based properties for pushdown systems is shown to be undecidable [2].

A natural question that arises is whether the bisimulation-based properties continue to be decidable for well-known classes of infinite state systems like pushdown systems, Petri nets and process algebras [13]. We show in this paper that the problem of checking any of these bisimulation properties is undecidable for each of these classes of systems. To show these, we adapt the proofs by

- Srba [14] showing the undecidability of checking weak bisimilarity for pushdown systems.
- Jancar [10] showing the undecidability of checking strong bisimilarity for Petri nets.
- Srba [15] showing the undecidability of checking weak bisimilarity for process algebras.

We note that the problem of checking bisimulation-based properties appears to be weaker than the problem of checking bisimilarity for given classes of systems, in the sense that the former reduces to the latter in the case when the

class is closed under the hiding and deletion of transitions. However, our results nonetheless show that the problem of checking these bisimulation-based properties continues to be undecidable for the classes mentioned above.

2 Bisimulation relations and games

We begin by defining the basic system model of labelled transition systems. For binary relations R and S, we denote relational composition and reflexive transitive closure by $R \cdot S$ and R^* respectively. For an alphabet Σ, we use Σ^* to denote the set of all finite words on Σ. The concatenation of two words u and v will be denoted by $u \cdot v$ or simply uv.

A *labelled transition system (LTS)* M is a tuple $(Q, \Sigma, \rightarrow, s_0)$ where Q is a set of states, Σ is a set of labels, $\rightarrow \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ is a set of labelled transitions and $s_0 \in Q$ is the initial state. We sometimes write $s \xrightarrow{a} t$ instead of $(s, a, t) \in \rightarrow$. For $q \in Q$, we write M_q to denote the LTS $(Q, \Sigma, \rightarrow, q)$. For $c \in \Sigma \cup \{\epsilon\}$, we define $\xrightarrow{c}= \{(s,t) \mid s \xrightarrow{c} t\}$. The *weak transition relation* $(\Rightarrow)$ induced by M is defined as follows. Let $c \in \Sigma \cup \{\epsilon\}$:

$$\stackrel{c}{\Rightarrow}= \begin{cases} \xrightarrow{\epsilon}{}^* \cdot \xrightarrow{c} \cdot \xrightarrow{\epsilon}{}^* & \text{if } c \in \Sigma \\ \xrightarrow{\epsilon}{}^* & \text{if } c = \epsilon. \end{cases}$$

The language generated by M, denoted by $L(M)$, is the set $\{a_1 a_2 \cdots a_n \in \Sigma^* \mid \exists s_1, s_2, \ldots, s_n,\ s \stackrel{a_1}{\Rightarrow} s_1 \stackrel{a_2}{\Rightarrow} \cdots \stackrel{a_n}{\Rightarrow} s_n \}$.

Let $M_1 = (Q_1, \Sigma, \rightarrow_1, s_1)$ and $M_2 = (Q_2, \Sigma, \rightarrow_2, s_2)$ be two LTS's. A relation $R \subseteq Q_1 \times Q_2$ is a *weak bisimulation* between M_1 and M_2 if and only if whenever $(s,t) \in R$ and $s \xrightarrow{c}_1 s'$ with $c \in \Sigma \cup \{\epsilon\}$ then there exists $t' \in Q_2$ such that $t \stackrel{c}{\Rightarrow}_2 t'$ and $(s', t') \in R$ and conversely, whenever $t \xrightarrow{c}_2 t'$ with $c \in \Sigma \cup \{\epsilon\}$ then there exists $s' \in Q_1$ such that $s \stackrel{c}{\Rightarrow}_1 s'$ and $(s', t') \in R$. For $p_1 \in Q_1$, $p_2 \in Q_2$, we write $p_1 \approx p_2$ if and only if there exists a weak bisimulation containing (p_1, p_2). M_1 is said to be weakly bisimilar to M_2, written $M_1 \approx M_2$, if and only if $s_1 \approx s_2$. It is easy to see that the union R_{max} of all weak bisimulations between M_1 and M_2 is also a weak bisimulation. Two states p and q of an LTS M are said to be weakly bisimilar, written $p \approx q$, if and only if there is a weak bisimulation between two copies of M containing (p, q).

Weak bisimilarity has an elegant characterisation in terms of *bisimulation games*. Though the results in the section are folklore in the literature, the details are not readily availabile in our experience. Hence we include the proofs of these results.

Definition 1. *Let p_1 and p_2 be two states in LTS M_1 and M_2 respectively. A* bisimulation game *starting from p_1 and p_2 is a game between two players:* an attacker *and* a defender. *The game is played in rounds. In each round the players change the current states q_1 and q_2 (initially p_1 and p_2) according to the following rule.*

1. *The attacker chooses an $i \in \{1, 2\}, c \in \Sigma \cup \{\epsilon\}$ and $q_i' \in Q_i$ such that $q_i \xrightarrow{c}_i q_i'$.*

2. *The defender responds by choosing a* $q'_{3-i} \in Q_{3-i}$ *such that* $q_{3-i} \stackrel{c}{\Rightarrow}_{3-i} q'_{3-i}$.
3. *The states* q'_1 *and* q'_2 *become the current states.*

Let $K = (\{1,2\} \times \Sigma \times Q_1 \times Q_2) \cdot (Q_1 \times Q_2)$. *A finite play is a string in the language* $(Q_1 \times Q_2) \cdot K^*$. *An infinite play is a string in* $(Q_1 \times Q_2) \cdot K^\omega$. *The positions* $1+2i$, $i \geq 0$, *in a play, are the positions of the attacker (where it is his turn to make a move). A valid move by the attacker extends this string with an element from* $(\{1,2\} \times \Sigma \times Q_1 \times Q_2)$ *representing his selection of the component and the transition. The positions* $2i$, $i \geq 0$, *in a play, are the positions of the defender. A valid move by the defender extends this string with an element from* $(Q_1 \times Q_2)$ *representing his selection of the transition obeying the above rule. A play is valid if and only if it is formed by the alternate sequence of valid moves from the attacker and the defender. Let the set of valid plays be denoted by Plays. Let* $Plays_A$ *and* $Plays_D$ *denote the set of valid plays ending with the attacker's position and the defender's position respectively. Then* $Plays = Plays_A \uplus Plays_D$. *A partial map associating valid moves to plays in* $Plays_P$, *is a strategy for the player P. A play* α *is according to a strategy* π *of a player P if and only if at every position of the player P in* α, *the move prescribed by* π *is taken. A strategy* π *is valid for a player P if and only if for every play in* $Plays_P$ *according to* π, π *is defined. A valid strategy for the defender is also a* winning strategy *for her. A valid strategy* $f\pi$ *for the attacker is winning if and only if there is no infinite play according to* π. *A valid strategy* π *of a player P naturally induces a* pruned tree t_π *as follows. A node in* t_π *corresponds to a valid play (an element of Plays). Edges correspond to the valid moves of the attacker and the defender. The edges include only – all possible valid moves for P's opponent and the moves prescribed by* π *for P.*

For the rest of the section, let $M_1 = (Q_1, \Sigma, \rightarrow_1, s_1)$ and $M_2 = (Q_2, \Sigma, \rightarrow_2, s_2)$ be two LTS's with countable number of states. Let p and q be states in M_1 and M_2 respectively. It is easy to see that:

Lemma 1. *In any bisimulation game on* M_1 *and* M_2 *at most one of the players has a winning strategy.* □

Lemma 2. *The states p and q are bisimilar i.e.,* $p \approx q$ *iff the defender has a winning strategy in the bisimulation game starting from (p,q).*

Proof. ($\Leftarrow$:) Suppose the defender has a winning strategy π. Consider the induced pruned tree t_π. Let R be the binary relation on states of M_1 and M_2 containing only the nodes corresponding to the attacker's positions in t_π. Since π is a valid strategy for the defender, for every pair $(q_1, q_2) \in R$, for every $q_1 \stackrel{c}{\rightarrow} q'_1$ (or $q_2 \stackrel{c}{\rightarrow} q'_2$) with $c \in \Sigma \cup \{\epsilon\}$, we have $q_2 \stackrel{c}{\Rightarrow} q'_2$ ($q_1 \stackrel{c}{\Rightarrow} q'_1$ respectively) with $(q'_1, q'_2) \in R$. Hence R is a weak bisimulation. Since $(p, q) \in R$, we have $p \approx q$.

