NONINTERFERENCE ANALYSIS OF REVERSIBLE SYSTEMS: AN APPROACH BASED ON BRANCHING BISIMILARITY

ANDREA ESPOSITO [a], ALESSANDRO ALDINI [a], MARCO BERNARDO [a], AND SABINA ROSSI [b]

[a] Dipartimento di Scienze Pure e Applicate, Università di Urbino, Italy
e-mail address: andrea.esposito@uniurb.it, alessandro.aldini@uniurb.it, marco.bernardo@uniurb.it

[b] Dipartimento di Scienze Ambientali, Informatica e Statistica, Università Ca' Foscari, Venezia, Italy
e-mail address: sabina.rossi@unive.it

Abstract. The theory of noninterference supports the analysis of information leakage and the execution of secure computations in multi-level security systems. Classical equivalence-based approaches to noninterference mainly rely on weak bisimulation semantics. We show that this approach is not sufficient to identify potential covert channels in the presence of reversible computations. As illustrated via a database management system example, the activation of backward computations may trigger information flows that are not observable when proceeding in the standard forward direction. To capture the effects of back-and-forth computations, it is necessary to switch to a more expressive semantics, which has been proven to be branching bisimilarity in a previous work by De Nicola, Montanari, and Vaandrager. In this paper we investigate a taxonomy of noninterference properties based on branching bisimilarity along with their preservation and compositionality features, then we compare it with the taxonomy of Focardi and Gorrieri based on weak bisimilarity.

1. Introduction

Noninterference was introduced by Goguen and Meseguer [GM82] to reason about the way in which illegitimate information flows can occur due to covert channels from high-level agents to low-level ones in multi-level security systems. Since the first definition, conceived for deterministic state machines, in the last four decades a lot of work has been done that led to a variety of extensions (dealing with nondeterminism or quantitative domains) in multiple frameworks (from language-based security to concurrency theory); see, e.g., [FG01, Ald06, Man11, HS12, ABG04, HMPR21] and the references therein. Analogously, the techniques proposed to verify information-flow security properties based on noninterference have followed several different approaches, ranging from the application of type theory [ZM04] and abstract interpretation [GM18] to control flow analysis and equivalence or model checking [FPR02, Mar03, AB11].

Noninterference guarantees that low-level agents cannot infer from their observations what high-level ones are doing. Regardless of its specific definition, noninterference is closely tied to the notion of behavioral equivalence [Gla01], because the idea is to compare the

Key words and phrases: Security, Noninterference, Reversibility, Process Calculi, Branching Bisimilarity.

Preprint submitted to
Logical Methods in Computer Science

system behavior with high-level actions being prevented and the system behavior with the same actions being hidden. One of the most established formal definitions of noninterference properties relies on weak bisimilarity in a process algebraic framework [Mil89], as it naturally lends itself to reason formally about covert channels and illegitimate information flows.

After the classification of security properties in [FG01], the literature concentrated on weak bisimilarity. Here we claim that it is worth studying nondeterministic noninterference in a different setting, relying on branching bisimulation semantics. This was introduced in [GW96] as a refinement of weak bisimilarity to preserve the branching structure of processes also when abstracting from unobservable actions. It features a complete axiomatization whose only τ-axiom is $a \,.\, (\tau \,.\, (x+y)+x) = a \,.\, (x+y)$, where dot stands for action prefix, plus stands for nondeterministic choice, a is an action, τ is an unobservable action, and x and y are process terms. Moreover, while weak bisimilarity can be verified in $O(n^2 \cdot m \cdot \log n)$, where m is the number of transitions and n is the number of states of the labeled transition system underlying the process at hand, branching bisimilarity can be verified more efficiently. An $O(m \cdot n)$ algorithm was provided in [GV90] and, more recently, an even faster $O(m \cdot \log n)$ algorithm has been developed in [JGKW20].

A clear motivation for switching from weak to branching bisimilarity is provided by reversible processes, whose computational model features both forward and backward computations [Lan61, Ben73]. Reversible computing has turned out to have interesting applications in biochemical reaction modeling [PUY12, Pin17], parallel discrete-event simulation [PP14, SOJB18], robotics [LES18], control theory [SPP19], wireless communications [SPP19], fault-tolerant systems [DK05, VKH10, LLM$^+$13, VS18], and concurrent program debugging [GLM14, LNPV18]. To the best of our knowledge, no information flow security approach exists for this setting, in which we will see that weak bisimilarity does not represent a proper tool for the comprehensive analysis of covert channels.

Behavioral equivalences for reversible processes must take into account the fact that computations are allowed to proceed not only forward, but also backward. To this aim, back-and-forth bisimilarity, introduced in [DMV90], requires that two systems are able to mimic each other's behavior stepwise not only in performing actions that follow the arrows of the labeled transition systems, but also in undoing those actions when going backward. Formally, back-and-forth bisimulations are defined on computation paths instead of states thus preserving not only causality but also history, as backward moves are constrained to take place along the same path followed in the forward direction even in the presence of concurrency. In [DMV90] it was shown that strong back-and-forth bisimilarity coincides with the usual notion of strong bisimilarity, while weak back-and-forth bisimilarity is surprisingly finer than standard weak bisimilarity, and it coincides with branching bisimilarity. In particular, this latter result will allow us to investigate the nature of covert channels in reversible systems by using a standard process calculus, i.e., without having to decorate executed actions like in [PU07, BR23] or store them into stack-based memories like in [DK04].

