Branching Bisimulation Semantics Enables Noninterference Analysis of Reversible Systems

Andrea Esposito

University of Urbino

Joint work with Alessandro Aldini and Marco Bernardo

Noninterference and Reversibility

- The notion of noninterference was first introduced by Goguen and Meseguer (1982).
- It allows us to reason about the way in which illegitimate information flows can occur in multi-level security systems by exploiting so-called covert channels.
- Noninterference guarantees that low-level agents can never infer from their observations what high-level agents are doing.
- Security property verification is carried out with different approaches: type theory, abstract interpretation, model checking, etc.
- Regardless of the specific implementation, noninterference is closely tied to the notion of behavioral equivalence among processes.

Noninterference and Reversibility

- In the process algebraic framework one of the most established formal definitions of equivalence employed to define noninterference properties is weak bisimilarity.
- We claim that it is worth studying nondeterministic noninterference in a different setting, relying on branching bisimulation semantics.
- This approach is justified by the fact that branching bisimilarity can be used to analyze reversible systems, since this equivalence has been proved to coincide with back-and-forth bisimilarity.

Labeled Transition Systems

- To represent the behavior of a process we use a labeled transition system, which is a state-transition graph whose transitions are labeled with actions.

Definition

A *labeled transition system (LTS)* is a triple $(S, A, \longrightarrow)$ where:

- $S \neq \emptyset$ is an at most countable set of states
- $A \neq \emptyset$ is a countable set of actions
  with $\tau \in A$ denoting the invisible or silent action
- $\longrightarrow \subseteq S \times A \times S$ is a transition relation.

Weak Bisimilarity

- Weak bisimilarity was introduced by Milner (1989) to abstract from the invisible (or internal) action τ.

Definition

$s_1 \approx s_2$ iff $(s_1, s_2) \in \mathcal{B}$ for some weakbisimulation $\mathcal{B}$.
A symmetric binary relation $\mathcal{B}$ over S is a *weak bisimulation* iff, whenever $(s_1, s_2) \in \mathcal{B}$, then:

- whenever $s_1 \xrightarrow{\tau} s_1'$, then $s_2 \stackrel{\tau^*}{\Longrightarrow} s_2'$ with $(s_1', s_2') \in \mathcal{B}$;
- whenever $s_1 \xrightarrow{a} s_1'$ for $a \in A \setminus \{\tau\}$, then $s_2 \stackrel{\tau^*}{\Longrightarrow} \xrightarrow{a} \stackrel{\tau^*}{\Longrightarrow} s_2'$ with $(s_1', s_2') \in \mathcal{B}$.

Branching Bisimilarity

- Introduced by Van Glabbeek and Wejland (1996) as a refinement of weak bisimulation that preserves the branching structure of processes even when abstracting from the invisible action τ.

Definition

$s_1 \approx_b s_2$ iff $(s_1, s_2) \in \mathcal{B}$ for some branching bisimulation $\mathcal{B}$.
A symmetric binary relation $\mathcal{B}$ over S is a *branching bisimulation* iff, whenever $(s_1, s_2) \in \mathcal{B}$, then for all actions $a \in A$:

- whenever $s_1 \xrightarrow{a} s_1'$, then:
  - either $a = \tau$ and $(s_1', s_2) \in \mathcal{B}$;
  - or $s_2 \stackrel{\tau^*}{\Longrightarrow} \bar{s}_2 \xrightarrow{a} s_2'$ with $(s_1, \bar{s}_2) \in \mathcal{B}$ and $(s_1', s_2') \in \mathcal{B}$.