($\Rightarrow$:) Suppose $p \approx q$. Let R be a weak bisimulation relation containing (p, q). We construct a winning strategy π for the defender. From the definition of weak bisimilarity, for every $p \stackrel{c}{\rightarrow} p'$, $c \in \Sigma \cup \{\epsilon\}$, we have $q \stackrel{c}{\Rightarrow} q'$ with $(p', q') \in R$ and for every $q \stackrel{c}{\rightarrow} q'$, we have $p \stackrel{c}{\Rightarrow} p'$ with $(q', p') \in R$. Then the winning strategy

π for the defender is simple. For every attacker's choice $p \xrightarrow{c} p'$ (or $q \xrightarrow{c} q'$) let the defender's move in π be $q \stackrel{c}{\Rightarrow} q'$ ($p \stackrel{c}{\Rightarrow} p'$ respectively). Since $(p', q') \in R$, the defender's strategy π may be extended similarly. Thus we have a winning strategy π for the defender. □

We now argue that the attacker has a winning strategy if and only if $p \not\approx q$. For systems with at most countably infinite state spaces, weak bisimilarity can be characterised via weak bisimulation approximants [8].

Definition 2. *The weak bisimulation approximants $\approx_\alpha$ between the LTS's M_1 and M_2, for all ordinals α are defined by (transfinite) induction as follows.*

- *$p \approx_0 q$ for all states $p \in Q_1$ and $q \in Q_2$.*
- *$p \approx_{\alpha+1} q$ iff*
  - *for each transition $p \xrightarrow{c}_1 p'$ in M_1 there is a transition $q \stackrel{c}{\Rightarrow}_2 q'$ in M_2 such that $p' \approx_\alpha q'$ and*
  - *for each transition $q \xrightarrow{c}_2 q'$ in M_2 there is a transition $p \stackrel{c}{\Rightarrow}_1 p'$ in M_1 such that $p' \approx_\alpha q'$.*
- *For all limit ordinals λ, $p \approx_\lambda q$ iff $p \approx_\alpha q$ for all $\alpha < \lambda$.*

Proposition 1 ([8]). *The states p and q are weakly bisimilar i.e., $p \approx q$ iff $p \approx_\alpha q$ for all ordinals α.*

Proof. ($\Leftarrow$:) Suppose $p \approx_{\alpha+1} q$ for all ordinals α. Then for every $p \xrightarrow{c} p'$ with $c \in \Sigma \cup \{\epsilon\}$, we have $q \stackrel{c}{\Rightarrow} q_\alpha$ such that $p' \approx_\alpha q_\alpha$. In the (transfinite) list q_0, $q_1, \ldots$ some state q' must appear infinitely often, as there are countably many states. Note that if $s \approx_\alpha t$ then $s \approx_\beta t$ for all $\beta \leq \alpha$. Hence we have q' such that $p' \approx_\alpha q'$ for all ordinals α. Hence $p \approx q$.

($\Rightarrow$:) Suppose $p \approx q$. We show that $p \approx_\alpha q$ for every ordinal α by transfinite induction on α. We know that $p \approx_0 q$.

We now show that $p \approx_{\alpha+1} q$ for every non-limit ordinal α. For every $p \xrightarrow{c} p'$ with $c \in \Sigma \cup \{\epsilon\}$, we have $q \stackrel{c}{\Rightarrow} q'$ such that $p' \approx q'$. From induction hypothesis, we have $p' \approx_\alpha q'$. Hence $p \approx_{\alpha+1} q$.

We now show that $p \approx_\lambda q$ for every limit ordinal λ. For every $p \xrightarrow{c} p'$ with $c \in \Sigma \cup \{\epsilon\}$, we have $q \stackrel{c}{\Rightarrow} q'$ such that $p' \approx q'$. Again from induction hypothesis, we have $p' \approx_\beta q'$ for every $\beta < \lambda$. Hence we have $p \approx_\lambda q$. □

Lemma 3. *The states p and q are not weakly bisimilar i.e., $p \not\approx q$ iff the attacker has a winning strategy in the bisimulation game starting from (p,q).*

Proof. ($\Leftarrow$:) Suppose the attacker has a winning strategy. From Lemma 1, the defender does not have a winning strategy. From Lemma 2, $p \not\approx q$.

($\Rightarrow$:) Suppose $p \not\approx q$. From Proposition 1, let α be the smallest ordinal such that $p \not\approx_\alpha q$.

Suppose α is a limit ordinal. From Definition 2 there exists β with $\beta < \alpha$ such that $p \not\approx_\beta q$. This contradicts the fact that α was the smallest such oridinal. Hence α cannot be a limit ordinal.

We now construct the winning strategy for the attacker. Since $p \not\approx_\alpha q$, there exists a transition, without loss of generality say $p \xrightarrow{c}_1 p'$, $c \in \Sigma \cup \{\epsilon\}$, such that for all transitions of the form $q \stackrel{c}{\Rightarrow}_2 q'$, $p' \not\approx_{\alpha-1} q'$. Let the attacker choose the transition $p \xrightarrow{c}_1 p'$. Now either the defender is stuck or the defender responds by playing some $q \stackrel{c}{\Rightarrow}_2 q'$. By construction, $p' \not\approx_{\alpha-1} q'$. We continue to choose the attacker's move in a similar way. Since the set of ordinals is well-founded, after a finite number of moves, the defender must get stuck. Hence we have a winning strategy for the attacker. □

3 Bisimulation-based information flow properties

We recall different bisimulation-based information flow properties defined in the literature.

Let $M = (Q, \Sigma, \rightarrow, s)$ be an LTS and $X \subseteq \Sigma$. Then $M \setminus X$ denotes the LTS obtained from M by deleting all transitions labelled by elements in X. M/X denotes the LTS obtained from M by replacing all transitions labelled by elements in X with ϵ (silencing).

Let the set of events Σ (or synonymously actions) be partitioned into inputs (I) and outputs (O). Let Σ again be partitioned into high (H) and low (L) events. Each event a in Σ has a complementary action which we denote by $\bar{a}$ in Σ. We assume the sets H and L are closed under complementation i.e, $\bar{\mathsf{H}} = \{\bar{a} \mid a \in \mathsf{H}\} = \mathsf{H}$ and $\bar{\mathsf{L}} = \{\bar{a} \mid a \in \mathsf{L}\} = \mathsf{L}$. Let $\mathcal{E}_\mathsf{H}$ denote the set of all systems whose language over Σ is a subset of H^*.