Once established that branching bisimilarity enables noninterference analysis of reversible systems, the novel contribution of this paper is the study of noninterference security properties based on branching bisimilarity. In addition to investigating preservation and compositionality features, we compare the resulting properties with those based on weak bisimilarity [FG01] and we develop a taxonomy of the former that can be naturally applied to those based on weak back-and-forth bisimilarity for reversible systems. Moreover, we illustrate how, in the setting of reversible systems, weak bisimilarity does not provide a

proper framework for the identification of subtle covert channels, while branching bisimilarity does. This is carried out through a database management system example.

This paper, which is a revised and extended version of [EAB23], is organized as follows. In Section 2 we recall background definitions and results for several bisimulation equivalences as well as a number of information-flow security properties based on weak bisimilarity, along with a process language to formalize those properties. In Section 3 we introduce the database management system example. In Section 4, after recasting the same information-flow security properties in terms of branching bisimilarity, we present some results about the preservation of those properties under branching bisimilarity and their compositionality with respect to the operators of the considered language, then we study the relationships among all the previously discussed properties and summarize them in a new taxonomy. In Section 5 we recall the notion of back-and-forth bisimulation and its relationship with the aforementioned bisimulations, emphasizing that weak back-and-forth bisimilarity coincides with branching bisimilarity, which allows us to apply our results to reversible systems. In Section 6 we add reversibility to the database management system example to illustrate the need of branching-bisimilarity-based noninterference. Finally, in Section 7 we provide some concluding remarks and directions for future work.

2. Background Definitions and Results

In this section we recall bisimulation equivalences (Section 2.1) and introduce a basic process language (Section 2.2) through which we will express bisimulation-based information-flow security properties (Section 2.3).

2.1. **Bisimulation Equivalences.**

Bisimilarity is one of the most important behavioral equivalences [Gla01]. To represent the behavior of a process, we use a labeled transition system [Kel76], which is a state-transition graph whose transitions are labeled with actions.

Definition 2.1. A *labeled transition system (LTS)* is a triple $(\mathcal{S}, \mathcal{A}_\tau, \longrightarrow)$ where $\mathcal{S}$ is a nonempty, at most countable set of states, $\mathcal{A}_\tau = \mathcal{A} \cup \{\tau\}$ is a countable set of actions including an unobservable action denoted by τ, and $\longrightarrow \subseteq \mathcal{S} \times \mathcal{A}_\tau \times \mathcal{S}$ is a transition relation. ■

A transition (s, a, s') is written $s \xrightarrow{a} s'$, where s is the source state, a is the transition label, and s' is the target state, in which case we say that s' is reachable from s via that a-transition. In general, we say that s' is reachable from s, written $s' \in \mathit{reach}(s)$, iff $s' = s$ or there is a finite sequence of transitions such that the target state of each of them coincides with the source state of the subsequent one, with the source of the first one being s and the target of the last one being s'.

Strong bisimilarity [Par81, Mil89] identifies processes that are able to mimic each other's behavior stepwise. Unlike trace equivalence, only processes with the same branching structure can be equated. For instance, $a \,.\, (x + y)$ and $a \,.\, x + a \,.\, y$ are told apart unless $x = y$.

Definition 2.2. Let $(\mathcal{S}, \mathcal{A}_\tau, \longrightarrow)$ be an LTS and $s_1, s_2 \in \mathcal{S}$. We say that s_1 and s_2 are *strongly bisimilar*, written $s_1 \sim s_2$, iff $(s_1, s_2) \in \mathcal{B}$ for some strong bisimulation $\mathcal{B}$. A symmetric binary relation $\mathcal{B}$ over $\mathcal{S}$ is a *strong bisimulation* iff, whenever $(s_1, s_2) \in \mathcal{B}$, then:

- for each $s_1 \xrightarrow{a} s_1'$ there exists $s_2 \xrightarrow{a} s_2'$ such that $(s_1', s_2') \in \mathcal{B}$. ■

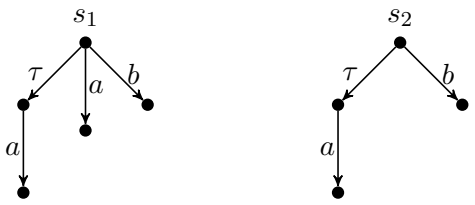

Figure 1: States s_1 and s_2 are weakly bisimilar but not branching bisimilar

Weak bisimilarity [Mil89] is coarser than strong bisimilarity because it is capable of abstracting from unobservable actions, which are denoted by τ. As an example, $a \,.\, \tau \,.\, x = a \,.\, x$. Let $s \stackrel{\tau^*}{\Longrightarrow} s'$ mean that $s' \in \mathit{reach}(s)$ and, whenever $s' \neq s$, there is a finite sequence of transitions that starts in s and terminates in s', where each transition is labeled with τ. Moreover, let $\stackrel{\tau^*}{\Longrightarrow} \stackrel{a}{\longrightarrow} \stackrel{\tau^*}{\Longrightarrow}$ mean that an a-transition is possibly preceded and followed by finitely many τ-transitions.