Process Language

- The set of process terms $\mathbb{P}$ is generated by the following syntax:

$$P ::= \underline{0} \mid a\,.\,P \mid P + P \mid P \parallel_S P \mid P \setminus L \mid P / L$$

  with $a \in \mathcal{A}_\tau$ and $L \subseteq \mathcal{A}$.
- For an adequate action representation of multi-level security systems, two disjoint sets for actions are needed:
  - one for actions performed by low level agents ($\mathcal{A}_\mathcal{L}$);
  - one for actions performed by high level agents ($\mathcal{A}_\mathcal{H}$).
- The set of visible actions will be denoted by $\mathcal{A} := \mathcal{A}_\mathcal{L} \cup \mathcal{A}_\mathcal{H}$.
- The overall set of actions will be denoted by $\mathcal{A}_\tau := \mathcal{A} \cup \{\tau\}$.

The Language

- $\underline{0}$ is the terminated process.
- $a\,.\,P$, for $a \in \mathcal{A}_\tau$, is the action prefix operator describing a process that initially performs action a.
- $P_1 + P_2$ is the alternative composition operator expressing a nondeterministic choice between two processes based on their executable actions.
- $P_1 \parallel_L P_2$, for $L \subseteq \mathcal{A}$, is the parallel composition operator that forces two processes to synchronize on any action in L.
- $P \setminus L$, for $L \subseteq \mathcal{A}$, is the restriction operator, which prevents the execution of actions in L.
- $P \,/\, L$, for $L \subseteq \mathcal{A}$, is the hiding operator, which turns all the executed actions in L into the invisible action τ.

Operational Semantic Rules

- Operational semantic rule for action prefix:

$$\boxed{a\,.\,P \xrightarrow{a} P}$$

- Operational semantic rules for the choice operator:

$$\boxed{\frac{P_1 \xrightarrow{a} P_1'}{P_1 + P_2 \xrightarrow{a} P_1'} \qquad \frac{P_2 \xrightarrow{a} P_2'}{P_1 + P_2 \xrightarrow{a} P_2'}}$$

Operational Semantic Rules

- Operational semantic rules for parallel composition:

$$\boxed{\frac{P_1 \xrightarrow{a} P_1' \quad a \notin L}{P_1 \parallel_L P_2 \xrightarrow{a} P_1' \parallel_L P_2} \qquad \frac{P_2 \xrightarrow{a} P_2' \quad a \notin L}{P_1 \parallel_L P_2 \xrightarrow{a} P_1 \parallel_L P_2'}}$$

- Operational semantic rule for synchronization:

$$\boxed{\frac{P_1 \xrightarrow{a} P_1' \quad P_2 \xrightarrow{a} P_2' \quad a \in L}{P_1 \parallel_L P_2 \xrightarrow{a} P_1' \parallel_L P_2'}}$$

Operational Semantic Rules

- Operational semantic rules for restriction and hiding:

$$\frac{P \xrightarrow{a} P' \quad a \notin L}{P \setminus L \xrightarrow{a} P' \setminus L}$$

$$\frac{P \xrightarrow{a} P' \quad a \in L}{P/L \xrightarrow{\tau} P'/L} \qquad \frac{P \xrightarrow{a} P' \quad a \notin L}{P/L \xrightarrow{a} P'/L}$$

Noninterference

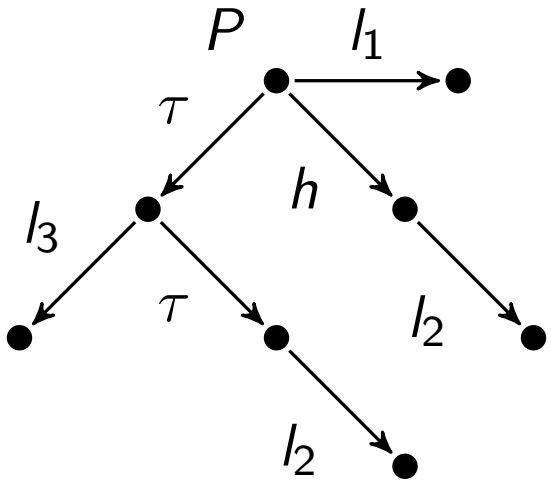
- The intuition behind noninterference in a two-level security system is that, whenever a group of agents at the high security level performs some actions, the effect of those actions should not be seen by any agent at the low security level.
- We examine a selection of weak-bisimilarity-based noninterference properties.
- Focardi and Gorrieri (2001) provided a characterization of some of these properties in a process algebraic framework, resulting in a study of properties and comparisons between these different properties.