Given $M_1 = (Q_1, \Sigma, \rightarrow_1, s_1)$, $M_2 = (Q_2, \Sigma, \rightarrow_2, s_2)$, the composition of M_1 and M_2 denoted by $M_1|M_2$ is defined to be $(Q_1 \times Q_2, \Sigma, \rightarrow, (s_1, s_2))$ where $(p, q) \xrightarrow{c} (p', q')$ if $p \xrightarrow{c}_1 p'$ or $q \xrightarrow{c}_2 q'$ and $(p, q) \xrightarrow{\epsilon} (p', q')$ if $p \xrightarrow{a}_1 p'$ and $q \xrightarrow{\bar{a}}_2 q'$.

The bisimulation-based information flow properties [5] are variants of the trace-based non-deterministic non-interference (NNI), a natural generalization of non-interference [7] to non-deterministic systems. The basic idea is that an LTS satisfies NNI when nothing about the execution of high input events leaks to the observation of a low-user. More precisely, an LTS M satisfies NNI if and only if $L((M \setminus (\mathsf{H} \cap \mathsf{I}))/\mathsf{H}) = L(M/\mathsf{H})$. The following defintions are taken from [5]. In the definition below we fix an LTS $M = (Q, \Sigma, \rightarrow, s)$ over Σ partitioned into I, O and H, L.

Definition 3. *a.* ***Bisimulation-based Non-deterministic Noninterference (BNNI).*** *M satisfies BNNI iff $M/\mathsf{H} \approx (M \setminus (\mathsf{I} \cap \mathsf{H}))/\mathsf{H}$.*

b. ***Bisimulation-based Strong Non-deterministic Non-interference (BSNNI).*** *M satisfies BSNNI iff $M/\mathsf{H} \approx M \setminus \mathsf{H}$.*

c. ***Bisimulation-based Non Deducibility on Compositions (BNDC).*** *M satisfies BNDC iff $\forall M' \in \mathcal{E}_\mathsf{H},\ M/\mathsf{H} \approx (M|M') \setminus \mathsf{H}$.*

d. ***Strong BNNI (SBNNI).*** *M satisfies SBNNI iff for all reachable states q in M, M_q satisfies BNNI.*

e. ***Strong BSNNI (SBSNNI).*** *M satisfies SBSNNI iff for all reachable states q in M, M_q satisfies BSNNI.*

f. ***Strong BNDC (SBNDC)****. M satisfies SBNDC iff for all reachable states q, r and for all $h \in \mathsf{H}$, such that $q \xrightarrow{h} r$ in M, $M_q \setminus \mathsf{H} \approx M_r \setminus \mathsf{H}$.*

There are other bisimulation-based properties proposed in [4] – persistent BNDC and dynamic BNDC. They both are shown to be equivalent to SBSNNI [4]. Hence we focus on the properties listed in Definition 3.

Consider the example LTS M in Fig. 2. We show that M does not satisfy BSNNI by describing the winning strategy for the attacker in the bisimulation game on $M \setminus \mathsf{H}$ and M/H. The attacker chooses the transition $0 \xrightarrow{\epsilon} 2$ in M/H. There are no ϵ-transitions from state 0 in $M \setminus \mathsf{H}$. Hence the defender is forced to stay at state 0. The attacker chooses the transition $2 \xrightarrow{\epsilon} 4$ in M/H. Again the defender is forced to stay at state 0. Now the attacker chooses the transition $0 \xrightarrow{lRl} 1$ in $M \setminus \mathsf{H}$. The defender is required to make a coresponding move from state 4 in M/H on lRl. As there is no such move, the attacker wins. Thus the attacker has a winning strategy and hence $M \setminus \mathsf{H} \not\approx M/\mathsf{H}$. Thus M does not satisfy BSNNI.

4 Model-checking Pushdown Systems

We now consider the problem of model-checking pushdown systems for bisimulation-based information flow properties. We first define some required notions.

Definition 4. *A* pushdown system (PDS) *is of the form $P = (Q, \Sigma, \Gamma, \rightarrow, s_0, S)$, where Q is a finite set of control states, Σ is a finite input alphabet, Γ is a finite stack alphabet, $\rightarrow \subseteq ((Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma) \times (Q \times \Gamma^*))$ is the transition relation, $s_0 \in Q$ is the starting state, and $S \in \Gamma$ is the initial stack symbol. If $((p, a, A), (q, B_1 B_2 \cdots B_k)) \in \rightarrow$, this means that whenever the machine is in state p with A on top of the stack, it can do an a-labelled transition to pop A off the stack, push $B_1 B_2 \cdots B_k$ onto the stack (such that B_1 becomes the new top of the stack symbol), and enter state q. If $((p, \epsilon, A), (q, B_1 B_2 \cdots B_k)) \in \rightarrow$, this means that whenever the machine is in state p with A on top of the stack, it can do an ϵ-labelled transition to pop A off the stack, push $B_1 B_2 \cdots B_k$ onto the stack and enter state q.*

A PDS $P = (Q, \Sigma, \Gamma, \longrightarrow, s_0, S)$ induces an LTS $M_P = (Q \times \Gamma^*, \Sigma, \rightarrow, (s_0, S))$. The *configurations* of P form the states of M_P. A configuration of P describes the current state and the current stack contents. Given a configuration $(p, A\beta)$ for some $A \in \Gamma$ and $\beta \in \Gamma^*$, the next configuration relation $\rightarrow$ on any $c \in \Sigma \cup \{\epsilon\}$ gives $(q, \gamma\beta)$ if $((p, c, A), (q, \gamma)) \in \longrightarrow$. This is written $(p, A\beta) \xrightarrow{c} (q, \gamma\beta)$. We will write a configuration of the form (p, α) as simply $p\alpha$ in the sequel.

The problem of model checking a bisimulation-based information flow property θ for PDS's is – given a PDS P, does M_P satisfy θ? We show that this problem is undecidable for each of the properties in Definition 3.

Srba in [14] shows that the problem of checking weak bisimilarity between two pushdown systems is undecidable. The idea is to reduce the halting problem of

Minsky machines with two counters to the problem of checking weak bisimilarity between two pushdown systems.

Definition 5. *A Minsky machine R with two counters c_1 and c_2 is a finite sequence $R = (L_1 : I_1, L_2 : I_2, \ldots, L_{n-1} : I_{n-1}, L_n : halt)$, where $n \geq 1, L_1, \ldots, L_n$ are pairwise different labels, and $I_1, \ldots, I_{n-1}$ are instructions of the following two types–* ***increment:*** *$c_r := c_r + 1$; goto L_j,* ***test and decrement:*** *if $c_r = 0$ then goto L_j else $c_r := c_r - 1$; goto L_k, where $1 \leq r \leq 2$ and $1 \leq j, k \leq n$. A configuration of a Minsky machine R is a triple (L_i, v_1, v_2) where L_i is the instruction label ($1 \leq i \leq n$), and v_1, v_2 are nonnegative integers representing the values of counters c_1 and c_2 respectively. The transition relation on configurations is defined in a natural way.*

The problem of deciding whether a Minsky machine R halts with an initial counter values set ot zero is undecidable [11].

