

Branching Bisimulation Semantics Enables Noninterference Analysis of Reversible Systems

Andrea Esposito

University of Urbino

Joint work with Alessandro Aldini and Marco Bernardo

- The notion of **noninterference** was first introduced by Goguen and Meseguer (1982).
- Used to reason about the way in which illegitimate information flows can occur in **multi-level security systems** by exploiting covert channels.
- Noninterference guarantees that low-level agents can never infer from their observations what high-level agents are doing.
- Regardless of the specific implementation, noninterference is closely tied to the notion of **behavioral equivalence** among processes.

Noninterference in Reversible Systems

- In the **process algebraic** framework one of the most established formal definitions of equivalence employed for noninterference properties is **weak bisimilarity**.

Noninterference in Reversible Systems

- In the **process algebraic** framework one of the most established formal definitions of equivalence employed for noninterference properties is **weak bisimilarity**.
- **Is it adequate to study noninterference in reversible systems?**

Noninterference in Reversible Systems

- In the **process algebraic** framework one of the most established formal definitions of equivalence employed for noninterference properties is **weak bisimilarity**.
- **Is it adequate to study noninterference in reversible systems?**
- Computations can go not only forward but also backward.

Noninterference in Reversible Systems

- In the **process algebraic** framework one of the most established formal definitions of equivalence employed for noninterference properties is **weak bisimilarity**.
- **Is it adequate to study noninterference in reversible systems?**
- Computations can go not only forward but also backward.
- **Branching bisimilarity** has been proved to coincide with **weak back-and-forth bisimilarity**.
- We thus study noninterference of reversible systems based on **branching bisimilarity**.

Labeled Transition Systems

- To represent process behavior we use a **labeled transition system**, 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** 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 **weak bisimulation** \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 $s_2 \xRightarrow{\tau^*} s'_2$ with $(s'_1, s'_2) \in \mathcal{B}$;
- whenever $s_1 \xrightarrow{a} s'_1$ for $a \in A \setminus \{\tau\}$, then $s_2 \xRightarrow{\tau^*} \xrightarrow{a} \xRightarrow{\tau^*} s'_2$ with $(s'_1, s'_2) \in \mathcal{B}$.

Branching Bisimilarity

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

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 $s_2 \xrightarrow{\tau^*} \bar{s}_2 \xrightarrow{a} s'_2$ 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 the following, where $a \in \mathcal{A}_\tau$ and $L \subseteq \mathcal{A}$:

\mathbb{P}	::=	$\underline{0}$	Terminated process.
		$a.P$	Action prefix.
		$P_1 + P_2$	Alternative choice.
		$P_1 \parallel_L P_2$	Paralell composition.
		$P \setminus L$	Restriction.
		P / L	Hiding.

Process Language

- The set of process terms \mathbb{P} is the following, where $a \in \mathcal{A}_\tau$ and $L \subseteq \mathcal{A}$:

\mathbb{P}	$::=$	$\underline{0}$	Terminated process.
		$a.P$	Action prefix.
		$P_1 + P_2$	Alternative choice.
		$P_1 \parallel_L P_2$	Parallel composition.
		$P \setminus L$	Restriction.
		P / L	Hiding.

- Two sets of actions for multi-level security systems:
 - Low level actions: \mathcal{A}_L .
 - High level actions: \mathcal{A}_H .
- Set of visible actions: $\mathcal{A} := \mathcal{A}_L \cup \mathcal{A}_H$.
- Overall set of actions: $\mathcal{A}_\tau := \mathcal{A} \cup \{\tau\}$.
- Restriction and hiding are needed to formalize noninterference.

Operational Semantic Rules

- Operational semantic rule for **action prefix**:

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

- Operational semantic rules for **choice**:

$$\frac{P_1 \xrightarrow{a} P'_1}{P_1 + P_2 \xrightarrow{a} P'_1} \quad \frac{P_2 \xrightarrow{a} P'_2}{P_1 + P_2 \xrightarrow{a} P'_2}$$

Operational Semantic Rules

- Operational semantic rules for **parallel composition**:

$$\boxed{\frac{P_1 \xrightarrow{a} P'_1 \quad a \notin L}{P_1 \parallel_L P_2 \xrightarrow{a} P'_1 \parallel_L P_2} \quad \frac{P_2 \xrightarrow{a} P'_2 \quad a \notin L}{P_1 \parallel_L P_2 \xrightarrow{a} P_1 \parallel_L P'_2}}$$