Noninterference

- The first property we examine is the *Bisimulation-based Strong Nondeterministic Non Interference* (BSNNI).
- It is satisfied by any process that behaves the same when its high-level actions are removed or are hidden.

Definition

Let $P \in \mathbb{P}$. $P \in \mathrm{BSNNI} \iff P \setminus \mathcal{A}_{\mathcal{H}} \approx P / \mathcal{A}_{\mathcal{H}}$.

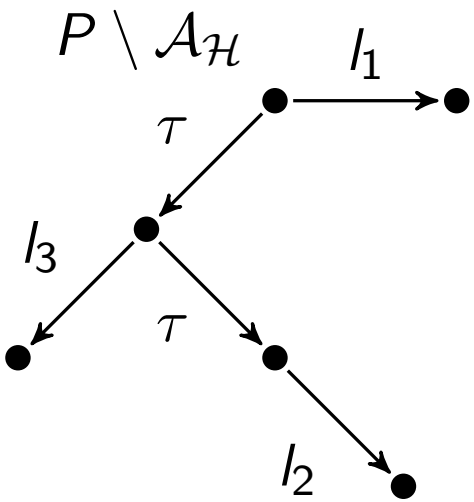


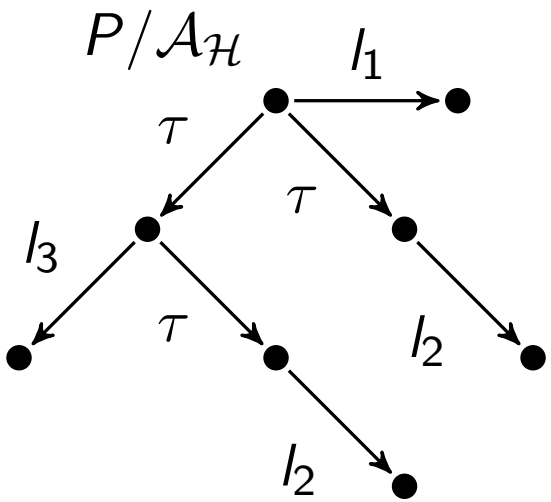
Noninterference

- The first property we examine is the *Bisimulation-based Strong Nondeterministic Non Interference* (BSNNI).
- It is satisfied by any process that behaves the same when its high-level actions are removed or are hidden.

Definition

Let $P \in \mathbb{P}$. $P \in \mathrm{BSNNI} \iff P \setminus \mathcal{A}_{\mathcal{H}} \approx P / \mathcal{A}_{\mathcal{H}}$.





Noninterference

- BSNNI is not powerful enough to capture covert channels that derive from the behavior of the high-level agent interacting with the system, so other stronger properties have been studied in the literature.
- *Non Deducibility on Composition* (BNDC) requires to check explicitly the interaction between the system and every possible high-level agent.
- *Strong* BSNNI (SBSNNI) requires that at any reachable state the property BSNNI must be satisfied.
- *Strong Non Deducibility on Composition* (SBNDC) requires that the low-level view of every reachable state of a system must be the same before and after the execution of every high level action.