Given a Minsky machine R with two counters c_1, c_2, Srba constructs a pushdown system P_R on a stack alphabet $\{C_1, C_2, S\}$ and two configurations of P_R: p_1S and p'_1S such that R halts if and only if $p_1S \approx p'_1S$. The proof idea is as follows. A configuration of R, (L_i, v_1, v_2), is represented by a pair of processes $p_i\gamma S$ and $p'_i\gamma' S$ where $\gamma, \gamma' \in \{C_1, C_2\}^*$ such that the number of occurrences of C_1 and C_2 in γ (and also in γ') is equal to v_1 and v_2 respectively. The instruction of the type $L_i : c_r := c_r + 1; \text{goto } L_j$, where $1 \leq j \leq n$ and $1 \leq r \leq 2$, is simulated by $p_iX \xrightarrow{a} p_jC_rX$ and $p'_iX \xrightarrow{a} p'_jC_rX$. To simulate a test and decrement instruction, say L_i : if $c_r = 0$ then goto L_j else $c_r := c_r - 1$; goto L_k, where $1 \leq j, k \leq n$ and $1 \leq r \leq 2$, consider the bisimulation game at $(p_i\gamma S, p'_i\gamma' S)$. The attacker forces the defender to rearrange the stack contents at γ and γ' such that C_r's are brought on top. Then C_r is popped if there is one at both γ and γ'. The crucial transition distinguishing p_n and p'_n is: $p_nX \xrightarrow{halt} p_nX$. When R halts, the attacker's aim is to reach p_n by faithfully simulating R's halting computation. Then he chooses *halt* transition for which the defender cannot match. Hence the attacker wins and $p_1S \not\approx p'_1S$. When R diverges, the defender forces the attacker to correctly simulate the moves of R. The attacker never reaches p_n, hence inducing an infinite game. Thus the defender wins and $p_1S \approx p'_1S$. The reader is referred to [14] for the detailed proof.

Theorem 1 ([14]). *R halts iff $p_1S \not\approx p'_1S$ in M_{P_R}.* □

From Srba's construction, we observe that:

1. p_1S has no ϵ-transitions
2. if there is a winning strategy for the attacker from (p_1S, p'_1S) then there is one from (p_1S, p'_1S) beginning with a transition from p_1S.

In general, let P be a PDS and $p_1\alpha$, $p_2\beta$ its configurations satisfying the conditions:

1. $p_1\alpha$ has no ϵ-transitions

2. if there is a winning strategy for the attacker from $(p_1\alpha, p_2\beta)$ then there is one from $(p_1\alpha, p_1\beta)$ beginning with a transition from $p_1\alpha$.

Then we call the problem of checking whether $p_1\alpha \approx p_2\beta$ the *restricted PDS bisimulation problem*. It follows then from the construction in [14] that:

Theorem 2. *The restricted PDS bisimulation problem is undecidable.* □

We now reduce the restricted PDS bisimulation problem to the problem of checking each of the bisimulation-based information flow properties for PDS's. Let the PDS $P = (Q, \Sigma, \Gamma, \longrightarrow, s_0, S)$ and its configurations $p_1\alpha$, $p_2\beta$ be an instance of the restricted PDS bisimulation problem. We construct P' from P such that $P' = (Q \cup \{s\}, \Sigma \cup \{k, \bar{k}\}, \Gamma, \longrightarrow', s, S)$ such that $s \notin Q$ and $\longrightarrow' = \longrightarrow \cup \{((s, k, S), (p_1, \alpha)), ((s, \epsilon, S), (p_2, \beta))\}$ where $k, \bar{k}$ are the only high (and input) events. That is $\mathsf{H} = \mathsf{I} = \{k, \bar{k}\}$. Informally, the induced LTS $M_{P'}$ of P' has a new start state sS with a high-event k edge – $sS \xrightarrow{k} p_1\alpha$ and an ϵ-edge – $sS \xrightarrow{\epsilon} p_2\beta$. The initial part of the induced LTS $M_{P'}$ is shown in Fig. 3. We fix the PDS P, its configurations $p_1\alpha$, $p_2\beta$ and the PDS P' constructed from P as described above for the rest of the section.

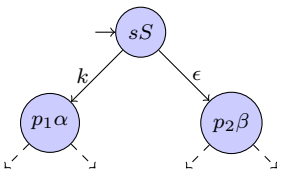

Fig. 3. $M_{P'}$

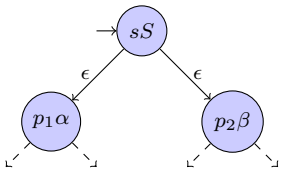

Fig. 4. $M_{P'}/\mathsf{H}$

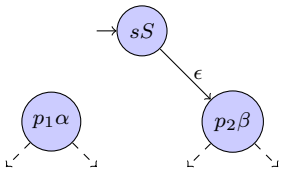

Fig. 5. $M_{P'} \setminus \mathsf{H}$

Lemma 4. *The configurations $p_1\alpha$ and $p_2\beta$ are weakly bisimilar i.e., $p_1\alpha \approx p_2\beta$ in M_P iff $M_{P'}$ satisfies BSNNI.*

Proof. ($\Leftarrow$:) Suppose $p_1\alpha \not\approx p_2\beta$ in M_P. Then we have a winning strategy π for the attacker from $p_1\alpha$ and $p_2\beta$ in M_P beginning with a move from $p_1\alpha$. We claim that the attacker has a winning strategy in the game starting at sS (of $M_{P'} \setminus \mathsf{H}$) and sS (of $M_{P'}/\mathsf{H}$). We now describe that strategy. The attacker chooses sS of $M_{P'}/\mathsf{H}$ and takes the edge ϵ to $p_1\alpha$ (Fig. 4). The defender now has to make a move from sS of $M_{P'} \setminus \mathsf{H}$ (Fig. 5) and has many choices.

- **Defender makes an ϵ-move to** $p_2\beta$. The attacker plays π and wins.
- **Defender stays at** sS. The attacker makes the first move according to π. From the definition of the restricted PDS bisimulation problem, the first move of π is a non-ϵ edge from $p_1\alpha$, say $p_1\alpha \xrightarrow{a} r$, for some state r in $M_{P'}/\mathsf{H}$. The defender is forced to respond with the same non-ϵ move from $p_2\beta$, say $sS \xrightarrow{\epsilon} p_2\beta \stackrel{a}{\Rightarrow} r'$ for some state r' in $M_{P'} \setminus \mathsf{H}$. Now the attacker can play according to π and win, since π also serves as the winning strategy for the attacker from (r, r').

– **Defender takes** $sS \xrightarrow{\epsilon} p_2\beta \stackrel{\epsilon}{\Rightarrow} q$. Note that the non-$\epsilon$ responses enabled at q are also enabled at $p_2\beta$. Hence the attacker can play according to π and win.

Thus the attacker has a winning strategy and hence sS (of $M_{P'} \setminus \mathsf{H}$) and sS (of $M_{P'}/\mathsf{H}$) are not weakly bisimilar. Thus $M_{P'}$ does not satisfy BSNNI.

($\Rightarrow$:) Suppose $p_1\alpha \approx p_2\beta$ in M_P. Then we have a winning strategy π for the defender from $p_1\alpha$ and $p_2\beta$ in M_P. We now describe the winning strategy for the defender from sS (of $M_{P'} \setminus \mathsf{H}$) and sS (of $M_{P'}/\mathsf{H}$). There are three cases:

– **Attacker chooses the transition** $sS \xrightarrow{\epsilon} p_1\alpha$ **in** $M_{P'}/\mathsf{H}$. The defender chooses $sS \xrightarrow{\epsilon} p_2\beta$ of $M_{P'} \setminus \mathsf{H}$ and thereafter plays π to win.
– **Attacker chooses the transition** $sS \xrightarrow{\epsilon} p_2\beta$ **in** $M_{P'}/\mathsf{H}$. The defender chooses $sS \xrightarrow{\epsilon} p_2\beta$ of $M_{P'} \setminus \mathsf{H}$ and imitates the attacker from here on. Either the attacker gets stuck or goes on to play the infinite bisimulation game. In both cases, the defender wins.
– **Attacker chooses** $sS \xrightarrow{\epsilon} p_2\beta$ **in** $M_{P'} \setminus \mathsf{H}$. The defender chooses $sS \xrightarrow{\epsilon} p_2\beta$ of $M_{P'}/\mathsf{H}$ and and imitates the attacker from here on. Either the attacker gets stuck or goes on to play the infinite bisimulation game. In both cases, the defender wins.