- 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 for **restriction and hiding**:

$$\boxed{\begin{array}{c} \frac{P \xrightarrow{a} P' \quad a \notin L}{P \setminus L \xrightarrow{a} P' \setminus L} \\ \\ \frac{P \xrightarrow{a} P' \quad a \in L}{P/L \xrightarrow{\tau} P'/L} \quad \frac{P \xrightarrow{a} P' \quad a \notin L}{P/L \xrightarrow{a} P'/L} \end{array}}$$

- 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 recall some **weak-bisimilarity**-based noninterference properties.
- Focardi and Gorrieri (2001) provided a characterization of these properties in a **process algebraic** framework, resulting in a study of their features and comparisons between them.

Weak-Bisimulation-Based Properties

- 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 forbidden or are hidden.

Definition

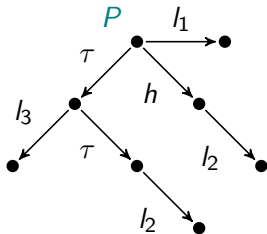
Let $P \in \mathbb{P}$. $P \in \text{BSNNI} \iff P \setminus \mathcal{A}_H \approx P / \mathcal{A}_H$.

Weak-Bisimulation-Based Properties

- 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 forbidden or are hidden.

Definition

Let $P \in \mathbb{P}$. $P \in \text{BSNNI} \iff P \setminus \mathcal{A}_H \approx P / \mathcal{A}_H$.

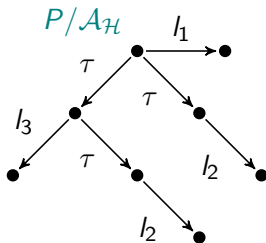
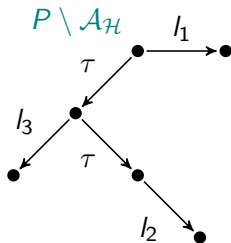


Weak-Bisimulation-Based Properties

- 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 forbidden or are hidden.

Definition

Let $P \in \mathbb{P}$. $P \in \text{BSNNI} \iff P \setminus \mathcal{A}_H \approx P / \mathcal{A}_H$.



Weak-Bisimulation-Based Properties

- **BSNNI** is not powerful enough to capture covert channels that derive from the behavior of high-level agents 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 BNDC* (**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.

Definition

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

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

- Focardi and Gorrieri showed the following taxonomy:

SBNDC \longrightarrow SBSNNI \longrightarrow BNDC \longrightarrow 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 $P \in \mathbb{P}$:

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

- \approx_b preserves all the four properties.

Theorem

Let $P_1, P_2 \in \mathbb{P}$ and $\mathcal{P} \in \{\text{BrSNNI}, \text{BrNDC}, \text{SBrSNNI}, \text{SBrNDC}\}$.

If $P_1 \approx_b P_2$, 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.

- The stronger properties are preserved by the operators of \mathbb{P} .

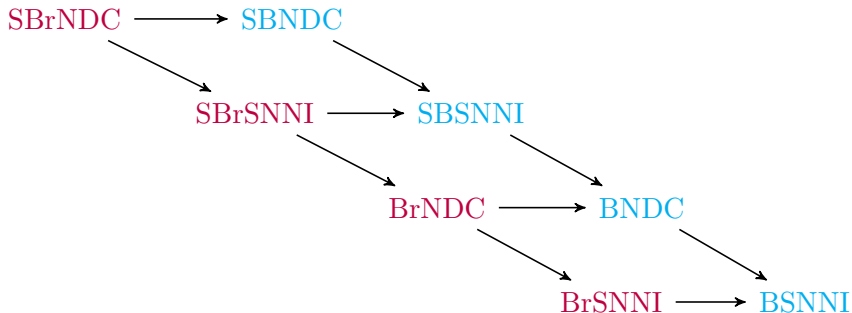
Theorem

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

- 1 $P \in \mathcal{P} \implies a.P \in \mathcal{P}$ for all $a \in \mathcal{A}_\tau \setminus \mathcal{A}_H$.
- 2 $P_1, P_2 \in \mathcal{P} \implies P_1 \parallel_L P_2 \in \mathcal{P}$ for all $L \subseteq \mathcal{A}$.
- 3 $P \in \mathcal{P} \implies P \setminus L \in \mathcal{P}$
for all $L \subseteq \mathcal{A}_L$ if $\mathcal{P} = \text{SBrSNNI}$, $L \subseteq \mathcal{A}$ if $\mathcal{P} = \text{SBrNDC}$.
- 4 $P \in \mathcal{P} \implies P / L \in \mathcal{P}$ for all $L \subseteq \mathcal{A}_L$.