Noninterference

Definition

Let $P \in \mathbb{P}$:

- $P \in \mathrm{BSNNI} \iff P \setminus \mathcal{A}_{\mathcal{H}} \approx P \,/\, \mathcal{A}_{\mathcal{H}}$.
- $P \in \mathrm{BNDC} \iff$ for all $Q \in \mathbb{P}$ such that every $Q' \in \mathit{reach}(Q)$ can execute only actions in $\mathcal{A}_{\mathcal{H}}$ and for all $L \subseteq \mathcal{A}_{\mathcal{H}}$, $P \setminus \mathcal{A}_{\mathcal{H}} \approx ((P \parallel_L Q) \,/\, L) \setminus \mathcal{A}_{\mathcal{H}}$.
- $P \in \mathrm{SBSNNI} \iff P \in \mathrm{BSNNI}$ and for all $P' \in \mathit{reach}(P)$, $P' \in \mathrm{BSNNI}$.
- $P \in \mathrm{SBNDC} \iff$ for all $P' \in \mathit{reach}(P)$ and for all P'' such that $P' \xrightarrow{a} P''$ for some $a \in \mathcal{A}_{\mathcal{H}}$, $P' \setminus \mathcal{A}_{\mathcal{H}} \approx P'' \setminus \mathcal{A}_{\mathcal{H}}$.

Relation among Properties

- Focardi and Gorrieri showed that the following taxonomy of information-flow security properties holds:

$$\text{SBNDC} \longrightarrow \text{SBSNNI} \longrightarrow \text{BNDC} \longrightarrow \text{BSNNI}$$

Branching-Bisimulation-Based Properties

- We recast information-flow security definitions in terms of branching bisimilarity and investigate their characteristics as well as their relationships with the definitions based on weak bisimilarity.

Definition

Let $P \in \mathbb{P}$:

- $P \in \mathrm{BrSNNI} \iff P \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P \,/\, \mathcal{A}_{\mathcal{H}}$.
- $P \in \mathrm{BrNDC} \iff$ for all $Q \in \mathbb{P}$ such that every $Q' \in \mathit{reach}(Q)$ can execute only actions in $\mathcal{A}_{\mathcal{H}}$ and for all $L \subseteq \mathcal{A}_{\mathcal{H}}$, $P \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} ((P \parallel_L Q) \,/\, L) \setminus \mathcal{A}_{\mathcal{H}}$.
- $P \in \mathrm{SBrSNNI} \iff P \in \mathrm{BrSNNI}$ and for all $P' \in \mathit{reach}(P)$, $P' \in \mathrm{BrSNNI}$.
- $P \in \mathrm{SBrNDC} \iff$ for all $P' \in \mathit{reach}(P)$ and for all P'' such that $P' \xrightarrow{a} P''$ for some $a \in \mathcal{A}_{\mathcal{H}}$, $P' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P'' \setminus \mathcal{A}_{\mathcal{H}}$.

Preservation

Theorem

Let $P_1, P_2 \in \mathbb{P}$ *and* $\mathcal{P} \in \{\mathrm{BrSNNI}, \mathrm{BrNDC}, \mathrm{SBrSNNI}, \mathrm{SBrNDC}\}$.
If $P_1 \approx_{\mathrm{b}} P_2$, *then* $P_1 \in \mathcal{P} \iff P_2 \in \mathcal{P}$.

- This is very useful in automated property verification as it can be more convenient to work with a reduced system, i.e., a system equivalent to the one we are checking but with a smaller state space.