Thus the defender has a winning strategy and hence $M_{P'} \setminus \mathsf{H} \approx M_{P'}/\mathsf{H}$. Thus $M_{P'}$ satisfies BSNNI. □

Lemma 5. *$M_{P'}$ satisfies BSNNI iff $M_{P'}$ satisfies BNNI.*

Proof. As $\mathsf{H} = \mathsf{I} = \{k, \bar{k}\}$, we have $\mathsf{H} \cap \mathsf{I} = \mathsf{H}$. Hence $M_{P'} \setminus (\mathsf{H} \cap \mathsf{I}) = M_{P'} \setminus \mathsf{H}$. Hence $M_{P'}$ satisfies BNNI iff $M_{P'}$ satisfies BSNNI. □

We now consider the problem of checking BNDC for pushdown systems. Let $M = (Q_M, \mathsf{H}, \rightarrow_M, m_0)$ be any LTS in $\mathcal{E}_{\mathsf{H}}$. We define an equivalence relation $\equiv$ on the states of $(M_{P'}|M) \setminus \mathsf{H}$ as $(q\gamma, m) \equiv (q'\gamma', m')$ if and only if $q\gamma = q'\gamma'$. For every state $(q\gamma, m)$ of $(M_{P'}|M) \setminus \mathsf{H}$, let $[q\gamma] = \{(q'\gamma', m') \mid (q'\gamma', m') \equiv (q\gamma, m)\}$ denote its equivalence class. Let $N = (Q_N, \Sigma, \rightarrow_N, [sS])$ denote the quotient LTS $((M_{P'}|M) \setminus \mathsf{H})/\equiv$, where $Q_N = \{[q\gamma] \mid q\gamma \text{ is a state in } M_{P'}\}$, $[q\gamma] \xrightarrow{c}_N [q'\gamma']$, $c \in \Sigma \cup \{\epsilon\}$, if and only if there exist states m, m' in M such that $(q\gamma, m) \xrightarrow{c} (q'\gamma', m')$ in $(M_{P'}|M) \setminus \mathsf{H}$. Let N' be the LTS same as N with all the ϵ self loops deleted. The Fig. 6 shows a part of the LTS N'. The transition $[sS] \xrightarrow{\epsilon}_N [p_1\alpha]$ is represented using dotted arrow indicating that the transition may or may not be present. This transition is present if and only if there is a transition of the form $m_0 \stackrel{\bar{k}}{\Rightarrow}_M m$ for some state m. We note that M can have only transitions with labels $k, \bar{k}$ or ϵ. Thus the ϵ-transitions from M and the ϵ-transitions due to synchronization between M and $M_{P'}$ on $k, \bar{k}$, are the only possible contributions from M to N.

Let R and S be two LTS's. Let R^ϵ be any LTS constructed from R by adding ϵ self loops arbitrarily. Then it is easy to see that:

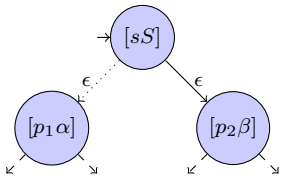

Fig. 6. N'

Lemma 6. *$R \approx S$ iff $R^{\epsilon} \approx S$.*

Lemma 7. *$(M_{P'}|M) \setminus \mathsf{H} \approx N'$.*

Proof. We construct the winning strategy for the defender. The strategy is essentially to mimic the moves of the attacker. The defender chooses to maintain the game at same positions (with respect to $M_{P'}$). That is, at any point the attacker starts from $((q\gamma, m)$ and $[q\gamma])$ where $q\gamma$ and m are states of $M_{P'}$ and M respectively. Consider the case when the attacker chooses the transition $[sS] \xrightarrow{\epsilon}_N [p_1\alpha]$ in N'. We observe that this happens only when M has a transition of the form $m_0 \stackrel{\bar{k}}{\Rightarrow}_M m$ for some state m. The defender chooses the transition $(sS, m_0) \stackrel{\epsilon}{\Rightarrow} (p_1\alpha, m)$ in $(M_{P'}|M) \setminus \mathsf{H}$, leaving the attacker to play from $[p_1\alpha]$ of N' and $(p_1\alpha, m)$ of $(M_{P'}|M) \setminus \mathsf{H}$. All the other cases are easy to see. Eventually either the attacker gets stuck or goes on to play the infinite bisimulation game. In both cases, the defender wins. Hence $(M_{P'}|M) \setminus \mathsf{H} \approx N'$. for the attacker's initial choices. □

Lemma 8. *The configurations $p_1\alpha$ and $p_2\beta$ are weakly bisimilar i.e., $p_1\alpha \approx p_2\beta$ in M_P iff $M_{P'}$ satisfies BNDC.*

Proof. ($\Leftarrow$:) Suppose $p_1\alpha \not\approx p_2\beta$ in M_P. From Lemma 4, we know that $M_{P'}$ does not satisfy BSNNI. That is, $M_{P'} \setminus \mathsf{H} \not\approx M_{P'}/\mathsf{H}$. Consider the LTS $M = (\{m\}, \mathsf{H}, \emptyset, m)$. We note that $M \in \mathcal{E}_{\mathsf{H}}$. It is easy to see that $M_{P'}|M$ is isomorphic to $M_{P'}$. This implies that $(M_{P'}|M) \setminus \mathsf{H} \not\approx M_{P'}/\mathsf{H}$. Hence $M_{P'}$ does not satisfy BNDC.

($\Rightarrow$:) Suppose $p_1\alpha \approx p_2\beta$ in M_P. Then we have a winning strategy π for the defender from $p_1\alpha$ and $p_2\beta$ in M_P. Let $M = (Q_M, \mathsf{H}, \rightarrow_M, m)$ be any LTS in $\mathcal{E}_{\mathsf{H}}$. From Lemma 7, we know that $(M_{P'}|M) \setminus \mathsf{H} \approx N'$. It is easy to see that the subtrees of $[p_1\alpha]$ and $[p_2\beta]$ in N' are isomorphic to the subtrees of $p_1\alpha$ and $p_2\beta$ in $M_{P'}$ respectively. We now show that $N' \approx M_{P'}/\mathsf{H}$. We construct a winning strategy for the defender. Consider the different cases for the attacker's choices.

- **Attacker chooses the transition** $[sS] \xrightarrow{\epsilon}_N [p_2\beta]$ **in** N'. The defender chooses $sS \stackrel{\epsilon}{\rightarrow} p_2\beta$ of $M_{P'}/\mathsf{H}$. The defender imitates the attacker choices (with respect to the states from $M_{P'}$) from here on. Either the attacker gets stuck or goes on to play the infinite bisimulation game. In both cases, the defender wins.