Definition 2.3. Let $(\mathcal{S}, \mathcal{A}_\tau, \longrightarrow)$ be an LTS and $s_1, s_2 \in \mathcal{S}$. We say that s_1 and s_2 are *weakly bisimilar*, written $s_1 \approx s_2$, iff $(s_1, s_2) \in \mathcal{B}$ for some weak bisimulation $\mathcal{B}$. A symmetric binary relation $\mathcal{B}$ over $\mathcal{S}$ is a *weak bisimulation* iff, whenever $(s_1, s_2) \in \mathcal{B}$, then:

- for each $s_1 \stackrel{\tau}{\longrightarrow} s_1'$ there exists $s_2 \stackrel{\tau^*}{\Longrightarrow} s_2'$ such that $(s_1', s_2') \in \mathcal{B}$;
- for each $s_1 \stackrel{a}{\longrightarrow} s_1'$ with $a \in \mathcal{A}$ there exists $s_2 \stackrel{\tau^*}{\Longrightarrow} \stackrel{a}{\longrightarrow} \stackrel{\tau^*}{\Longrightarrow} s_2'$ such that $(s_1', s_2') \in \mathcal{B}$. ■

Branching bisimilarity [GW96], which is coarser than strong bisimilarity too, is finer than weak bisimilarity because it preserves the branching structure of processes even when abstracting from τ-actions – see the condition $(s_1, \bar{s}_2) \in \mathcal{B}$ in the definition below.

Definition 2.4. Let $(\mathcal{S}, \mathcal{A}_\tau, \longrightarrow)$ be an LTS and $s_1, s_2 \in \mathcal{S}$. We say that s_1 and s_2 are *branching bisimilar*, written $s_1 \approx_{\mathrm{b}} s_2$, iff $(s_1, s_2) \in \mathcal{B}$ for some branching bisimulation $\mathcal{B}$. A symmetric binary relation $\mathcal{B}$ over $\mathcal{S}$ is a *branching bisimulation* iff, whenever $(s_1, s_2) \in \mathcal{B}$, then:

- for each $s_1 \stackrel{a}{\longrightarrow} s_1'$:
  - either $a = \tau$ and $(s_1', s_2) \in \mathcal{B}$;
  - or there exists $s_2 \stackrel{\tau^*}{\Longrightarrow} \bar{s}_2 \stackrel{a}{\longrightarrow} s_2'$ such that $(s_1, \bar{s}_2) \in \mathcal{B}$ and $(s_1', s_2') \in \mathcal{B}$. ■

An example that highlights the higher distinguishing power of branching bisimilarity is given in Figure 1, where the LTS is depicted as a directed graph in which vertices represent states and action-labeled edges represent transitions. The states s_1 and s_2 are weakly bisimilar but not branching bisimilar. The only transition that distinguishes s_1 from s_2 is the a-transition of s_1, which can be mimicked by s_2 according to weak bisimilarity by performing its τ-transition followed by its a-transition. However, s_2 cannot respond in the same way according to branching bisimilarity. If s_2 performs the τ-transition followed by the a-transition, then the state reached after the τ-transition should be branching bisimilar to s_1, which is not the case because of the b-transition departing from s_1.

2.2. **A Process Language with High and Low Actions.** We now introduce a basic process calculus to formalize the security properties of interest. To address two security levels, actions are divided into high and low. We partition the set $\mathcal{A}$ of observable actions

Prefix	$a \,.\, P \xrightarrow{a} P$
Choice	$\dfrac{P_1 \xrightarrow{a} P_1'}{P_1 + P_2 \xrightarrow{a} P_1'} \qquad \dfrac{P_2 \xrightarrow{a} P_2'}{P_1 + P_2 \xrightarrow{a} P_2'}$
Interleaving	$\dfrac{P_1 \xrightarrow{a} P_1' \quad a \notin L}{P_1 \parallel_L P_2 \xrightarrow{a} P_1' \parallel_L P_2} \qquad \dfrac{P_2 \xrightarrow{a} P_2' \quad a \notin L}{P_1 \parallel_L P_2 \xrightarrow{a} P_1 \parallel_L P_2'}$
Synchronization	$\dfrac{P_1 \xrightarrow{a} P_1' \quad P_2 \xrightarrow{a} P_2' \quad a \in L}{P_1 \parallel_L P_2 \xrightarrow{a} P_1' \parallel_L P_2'}$
Restriction	$\dfrac{P \xrightarrow{a} P' \quad a \notin L}{P \setminus L \xrightarrow{a} P' \setminus L}$
Hiding	$\dfrac{P \xrightarrow{a} P' \quad a \in L}{P / L \xrightarrow{\tau} P' / L} \qquad \dfrac{P \xrightarrow{a} P' \quad a \notin L}{P / L \xrightarrow{a} P' / L}$
Constant	$\dfrac{C \triangleq P \quad P \xrightarrow{a} P'}{C \xrightarrow{a} P'}$

Table 1: Operational semantic rules

into $\mathcal{A}_\mathcal{H} \cup \mathcal{A}_\mathcal{L}$, with $\mathcal{A}_\mathcal{H} \cap \mathcal{A}_\mathcal{L} = \emptyset$, where $\mathcal{A}_\mathcal{H}$ is the set of high-level actions, ranged over by h, and $\mathcal{A}_\mathcal{L}$ is the set of low-level actions, ranged over by l. Note that $\tau \notin \mathcal{A}_\mathcal{H} \cup \mathcal{A}_\mathcal{L}$.