Compositionality

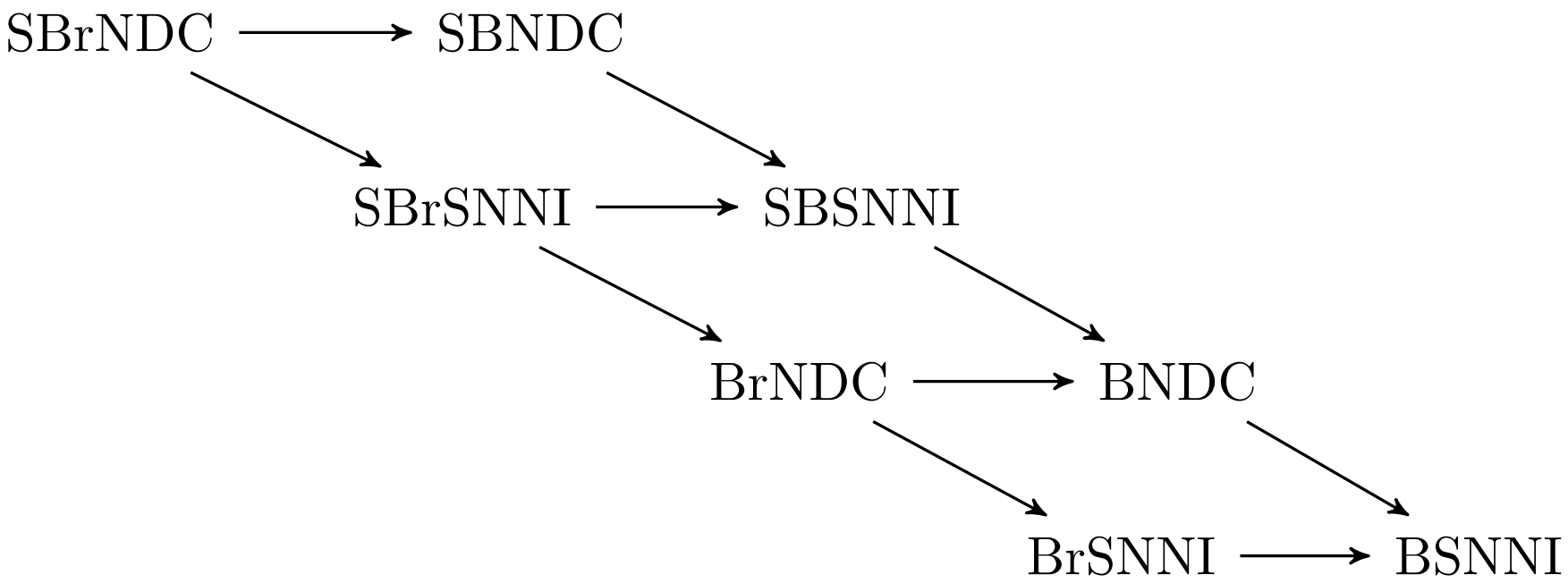
Theorem

Let $P, P_1, P_2 \in \mathbb{P}$ *and* $\mathcal{P} \in \{\mathrm{SBrSNNI}, \mathrm{SBrNDC}\}$. *Then:*

1. $P \in \mathcal{P} \Longrightarrow a\,.\,P \in \mathcal{P}$ *for all* $a \in \mathcal{A}_\tau \setminus \mathcal{A}_\mathcal{H}$.
2. $P_1, P_2 \in \mathcal{P} \Longrightarrow P_1 \parallel_L P_2 \in \mathcal{P}$ *for all* $L \subseteq \mathcal{A}$.
3. $P \in \mathcal{P} \Longrightarrow P \setminus L \in \mathcal{P}$ *for all* $L \subseteq \mathcal{A}_\mathcal{L}$ *if* $\mathcal{P} = \mathrm{SBrSNNI}$, $L \subseteq \mathcal{A}$ *if* $\mathcal{P} = \mathrm{SBrNDC}$.
4. $P \in \mathcal{P} \Longrightarrow P / L \in \mathcal{P}$ *for all* $L \subseteq \mathcal{A}_\mathcal{L}$.