– **Attacker chooses the transition** $[sS] \xrightarrow{\epsilon}_N [p_1\alpha]$ **in** N'. The defender chooses $sS \xrightarrow{\epsilon} p_1\alpha$ from $M_{P'}/\mathsf{H}$. The defender imitates the attacker choices (with respect to the states of $M_{P'}$) from here on. Either the attacker gets stuck or goes on to play the infinite bisimulation game. In both cases, the defender wins.
– **Attacker chooses the transition** $sS \xrightarrow{\epsilon} p_2\beta$ **in** $M_{P'}/\mathsf{H}$. The defender chooses $[sS] \xrightarrow{\epsilon} [p_2\beta]$. The defender imitates the attacker choices (with respect to the states of $M_{P'}$) from here on. Either the attacker gets stuck or goes on to play the infinite bisimulation game. In both cases, the defender wins.
– **Attacker chooses the transition** $sS \xrightarrow{\epsilon} p_1\alpha$ **in** $M_{P'}/\mathsf{H}$. Note that there may not be a transition of the form $[sS] \xrightarrow{\epsilon}_N [p_1\alpha]$ as shown in Fig. 6. The defender chooses $[sS] \xrightarrow{\epsilon} [p_2\beta]$. The defender plays π from here on and wins.

Hence the defender has a winning strategy and thus $N' \approx M_{P'}/\mathsf{H}$. From Lemma 7 and the transitive property of $\approx$, we have $(M_{P'}|M) \setminus \mathsf{H} \approx M_{P'}/\mathsf{H}$ for any $M \in \mathcal{E}_H$. Thus $M_{P'}$ satisfies BNDC. □

It follows from Lemmas 4, 5 and 8 that the problem of checking BNNI, BSNNI and BNDC for pushdown systems is undecidable. Now we consider the properties SBNNI, SBSNNI and SBNDC.

Lemma 9. *The configurations $p_1\alpha$ and $p_2\beta$ are weakly bisimilar i.e., $p_1\alpha \approx p_2\beta$ in M_P iff $M_{P'}$ satisfies SBSNNI.*

Proof. ($\Leftarrow$:) Suppose $p_1\alpha \not\approx p_2\beta$ in M_P. Then from Lemma 4, $M_{P'}$ does not satisfy BSNNI. Hence $M_{P'}$ does not satisfy SBSNNI.

($\Rightarrow$:) Suppose $M_{P'}$ does not satisfy SBSNNI. Then there exists some state m in $M_{P'}$ such that m of $M_{P'} \setminus \mathsf{H}$ and m of $M_{P'}/\mathsf{H}$ are not weakly bisimilar. Then there exists a winning strategy π for the attacker from m of $M_{P'}/\mathsf{H}$ and m of $M_{P'} \setminus \mathsf{H}$. Note that there are no H-edges in $M_{P'}$ except for $sS \xrightarrow{k} p_1\alpha$. Hence for all states m other than sS, m of $M_{P'}/\mathsf{H}$ and m of $M_{P'} \setminus \mathsf{H}$ are weakly bisimilar. This implies that sS of $M_{P'}/\mathsf{H}$ and sS of $M_{P'} \setminus \mathsf{H}$ are not weakly bisimilar. Then $M_{P'}$ does not satisfy BSNNI. From Lemma 4, we have $p_1\alpha \not\approx p_2\beta$ in M_P. □

Lemma 10. *$M_{P'}$ satisfies SBSNNI iff $M_{P'}$ satisfies SBNNI.*

Proof. From Lemma 5, $M_{P'}$ satisfies BSNNI if and only if $M_{P'}$ satisfies BNNI. Hence $M_{P'}$ satisfies SBSNNI if and only if $M_{P'}$ satisfies SBNNI. □

Lemma 11. *The configurations $p_1\alpha$ and $p_2\beta$ are weakly bisimilar i.e., $p_1\alpha \approx p_2\beta$ in M_P iff $M_{P'}$ satisfies SBNDC.*

Proof. ($\Leftarrow$:) Suppose $p_1\alpha \not\approx p_2\beta$ in M_P. Then there is a winning strategy π for the attacker from $p_1\alpha$ and $p_2\beta$ in M_P begining with $p_1\alpha$. We show that the strategy π serves as the winning strategy for the attacker from sS and $p_1\alpha$ of $M_{P'} \setminus \mathsf{H}$ as well. From the definition of the restricted PDS bisimulation problem,

the attacker chooses a non-ϵ transition from $p_1\alpha$ in π as the first move, say $p_1\alpha \xrightarrow{a} q$ for some $a \in \Sigma$ and $q \in Q \times \Gamma^*$. The defender is forced to choose $sS \xrightarrow{\epsilon} p_2\beta \stackrel{a}{\Rightarrow} q'$ for some $q' \in Q \times \Gamma^*$. For any choice of q' from the defender, π serves as the winning strategy for the attacker from sS and $p_1\alpha$ of $M_{P'} \setminus \mathsf{H}$ as well. Thus sS of $M_{P'} \setminus \mathsf{H}$ and $p_1\alpha$ of $M_{P'} \setminus \mathsf{H}$ are not weakly bisimilar. Hence $M_{P'}$ does not satisfy SBNDC.

($\Rightarrow$:) Suppose $p_1\alpha \approx p_2\beta$. Then there is a winning strategy π for the defender from $p_1\alpha$ and $p_2\beta$ in M_P. We now describe the winning strategy for the defender from sS and $p_1\alpha$ of $M_{P'} \setminus \mathsf{H}$. Consider the different cases for the attacker's choices.

- **Attacker chooses** $sS \xrightarrow{\epsilon} p_2\beta$. The defender stays at $p_1\alpha$ itself. From the next round, the defender plays according to π and wins.
- **Atacker chooses some transition from** $p_1\alpha$. The defender chooses the transition from $p_2\beta$ according to π after $sS \xrightarrow{\epsilon} p_2\beta$.

Thus defender has a winning strategy and hence $sS \approx p_1\alpha$ in $M_{P'} \setminus \mathsf{H}$. Hence $M_{P'}$ satisfies SBNDC. □

Finally from Lemmas 2, 3, 6, 7, 8 and 9 we have:

Theorem 3. *The problem of model-checking pushdown systems for any of the bisimulation-based properties - BNNI, BSNNI, BNDC, SBNNI, SBSNNI and SBNDC is undecidable.* □

5 Model checking Petri nets

We study the problem of model checking each of the bisimulation-based information flow properties in Definition 3 for Petri nets. We begin by defining a Petri net. Let $\mathbb{N}$ denote the set of nonnegative integers.

Definition 6. *A Petri net (PN) is a tuple $N = (P, T, \Sigma, F, L, M_0)$, where P and T are finite disjoint sets of places and transitions respectively, Σ is a finite set of actions, $F : (P \times T) \cup (T \times P) \mapsto \mathbb{N}$ is a flow function, $L : T \mapsto \Sigma \cup \{\epsilon\}$ is a labelling and M_0 is an initial marking (a marking is a function $M : P \mapsto \mathbb{N}$ that gives the number of tokens for each place).*

A PN $N = (P, T, \Sigma, F, L, M_0)$ naturally induces an LTS $M_N = (Q, \Sigma, \rightarrow, M_0)$ where Q is the set of markings and $\rightarrow$ is the set of transitions. A transition t is enabled at a marking M, denoted by $M \xrightarrow{t}$, if $M(p) \geq F(p,t)$, for every $p \in P$. A transition t enabled at a marking M may fire yiedling the marking M', denoted by $M \xrightarrow{t} M'$, where $M'(p) = M(p) - F(p,t) + F(t,p)$, for all $p \in P$. For any $c \in \Sigma \cup \{\epsilon\}$, by $M \xrightarrow{c} M'$ we mean that $M \xrightarrow{t} M'$ for some t with $L(t) = c$.