The set $\mathbb{P}$ of process terms is obtained by considering typical operators from CCS [Mil89] and CSP [BHR84]. In addition to prefix and choice, we have restriction and hiding as they are necessary to formalize noninterference properties, the CSP parallel composition so as not to turn into τ the synchronization between high-level actions as would happen with the CCS parallel composition, and recursion (which was not considered in [EAB23]). The syntax is:

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

where:

- $\underline{0}$ is the terminated process.
- $a \,.\, \_$, for $a \in \mathcal{A}_\tau$, is the action prefix operator describing a process that initially performs action a.
- $\_ + \_$ is the alternative composition operator expressing a nondeterministic choice between two processes based on their initially executable actions.
- $\_ \parallel_L \_$, for $L \subseteq \mathcal{A}$, is the parallel composition operator forcing two processes to proceed independently on every action not in L and to synchronize on every action in L.
- $\_ \setminus L$, for $L \subseteq \mathcal{A}$, is the restriction operator, which prevents the execution of actions in L.
- $\_ / L$, for $L \subseteq \mathcal{A}$, is the hiding operator, which turns all the executed actions in L into the unobservable action τ.
- C is a process constant equipped with a defining equation of the form $C \triangleq P$, where every constant possibly occurring in P – including C itself thus allowing for recursion – must be in the scope of an action prefix operator.

The operational semantic rules for the process language are shown in Table 1 and produce the LTS $(\mathbb{P}, \mathcal{A}_\tau, \longrightarrow)$ where $\longrightarrow \subseteq \mathbb{P} \times \mathcal{A}_\tau \times \mathbb{P}$, to which the bisimulation equivalences defined in the previous section are applicable.

2.3. **Weak-Bisimilarity-Based Information-Flow Properties.** 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 visible by any agent at the low security level. Below is a representative selection of weak-bisimilarity-based noninterference properties – *Nondeterministic Non-Interference* (NNI) and *Non-Deducibility on Composition* (NDC) – whose definitions and relationships are recalled from [FG01] and, as far as P_BNDC is concerned, from [FR06] (this last property was not considered in [EAB23]).

Definition 2.5. 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 $Q' \in \mathit{reach}(Q)$ has only actions in $\mathcal{A}_\mathcal{H}$ and for all $L \subseteq \mathcal{A}_\mathcal{H}$, $P \setminus \mathcal{A}_\mathcal{H} \approx ((P \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}$.
- $P \in \text{SBSNNI} \iff$ for all $P' \in \mathit{reach}(P)$, $P' \in \text{BSNNI}$.
- $P \in \text{P\_BNDC} \iff$ for all $P' \in \mathit{reach}(P)$, $P' \in \text{BNDC}$.
- $P \in \text{SBNDC} \iff$ for all $P' \in \mathit{reach}(P)$ and for all P'' for which there exists $h \in \mathcal{A}_\mathcal{H}$ such that $P' \xrightarrow{h} P''$, $P' \setminus \mathcal{A}_\mathcal{H} \approx P'' \setminus \mathcal{A}_\mathcal{H}$. ■

Theorem 2.6. $\text{SBNDC} \subset \text{SBSNNI} = \text{P\_BNDC} \subset \text{BNDC} \subset \text{BSNNI}$. ■

Historically, one of the first and most intuitive proposals has been *Bisimulation-based Strong Nondeterministic Non-Interference* (BSNNI). Basically, it is satisfied by any process P that behaves the same when its high-level actions are prevented (as modeled by $P \setminus \mathcal{A}_\mathcal{H}$) or when they are considered as hidden, unobservable actions (as modeled by $P / \mathcal{A}_\mathcal{H}$). The equivalence between these two low-level views of P states that a low-level agent cannot observe the high-level behavior of the system. For instance, in $l \,.\, \underline{0} + h \,.\, l \,.\, \underline{0}$ a low-level agent that observes the execution of l cannot infer anything about the execution of h. Indeed, $(l \,.\, \underline{0} + h \,.\, l \,.\, \underline{0}) \setminus \{h\} \approx (l \,.\, \underline{0} + h \,.\, l \,.\, \underline{0}) / \{h\}$ because the former process is isomorphic to $l \,.\, \underline{0}$, the latter process is isomorphic to $l \,.\, \underline{0} + \tau \,.\, l \,.\, \underline{0}$, and $l \,.\, \underline{0} \approx l \,.\, \underline{0} + \tau \,.\, l \,.\, \underline{0}$.