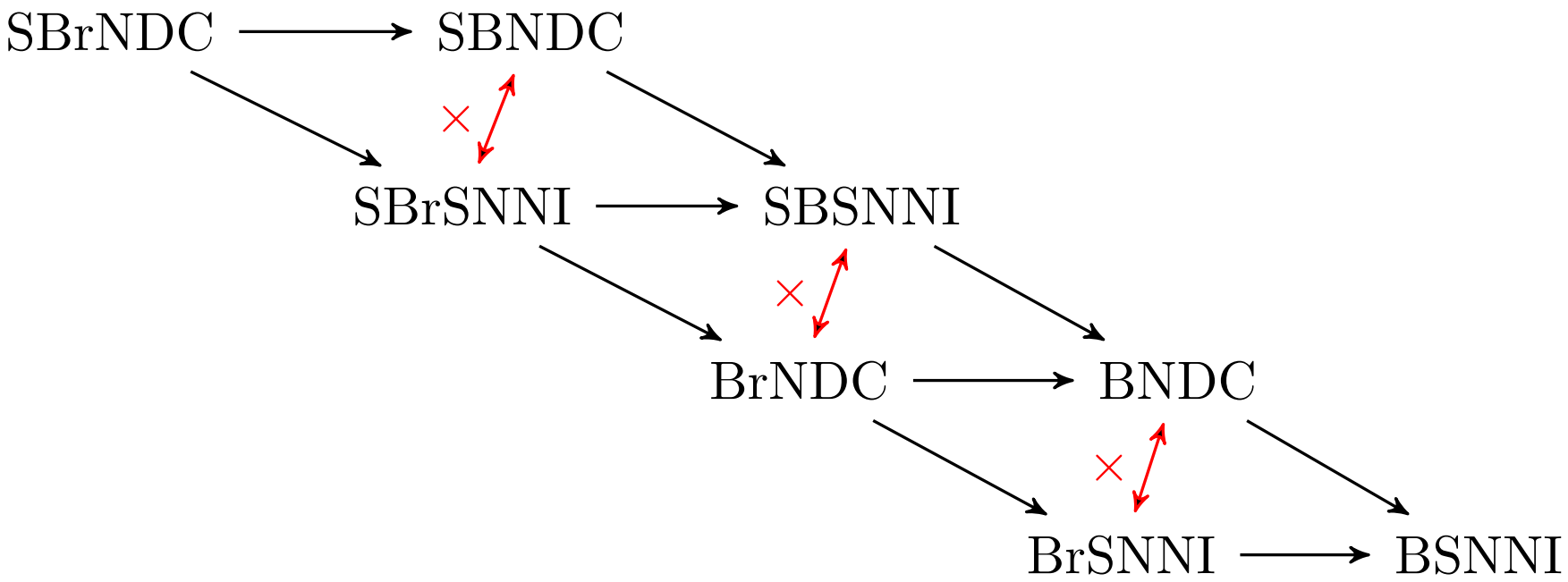
Extended Taxonomy

- Taxonomy of security properties based on weak and branching bisimulation.



Extended Taxonomy

- Taxonomy of security properties based on weak and branching bisimulation.



Axiom-Based Methodology

- τ-axioms for weak bisimulation:

$$\begin{aligned} \tau\,.\,x + x &= \tau\,.\,x \\ a.(\tau\,.\,x + y) + a\,.\,x &= a\,.\,(\tau\,.\,x + y) \end{aligned}$$

- The strategy consists of constructing new processes from the ones in the last two τ-axioms such that the former are weakly bisimilar by construction, but not branching bisimilar.
- Then from such a pair of processes we define a new process P such that $P \setminus \mathcal{A}_{\mathcal{H}}$ and $P/\mathcal{A}_{\mathcal{H}}$ are isomorphic to the processes constructed from the terms of those τ-axioms.
- The process P will be BSNNI but not $\mathrm{SBrSNNI}$.

Axiom-Based Methodology

Theorem

From $\tau \,.\, x + x = \tau \,.\, x$ *it is possible to construct* $P \in \mathbb{P}$ *such that* $P \in \mathrm{BSNNI} \cap \mathrm{SBSNNI}$, *but* $P \notin \mathrm{BrSNNI} \cup \mathrm{SBrSNNI}$.

Theorem

From $a \,.\, (\tau \,.\, x + y) + a \,.\, x = a \,.\, (\tau \,.\, x + y)$ *it is possible to construct* $P \in \mathbb{P}$ *such that* $P \in \mathrm{BSNNI} \cap \mathrm{SBSNNI}$, *but* $P \notin \mathrm{BrSNNI} \cup \mathrm{SBrSNNI}$.