The problem of model checking a bisimulation-based information flow property θ for PN's is – given a PN N, does M_N satisfy θ? We show that this problem is undecidable for each of the properties in Definition 3.

Jancar [10] shows that the problem of checking strong bisimilarity for PN's is undecidable by a reduction from the halting problem of Minsky machines. Given a Minsky machine R with two counters c_1 and c_2 (cf. Definition 5), he constructs PN's $N_1 = (P_1, T_1, F_1, L_1, M_1)$ and $N_2 = (P_2, T_2, F_2, L_2, M_2)$ such that R halts if and only if $M_1 \not\approx M_2$. For every instruction label L_i, $1 \leq i \leq n$, of R, the places s_i^1 and s_i^2 are created in N_1 and N_2 respectively. The places c_1^1, c_2^1 and c_1^2, c_2^2 are created corresponding to the counters c_1 and c_2 of R in N_1 and N_2 respectively. The PN's N_1 and N_2 simulate the moves of R. At s_n^1 the transition t_F^1 is enabled only when s_n^1 has at least one token. The transition t_F^2 is not enabled even when s_n^2 has tokens. So, when R halts, the attacker simulates R's halting computation in N_1 forcing the defender to simulate R's moves in N_2. The attacker reaches the marking with s_n^1 having at least one token. He wins by making a t_F^1 move for which the defender does not have a matching response. Hence the attacker wins and $M_1 \not\approx M_2$. When R diverges, the defender forces the attacker to simulate R moves either in N_1 or N_2. This induces an infinite game and the defender wins. Thus $M_1 \approx M_2$.

As in the case of Srba's pushdown system construction, here also we observe that if there is a winning strategy for the attacker from (M_1, M_2), there is one beginning with M_1. There are no ϵ-transitions in N_1 and N_2. Hence in general, let M_1 and M_2 be markings in N_1 and N_2 respectively such that M_1 does not have any ϵ-transitions and if there is a winning strategy for the attacker from (M_1, M_2), then there is one beginning with M_1. Then we call the problem of checking whether $M_1 \approx M_2$, the restricted PN bisimulation problem. It follows then from Jancar's construction that:

Theorem 4. *The restricted PN bisimulation problem is undecidable.* □

We reduce the restricted PN bisimulation problem to the problem of checking each of the bisimulation-based information flow properties for PN's. Let the PN's $N_1 = (P_1, T_1, F_1, L_1, M_1)$ and $N_2 = (P_2, T_2, F_2, L_2, M_2)$ be an instance of the restricted PN bisimulation problem. We assume that the sets P_1, P_2 and T_1, T_2 are disjoint. We construct a PN N from N_1 and N_2 such that $N = (P_1 \cup P_2 \cup \{s\}$, $T_1 \cup T_2 \cup \{t_k, t_\epsilon\}$, $\Sigma \cup \{k, \bar{k}\}$, F, L, $M)$ where $k, \bar{k}$ are the only high (and input) events. That is $\mathsf{H} = \mathsf{I} = \{k, \bar{k}\}$. The initial marking M has one token at s and no tokens at all other places i.e., $M(s) = 1$ and $M(p) = 0$, $p \neq s$. The components F and L are described in Fig. 7.

$$F(x,y) = \begin{cases} F_1(x,y) & \text{if both } x \text{ and } y \text{ are in } N_1 \\ F_2(x,y) & \text{if both } x \text{ and } y \text{ are in } N_2 \\ 1 & \text{if } (x,y) = (s, t_k) \\ 1 & \text{if } (x,y) = (s, t_\epsilon) \\ M_1(y) & \text{if } x = t_k \text{ and } y \text{ in } N_1 \\ M_2(y) & \text{if } x = t_\epsilon \text{ and } y \text{ in } N_2 \end{cases} \qquad L(t) = \begin{cases} L_1(t) & \text{if } t \in T_1 \\ L_2(t) & \text{if } t \in T_2 \\ k & \text{if } t = t_k \\ \epsilon & \text{if } t = t_\epsilon \end{cases}$$

Fig. 7. Description of PN N

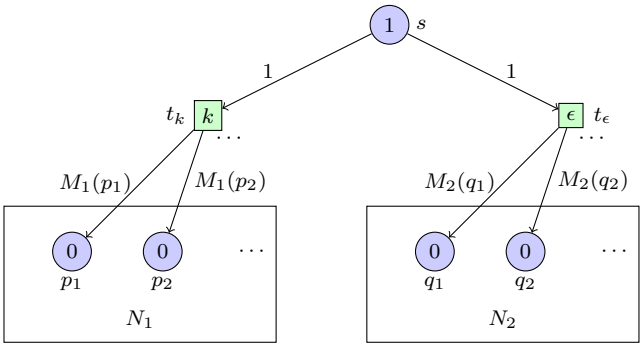

Fig. 8. Constructed Petri net N

The PN N is shown in Fig. 8. Informally, the induced LTS M_N of N has the initial marking M with a high-event k edge – $M \xrightarrow{k} M'$ where $M'(s) = 0$, $M'(p) = M_1(p)$ when $p \in P_1$, $M'(p) = 0$ when $p \in P_2$, and an ϵ-edge – $M \xrightarrow{\epsilon} M''$ where $M''(s) = 0$, $M''(p) = 0$ when $p \in P_1$, $M''(p) = M_2(p)$ when $p \in P_2$. The initial part of the induced LTS M_N is shown in Fig. 9. We fix the PN's N_1, N_2, its markings M_1, M_2 respectively and the PN N constructed from N_1, N_2 as described above for the rest of the section.

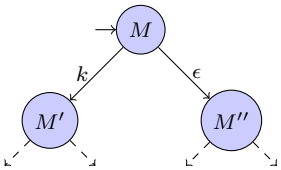

Fig. 9. M_N

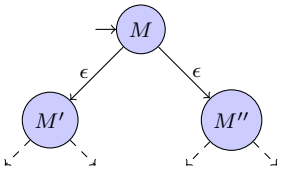

Fig. 10. M_N/H

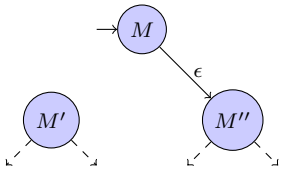

Fig. 11. $M_N \setminus \mathsf{H}$

Lemma 12. *The markings M_1 and M_2 are weakly bisimilar i.e., $M_1 \approx M_2$ iff M_N satisfies BSNNI.*

Proof. From Definition 3, we need to show that $M_1 \approx M_2$ if and only if M of M_N/H (cf. Fig. 10) and M of $M_N \setminus \mathsf{H}$ (cf. Fig. 11) are weakly bisimilar. It is easy to prove this from the arguments similar to the arguments in the proof of Lemma 4. □

Likewise from the similar arguments as in Section 4, we have:

Theorem 5. *The problem of model-checking Petri nets for any of the bisimulation-based properties - BNNI, BSNNI, BNDC, SBNNI, SBSNNI and SBNDC is undecidable.* □

6 Model checking Process algebras

We now study the problem of model checking each of the bisimulation-based information flow properties in Definition 3 for process algebras. We begin by defining a process algebra.