BSNNI is not powerful enough to detect information leakages that derive from the behavior of a high-level agent interacting with the system. For instance, $l \,.\, \underline{0} + h_1 \,.\, h_2 \,.\, l \,.\, \underline{0}$ is BSNNI for the same reason discussed above. However, a high-level agent like $h_1 \,.\, \underline{0}$ enables h_1 and then disables h_2, thus turning the low-level view of the system into $l \,.\, \underline{0} + \tau \,.\, \underline{0}$, which is clearly distinguishable from $l \,.\, \underline{0}$, as only in the former the low-level observer may not observe l. To overcome such a limitation, the most obvious solution consists of checking explicitly the interaction on any action set $L \subseteq \mathcal{A}_\mathcal{H}$ between the system and every possible high-level agent. The resulting property is *Bisimulation-based Non-Deducibility on Composition* (BNDC), which features a universal quantification over Q containing only high-level actions.

To circumvent the verification problems related to such a quantifier, several properties have been proposed that are stronger than BNDC. They all express some persistency conditions, stating that the security checks have to be extended to all the processes reachable from a secure one. Three of the most representative among such properties are: the variant of BSNNI that requires every reachable process to satisfy BSNNI itself, called *Strong* BSNNI (SBSNNI); the variant of BNDC that requires every reachable process to satisfy BNDC itself, called *Persistent* BNDC (P_BNDC); and *Strong* BNDC (SBNDC), which requires the low-level view of every reachable process to be the same before and after the execution of any high-level action, meaning that the execution of high-level actions must be completely transparent to low-level agents. We remind that P_BNDC and SBSNNI have been proven to be equivalent in [FR06].

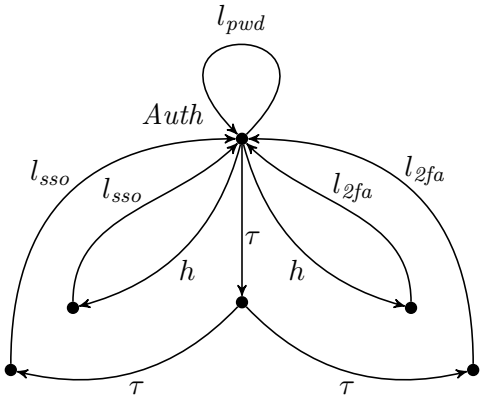

Figure 2: LTS underlying $Auth$

3. Use Case: DBMS Authentication – Part I

Consider a multi-threaded system supporting the execution of concurrent transactions operating on a healthcare database, where only authorized users can write their data. Then, depending on a policy governed by the database management system (DBMS), such data can be shared with a dedicated module feeding the training set of a machine learning (ML) facility, which is responsible for building a trained model for data analysis purposes.

On the one hand, different authentication mechanisms can be employed to identify users and ensure data authenticity for each transaction. We address a simple password-based mechanism (*pwd*), a more sophisticated two-factor authentication system (*2fa*), and finally a scheme based on single sign on (*sso*) [Boo20].

On the other hand, for security reasons related to sharing sensitive data with the ML module [AC19], only data transmitted through highly secure mechanisms, i.e., *2fa* and *sso*, can be used to feed the training set. In any case, for privacy issues, users must not be aware of whether their data are actually chosen to train the ML model or not [BFLX22]. Hence, to avoid that the use of highly secure authentication implicitly reveals the involvement of the ML module, the DBMS internally decides not to consider certain transactions for the training set.

For the sake of simplicity, we concentrate on the authentication policy followed by the DBMS whenever handling a write transaction. Therefore, we abstract away from the description of the ML module and of the database access operations. In particular, we consider the following process term $Auth$, whose LTS is depicted in Figure 2:

$$\begin{aligned} Auth \triangleq\ & l_{pwd} \,.\, Auth + \\ & (h \,.\, l_{sso} \,.\, Auth + h \,.\, l_{2fa} \,.\, Auth) + \\ & \tau \,.\, (\tau \,.\, l_{sso} \,.\, Auth + \tau \,.\, l_{2fa} \,.\, Auth) \end{aligned}$$

The actions $l_\star$ express that the transaction is conducted under the authentication method represented by $\star$. We treat them as low-level actions because they represent interactions between the users and the DBMS. The action h represents an interaction between the DBMS and the ML module, which is deemed to be high level as the activities of the ML module must be transparent to the users.

The first subterm of $Auth$ specifies that the DBMS is ready to offer the password-based mechanism, in which case the transaction data will not be passed to the ML module. The

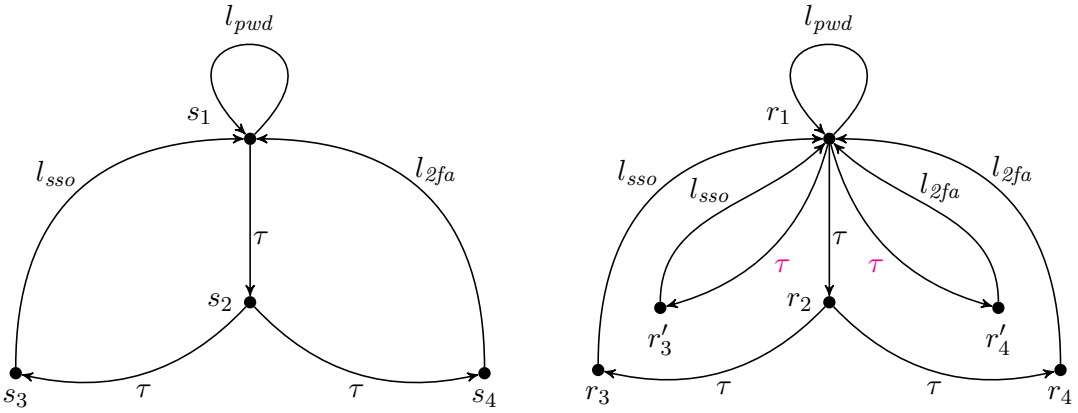

Figure 3: LTSs of the low-level views of $Auth$: $Auth \setminus \mathcal{A}_{\mathcal{H}}$ (left) and $Auth \,/\, \mathcal{A}_{\mathcal{H}}$ (right)