Back-and-Forth Bisimilarity

- Introduced by De Nicola, Montanari and Vandraager (1990).
- Back-and-forth bisimulations are defined over *computational paths* instead of states.
- This is needed in order to remain in an interleaving setting of concurrency, preserve causal consistency and the computational history.
- This means that whenever a process returns to a past state it must do it by reverting the same computational path performed in going forward.

Weak Back-and-Forth Bisimilarity

Definition

$s_1 \approx_{\mathrm{bf}} s_2$ iff $((s_1, \varepsilon), (s_2, \varepsilon)) \in \mathcal{B}$ for some weak back-and-forth bisimulation $\mathcal{B}$.

A symmetric binary relation $\mathcal{B}$ over R is a *weak back-and-forth bisimulation* iff, whenever $(\rho_1, \rho_2) \in \mathcal{B}$, then:

- whenever $\rho_1 \xrightarrow{\tau} \rho_1'$, then $\rho_2 \stackrel{\tau^*}{\Longrightarrow} \rho_2'$ with $(\rho_1', \rho_2') \in \mathcal{B}$;
- whenever $\rho_1' \xrightarrow{\tau} \rho_1$, then $\rho_2' \stackrel{\tau^*}{\Longrightarrow} \rho_2$ with $(\rho_1', \rho_2') \in \mathcal{B}$;
- whenever $\rho_1 \xrightarrow{a} \rho_1'$ for $a \in A \setminus \{\tau\}$, then $\rho_2 \stackrel{\tau^*}{\Longrightarrow} \xrightarrow{a} \stackrel{\tau^*}{\Longrightarrow} \rho_2'$ with $(\rho_1', \rho_2') \in \mathcal{B}$;
- whenever $\rho_1' \xrightarrow{a} \rho_1$ for $a \in A \setminus \{\tau\}$, then $\rho_2' \stackrel{\tau^*}{\Longrightarrow} \xrightarrow{a} \stackrel{\tau^*}{\Longrightarrow} \rho_2$ with $(\rho_1', \rho_2') \in \mathcal{B}$.

Comparisons

- Strong back-and-forth bisimilarity coincides with strong bisimilarity.
- Weak back-and-forth bisimilarity, however, is finer than weak bisimilarity.
- Surprisingly weak back-and-forth bisimilarity coincides with branching bisimilarity.

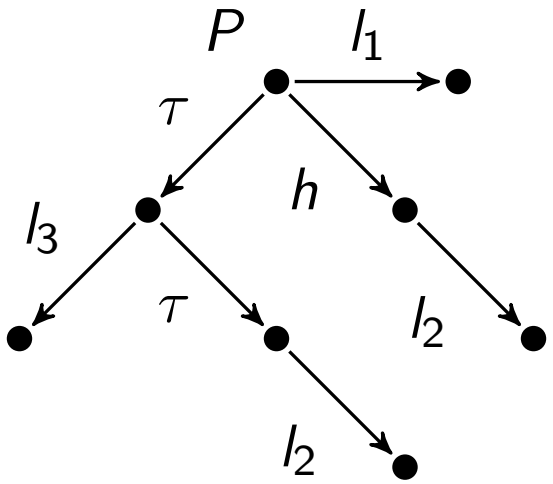
Theorem

$$s_1 \approx_{\mathrm{bf}} s_2 \text{ iff } s_1 \approx_{\mathrm{b}} s_2.$$

- This allows us to reason about reversible systems without resorting to a reversible calculus nor a path-based equivalence.
- All the results for branching-bisimulation-based properties can be extended to reversible systems.