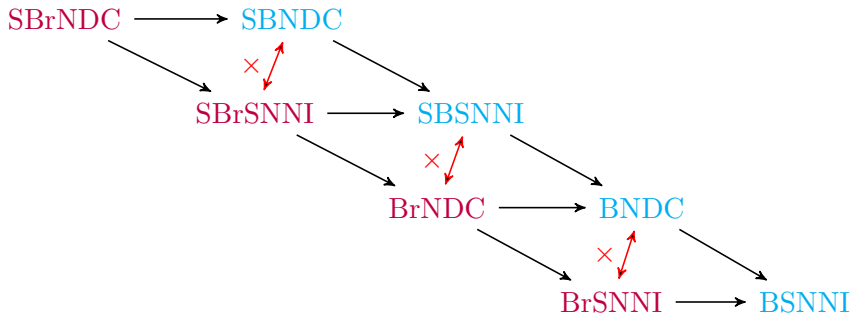
Extended Taxonomy

- Taxonomy of security properties based on **weak** and **branching** bisimilarities:



Extended Taxonomy

- Taxonomy of security properties based on **weak** and **branching** bisimilarities:



Back-and-Forth Bisimilarities

- Introduced by De Nicola, Montanari and Vandraager (1990).
- **Back-and-forth bisimulations** are defined over *computational paths* instead of states.
- This is needed to remain in an **interleaving setting** of concurrency.
- It preserves not only **causal consistency** but also **history**.
- 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

$s_1 \approx_{\text{bf}} s_2$ 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 $\rho_2 \xRightarrow{\tau^*} \rho'_2$ with $(\rho'_1, \rho'_2) \in \mathcal{B}$;
- whenever $\rho'_1 \xrightarrow{\tau} \rho_1$, then $\rho'_2 \xRightarrow{\tau^*} \rho_2$ with $(\rho'_1, \rho'_2) \in \mathcal{B}$;
- whenever $\rho_1 \xrightarrow{a} \rho'_1$ for $a \in A \setminus \{\tau\}$, then $\rho_2 \xRightarrow{\tau^*} \xrightarrow{a} \xRightarrow{\tau^*} \rho'_2$ with $(\rho'_1, \rho'_2) \in \mathcal{B}$;
- whenever $\rho'_1 \xrightarrow{a} \rho_1$ for $a \in A \setminus \{\tau\}$, then $\rho'_2 \xRightarrow{\tau^*} \xrightarrow{a} \xRightarrow{\tau^*} \rho_2$ with $(\rho'_1, \rho'_2) \in \mathcal{B}$.

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

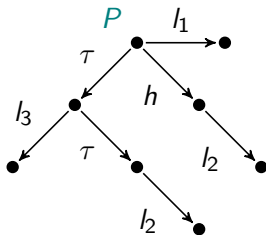
Theorem

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

- We can 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.

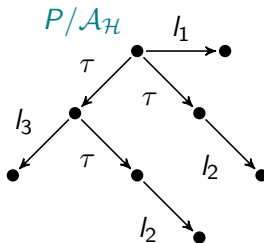
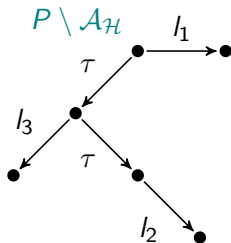
Example

- Let us look again at the **BSNNI**-secure process P :



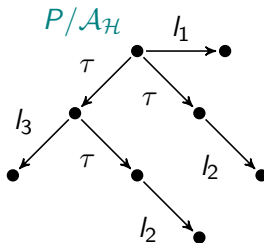
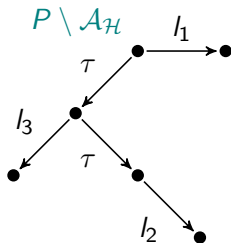
Example

- 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 systems 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 would like to study **noninterference** 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 because branching bisimulation up to \approx_b is too restrictive. How to recover recursion in our language?