Definition 7. *Let Const be a set of process constants. The class of process expressions over Const is given by* $E ::= \epsilon \mid X \mid E.E \mid E \| E$ *where 'ϵ' is the empty process, X ranges over Const, '.' is the operator of sequential composition, and $\|$ stands for parallel composition.*

A process algebra (PA) N is a tuple (P, Σ, Δ) where P is the initial process expression, Σ is an alphabet and Δ is a finite set of rules of the form $X \xrightarrow{c} E$ where $X \in Const$, $c \in \Sigma \cup \{\epsilon\}$ and E is a process expression.

A PA $N = (P, \Sigma, \Delta)$ determines an LTS $M_N = (Q, \Sigma, \rightarrow, P)$ where the states in Q are process expressions and the transition $\rightarrow$ is the least relation satisfying the following rules. Let $c \in \Sigma \cup \{\epsilon\}$.

$$\frac{(X \xrightarrow{c} E) \in \Delta}{X \xrightarrow{c} E} \qquad \frac{E \xrightarrow{c} E'}{E.F \xrightarrow{c} E'.F} \qquad \frac{E \xrightarrow{c} E'}{E \| F \xrightarrow{c} E' \| F} \qquad \frac{F \xrightarrow{c} F'}{E \| F \xrightarrow{c} E \| F'}$$

The problem of model checking a bisimulation-based information flow property θ for PA's is – given a PA N, does M_N satisfy θ? We show that this problem is undecidable for each of the properties in Definition 3.

Srba [15] has shown that the problem of checking weak bisimilarity for process algebras is undecidable by a reduction from Post's correspondence problem. The Post's correspondence problem (PCP) is defined as – given a nonempty alphabet Σ and two lists $A = [u_1, u_2, \ldots, u_n]$ and $B = [v_1, v_2 \ldots, v_n]$ where $n > 0$ and $u_k, v_k \in \Sigma^+$ for all $k, 1 \leq k \leq n$, the question is to decide whether the (A, B)-instance has a solution, i.e., whether there is an integer $m \geq 1$ and a sequence of indices $i_1, i_2, \ldots, i_m \in \{1, 2, \ldots, n\}$ such that $u_{i_1} u_{i_2} \cdots u_{i_m} = v_{i_1} v_{i_2} \cdots v_{i_m}$. According to the classical result due to Post, this problem is undecidable [12].

Given a (A, B)-instance of PCP, Srba constructs a PA N and two process expressions $X \| C$ and $X' \| C$ such that (A, B)-instance has a solution if and only if $X \| C \approx X' \| C$. As in the case of Srba's pushdown system construction, here also we observe that if there is a winning strategy for the attacker from $(X \| C, X' \| C)$, there is one beginning with $X \| C$. There are no ϵ-transitions at $X \| C$. Hence in general, let E and F be two process expressions of a PA N such that E does not have any ϵ-transitions and if there is a winning strategy for the attacker from (E, F), then there is one beginning with E. Then we call the problem of checking whether $E \approx F$, the restricted PA bisimulation problem. It follows then from Srba's construction that:

Theorem 6. *The restricted PA bisimulation problem is undecidable.* □

We reduce the restricted PA bisimulation problem to the problem of checking each of the bisimulation-based information flow properties for PA's. Let the PA

$N = (P, \Sigma, \Delta)$ and its process expressions E, F be an instance of the restricted PA bisimulation problem. Then we construct N' from N such that $N' = (S, \Sigma \cup \{k, \bar{k}\}, \Delta \cup \{S \xrightarrow{k} E, S \xrightarrow{\epsilon} F\})$ where $S \notin \mathit{Consts}$ of N, $k, \bar{k}$ are the only high (and input) events. That is $\mathsf{H} = \mathsf{I} = \{k, \bar{k}\}$.

From the similar arguments as in Section 4 and using the construction of N' as described above, we have:

Theorem 7. *The problem of model-checking process algebras for any of the bisimulation-based properties - BNNI, BSNNI, BNDC, SBNNI, SBSNNI and SBNDC is undecidable.* □

7 Conclusions

We have shown that model-checking bisimulation-based information flow properties, proposed in the literature, for some well-known classes of infinite state systems is undecidable.

The problem of checking when two deterministic pushdown systems are weakly bisimilar has been shown to be decidable in [16]. This does not imply directly the decidability of checking bisimulation-based properties for deterministic pushdown systems. This is because the *hiding* operation may make the system non-deterministic.

Basic process algebras (BPAs) and Basic parallel processes (BPPs) are subclasses of pushdown systems. The decision problem of checking two BPAs or two BPPs for weak bisimilarity is still open. However it is decidable to check whether two totally normed BPAs or two totally normed BPPs are weakly bisimilar [9]. It will be interesting to explore the model-checking problem for these classes.

Acknowledgements

We thank Jiri Srba, Colin Stirling and Faron Moller for insightful email discussions.

References

1. A. Bossi, R. Focardi, C. Piazza, and S. Rossi. A proof system for information flow security. *Logic Based Program Synthesis and Transformation*, pages 956–956, 2003.
2. Deepak D'Souza, Raveendra Holla, K. R. Raghavendra, and Barbara Sprick. Model-checking trace-based information flow properties. *Journal of Computer Security*, 19(1):101–138, 2011.
3. Deepak D'Souza, Raghavendra K. R., and Barbara Sprick. An automata based approach for verifying information flow properties. *Proceedings of the second workshop on Automated Reasoning for Security Protocol Analysis (ARSPA 2005), Electronic Notes in Theoretical Computer Science*, 135(1):39–58, July 2005.
4. R. Focardi and S. Rossi. Information flow security in dynamic contexts. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW02)*, pages 307–319. Citeseer, 2002.

5. Riccardo Focardi and Roberto Gorrieri. A classification of security properties for process algebras. *Journal of Computer Security, IOS Press*, 3(1):5–33, 1995.
6. Riccardo Focardi and Roberto Gorrieri. The compositional security checker: A tool for the verification of information flow security properties. *Software Engineering*, 23(9):550–571, 1997.
7. Joseph A. Goguen and José Meseguer. Security policies and security models. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 11–20, April 1982.
8. Will Harwood, Faron Moller, and Anton Setzer. Weak bisimulation approximants. In Zoltn sik, editor, *Computer Science Logic*, volume 4207 of *Lecture Notes in Computer Science*, pages 365–379. Springer Berlin / Heidelberg, 2006.
9. Y. Hirshfeld. Bisimulation trees and the decidability of weak bisimulations. *Electronic Notes in Theoretical Computer Science*, 5:2–13, 1997.
10. Petr Jancar. Decidability questions for bismilarity of petri nets and some related problems. In *Proceedings of the 11th Annual Symposium on Theoretical Aspects of Computer Science*, STACS '94, pages 581–592, London, UK, UK, 1994. Springer-Verlag.
11. Marvin L. Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.
12. E. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52(4):264–268, 1946.
13. J. Srba. Roadmap of infinite results. *Current Trends In Theoretical Computer Science, The Challenge of the New Century*, 2:337–350, 2004.
14. Jirí Srba. Undecidability of weak bisimilarity for pushdown processes. In *Proceedings of the 13th International Conference on Concurrency Theory*, CONCUR '02, pages 579–593, London, UK, 2002. Springer-Verlag.
15. Jiří Srba. Undecidability of weak bisimilarity for pa-processes. In *Proceedings of the 6th international conference on Developments in language theory*, DLT'02, pages 197–209, Berlin, Heidelberg, 2003. Springer-Verlag.
16. Colin Stirling. Decidability of bisimulation equivalence for pushdown processes. Technical report, 2000.