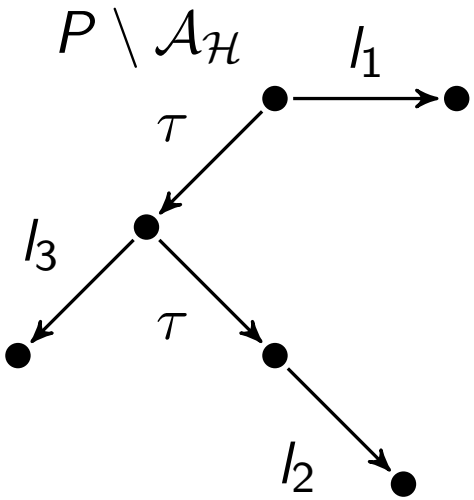
Example

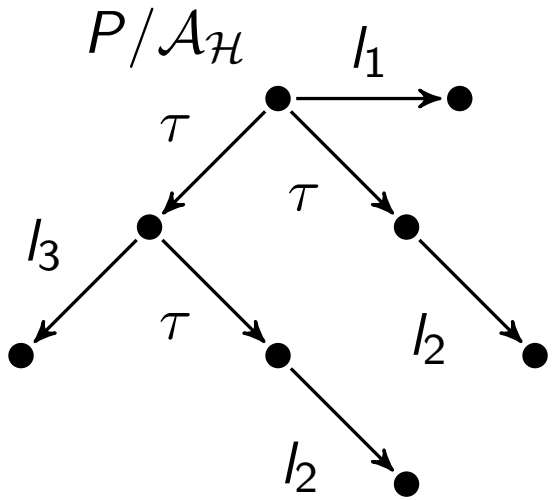
- Let us look again at the BSNNI-secure process P.



Example

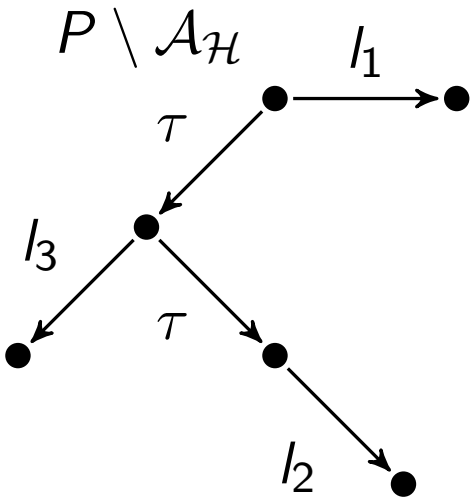
- If we take P as a reversible system we can see that it is not secure.

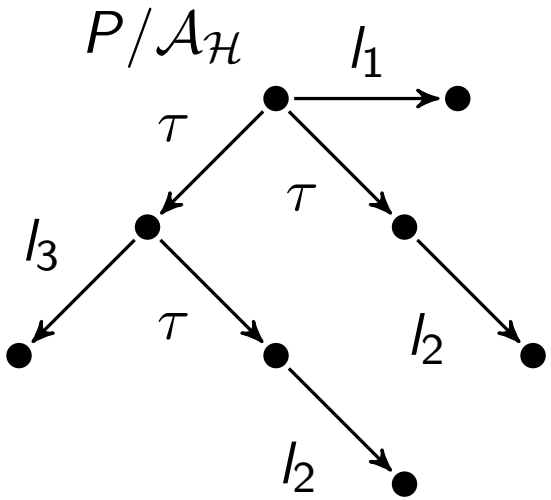




Example

- The information-flow can be also detected by employing BrSNNI.





Conclusions

- Potential covert channels arising in reversible context cannot be revealed by employing a standard weak bisimulation semantics.
- Indeed, the higher discriminating power of branching bisimilarity is necessary to capture information flows emerging whenever backward computations are activated.
- We have rephrased in the setting of branching bisimilarity the classical taxonomy of nondeterministic noninterference properties based on weak bisimilarity.
- We have introduced a methodology based on the τ-axioms of weak bisimilarity to prove the strictness of certain inclusion in the extended taxonomy.

Future Works

- We would like to study non interference and reversibility in contexts where quantitative aspects play an important role in processes, like probabilistic systems.
- Some of our proofs rely on the representation of processes as tree, we would like to improve our proofs as to include the recursion operator.