second subterm models the communication with the ML module informing that the transaction data – which must be protected through one of the two highly secure authentication mechanisms – will be included in the training set. Note that in this case the choice of the specific authentication method offered by the DBMS is nondeterministic and does not include the password-based mechanism. The third subterm specifies that the DBMS decides internally, through the first action τ, that the transaction data will not be passed to the ML module, even if the authentication method (chosen nondeterministically) is highly secure. Hence, in this case no interaction with the ML module is needed. The aim of this subterm is to mimic the behavior of the second subterm, thus acting as an obfuscation mechanism that shall not allow any user to detect the potential involvement of the ML module by simply observing the used authentication method.

Formally, the success of this obfuscation is guaranteed if the interaction with the ML module does not interfere with the low-level view of the system observed by any user, which can be verified as a noninterference property. More specifically, the ML module represents the high-level portion of the system that is expected not to interfere with the low-level behavior of any user interacting with the DBMS, thus justifying the use of the high-level action h modeling the interaction between such a module and the DBMS.

As far as $\approx$-based noninterference is concerned, $Auth$ does not leak any information from the high level to the low level. More precisely, the system is SBSNNI, and hence also BNDC and BSNNI by virtue of Theorem 2.6. Indeed, by observing Figure 3 – where the h-actions are forbidden on the left while they are transformed into the colored τ-actions on the right – it is easy to see that $Auth$ is BSNNI, i.e., $Auth \setminus \mathcal{A}_{\mathcal{H}} \approx Auth \,/\, \mathcal{A}_{\mathcal{H}}$. The weak bisimulation relating the two low-level views of $Auth$ is given by the following partition of the disjoint union of the two state spaces:

$$\{\{s_1, r_1\}, \{s_2, r_2\}, \{s_3, r_3, r'_3\}, \{s_4, r_4, r'_4\}\}$$

Since the only high-level action is enabled at the initial state of $Auth$, it then follows that $Auth$ is SBSNNI as well.

4. Security Properties Based on Branching Bisimilarity

While the literature on noninterference mainly concentrates on weak bisimulation semantics, in this article we address information-flow security in terms of branching bisimilarity.

Definition 4.1. BrSNNI, BrNDC, SBrSNNI, P_BrNDC, and SBrNDC are obtained from the corresponding properties in Definition 2.5 by replacing the weak bisimilarity check ($\approx$) with the branching bisimilarity check ($\approx_{\rm b}$). ■

In this section we first study their preservation and compositionality characteristics so as to assess their usefulness (Section 4.1) and then we investigate the relationships among them and with the corresponding properties based on weak bisimilarity (Section 4.2).

4.1. Preservation and Compositionality.

Similar to the weak bisimilarity case [FG01], all the $\approx_{\rm b}$-based noninterference properties turn out to be preserved by $\approx_{\rm b}$. This means that, whenever a process P_1 is secure under any of such properties, then every other branching bisimilar process P_2 is secure too according to the same property. This is very useful for automated property verification, as it allows one to work with the process with the smallest state space among the equivalent ones. To establish preservation, we first have to prove that $\approx_{\rm b}$ is a congruence with respect to $\_ \setminus L$, $\_ / L$, and $\_ \|_L \_$ for an arbitrary $L \subseteq \mathcal{A}$, because these three operators were not considered in the congruence results of [GW96, Gla93].

Lemma 4.2. *Let $P_1, P_2 \in \mathbb{P}$. If $P_1 \approx_{\rm b} P_2$ then:*

- *$P_1 \setminus L \approx_{\rm b} P_2 \setminus L$ for all $L \subseteq \mathcal{A}$.*
- *$P_1 / L \approx_{\rm b} P_2 / L$ for all $L \subseteq \mathcal{A}$.*
- *$P_1 \|_L P \approx_{\rm b} P_2 \|_L P$ and $P \|_L P_1 \approx_{\rm b} P \|_L P_2$ for all $L \subseteq \mathcal{A}$ and $P \in \mathbb{P}$.*

Proof. Let $\mathcal{B}$ be a branching bisimulation containing the pairs (P_1, P_2) and (P_2, P_1) and consider an arbitrary set $L \subseteq \mathcal{A}$ and an arbitrary process $P \in \mathbb{P}$. We have that:

- $P_1 \setminus L \approx_{\rm b} P_2 \setminus L$ because the symmetric relation $\mathcal{B}' = \{(Q_1 \setminus L, Q_2 \setminus L) \mid (Q_1, Q_2) \in \mathcal{B}\}$ is a branching bisimulation too, as we now show via the following two cases based on the operational semantic rules in Table 1:
  - If $Q_1 \setminus L \xrightarrow{\tau} Q_1' \setminus L$ with $Q_1 \xrightarrow{\tau} Q_1'$, then either $(Q_1', Q_2) \in \mathcal{B}$, or there exist $\bar{Q}_2$ and Q_2' such that $Q_2 \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \xrightarrow{\tau} Q_2'$ with $(Q_1, \bar{Q}_2) \in \mathcal{B}$ and $(Q_1', Q_2') \in \mathcal{B}$. Since the restriction operator does not apply to τ, in the former subcase $Q_2 \setminus L$ is allowed to stay idle with $(Q_1' \setminus L, Q_2 \setminus L) \in \mathcal{B}'$, while in the latter subcase $Q_2 \setminus L \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \setminus L \xrightarrow{\tau} Q_2' \setminus L$, with $(Q_1 \setminus L, \bar{Q}_2 \setminus L) \in \mathcal{B}'$ and $(Q_1' \setminus L, Q_2' \setminus L) \in \mathcal{B}'$.
  - If $Q_1 \setminus L \xrightarrow{a} Q_1' \setminus L$ with $Q_1 \xrightarrow{a} Q_1'$ and $a \notin L \cup \{\tau\}$, then there exist $\bar{Q}_2$ and Q_2' such that $Q_2 \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \xrightarrow{a} Q_2'$ with $(Q_1, \bar{Q}_2) \in \mathcal{B}$ and $(Q_1', Q_2') \in \mathcal{B}$. Since the restriction operator does not apply to τ and $a \notin L$, it follows that $Q_2 \setminus L \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \setminus L \xrightarrow{a} Q_2' \setminus L$ with $(Q_1 \setminus L, \bar{Q}_2 \setminus L) \in \mathcal{B}'$ and $(Q_1' \setminus L, Q_2' \setminus L) \in \mathcal{B}'$.
- $P_1 / L \approx_{\rm b} P_2 / L$ because the symmetric relation $\mathcal{B}' = \{(Q_1 / L, Q_2 / L) \mid (Q_1, Q_2) \in \mathcal{B}\}$ is a branching bisimulation too, as we now show via the following two cases based on the operational semantic rules in Table 1:
  - If $Q_1 / L \xrightarrow{\tau} Q_1' / L$ with $Q_1 \xrightarrow{\tau} Q_1'$, then either $(Q_1', Q_2) \in \mathcal{B}$, or there exist $\bar{Q}_2$ and Q_2' such that $Q_2 \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \xrightarrow{\tau} Q_2'$ with $(Q_1, \bar{Q}_2) \in \mathcal{B}$ and $(Q_1', Q_2') \in \mathcal{B}$. Since the hiding operator does not apply to τ, in the former subcase Q_2 / L is allowed to stay idle

with $(Q_1' \,/\, L, Q_2 \,/\, L) \in \mathcal{B}'$, while in the latter subcase $Q_2 \,/\, L \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \,/\, L \stackrel{\tau}{\longrightarrow} Q_2' \,/\, L$ with $(Q_1 \,/\, L, \bar{Q}_2 \,/\, L) \in \mathcal{B}'$ and $(Q_1' \,/\, L, Q_2' \,/\, L) \in \mathcal{B}'$.

- If $Q_1 \,/\, L \stackrel{a}{\longrightarrow} Q_1' \,/\, L$ with $Q_1 \stackrel{b}{\longrightarrow} Q_1'$ and $b \in L \wedge a = \tau$ or $b \notin L \cup \{\tau\} \wedge a = b$, then there exist $\bar{Q}_2$ and Q_2' such that $Q_2 \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \stackrel{b}{\longrightarrow} Q_2'$ with $(Q_1, \bar{Q}_2) \in \mathcal{B}$ and $(Q_1', Q_2') \in \mathcal{B}$. Since the hiding operator does not apply to τ, it follows that $Q_2 \,/\, L \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \,/\, L \stackrel{a}{\longrightarrow} Q_2' \,/\, L$, with $(Q_1 \,/\, L, \bar{Q}_2 \,/\, L) \in \mathcal{B}'$ and $(Q_1' \,/\, L, Q_2' \,/\, L) \in \mathcal{B}'$.

• $P_1 \parallel_L P \approx_{\rm b} P_2 \parallel_L P$ because the symmetric relation $\mathcal{B}' = \{(Q_1 \parallel_L Q, Q_2 \parallel_L Q) \mid (Q_1, Q_2) \in \mathcal{B} \wedge Q \in \mathbb{P}\}$ is a branching bisimulation too, as we now show via the following three cases based on the operational semantic rules in Table 1:

- If $Q_1 \parallel_L Q \stackrel{a}{\longrightarrow} Q_1' \parallel_L Q$ with $Q_1 \stackrel{a}{\longrightarrow} Q_1'$ and $a \notin L$, then either $a = \tau$ and $(Q_1', Q_2) \in \mathcal{B}$, or there exist $\bar{Q}_2$ and Q_2' such that $Q_2 \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \stackrel{a}{\longrightarrow} Q_2'$ with $(Q_1, \bar{Q}_2) \in \mathcal{B}$ and $(Q_1', Q_2') \in \mathcal{B}$. In the former subcase $Q_2 \parallel_L Q$ is allowed to stay idle with $(Q_1' \parallel_L Q, Q_2 \parallel_L Q) \in \mathcal{B}'$, while in the latter subcase $Q_2 \parallel_L Q \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \parallel_L Q \stackrel{a}{\longrightarrow} Q_2' \parallel_L Q$ with $(Q_1 \parallel_L Q, \bar{Q}_2 \parallel_L Q) \in \mathcal{B}'$ and $(Q_1' \parallel_L Q, Q_2' \parallel_L Q) \in \mathcal{B}'$.
- The case $Q_1 \parallel_L Q \stackrel{a}{\longrightarrow} Q_1 \parallel_L Q'$ with $Q \stackrel{a}{\longrightarrow} Q'$ and $a \notin L$ is trivial.
- If $Q_1 \parallel_L Q \stackrel{a}{\longrightarrow} Q_1' \parallel_L Q'$ with $Q_1 \stackrel{a}{\longrightarrow} Q_1'$, $Q \stackrel{a}{\longrightarrow} Q'$, and $a \in L$, then there exist $\bar{Q}_2$ and Q_2' such that $Q_2 \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \stackrel{a}{\longrightarrow} Q_2'$ with $(Q_1, \bar{Q}_2) \in \mathcal{B}$ and $(Q_1', Q_2') \in \mathcal{B}$. Thus $Q_2 \parallel_L Q \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_2 \parallel_L Q \stackrel{a}{\longrightarrow} Q_2' \parallel_L Q'$ with $(Q_1 \parallel_L Q, \bar{Q}_2 \parallel_L Q) \in \mathcal{B}'$ and $(Q_1' \parallel_L Q', Q_2' \parallel_L Q') \in \mathcal{B}'$.

The proof of $P \parallel_L P_1 \approx_{\rm b} P \parallel_L P_2$ is similar. □

Theorem 4.3. *Let $P_1, P_2 \in \mathbb{P}$ and $\mathcal{P} \in \{\text{BrSNNI}, \text{BrNDC}, \text{SBrSNNI}, \text{P\_BrNDC}, \text{SBrNDC}\}$. If $P_1 \approx_{\rm b} P_2$, then $P_1 \in \mathcal{P} \iff P_2 \in \mathcal{P}$.*

Proof. A straightforward consequence of the definition of the five properties, i.e., Definition 4.1, and Lemma 4.2. □

As far as modular verification is concerned, like in the weak-bisimilarity-based case [FG01] only the local properties SBrSNNI and SBrNDC are compositional, i.e., are preserved by some operators of the calculus in certain circumstances; this holds also for P_BrNDC because we will see later on that P_BrNDC coincides with SBrSNNI (Theorem 4.8). Unlike the compositionality results presented in [FG01], ours are related not only to parallel composition and restriction, but also to action prefix and hiding. Like in the weak-bisimilarity-based case [FG01], no property relying on branching bisimilarity is compositional with respect to alternative composition. For instance, let us consider processes P_1 and P_2 respectively given by $l \,.\, \underline{0}$ and $h \,.\, \underline{0}$. Both are BrSNNI, as $l \,.\, \underline{0} \setminus \{h\} \approx_{\rm b} l \,.\, \underline{0} \,/\, \{h\}$ and $h \,.\, \underline{0} \setminus \{h\} \approx_{\rm b} h \,.\, \underline{0} \,/\, \{h\}$, but $P_1 + P_2 \notin \text{BrSNNI}$ because $(l \,.\, \underline{0} + h \,.\, \underline{0}) \setminus \{h\} \approx_{\rm b} l \,.\, \underline{0} \not\approx_{\rm b} l \,.\, \underline{0} + \tau \,.\, \underline{0} \approx_{\rm b} (l \,.\, \underline{0} + h \,.\, \underline{0}) \,/\, \{h\}$. It can be easily checked that $P_1 + P_2 \notin \mathcal{P}$ also for $\mathcal{P} \in \{\text{BrNDC}, \text{SBrSNNI}, \text{P\_BrNDC}, \text{SBrNDC}\}$.

Compositionality with respect to parallel composition $\parallel_L$ is limited, for SBrSNNI and P_BrNDC, to the case in which no synchronization can take place between high-level actions, i.e., $L \subseteq \mathcal{A}_{\mathcal{L}}$, while in [FG01] the compositionality of SBSNNI holds for every $L \subseteq \mathcal{A}$. As an example, P_1 given by $h \,.\, \underline{0} + l_1 \,.\, \underline{0} + \tau \,.\, \underline{0}$ and P_2 given by $h \,.\, \underline{0} + l_2 \,.\, \underline{0} + \tau \,.\, \underline{0}$ are SBrSNNI, but $P_1 \parallel_{\{h\}} P_2$ is not because the transition $(P_1 \parallel_{\{h\}} P_2) \,/\, \mathcal{A}_{\mathcal{H}} \stackrel{\tau}{\longrightarrow} (\underline{0} \parallel_{\{h\}} \underline{0}) \,/\, \mathcal{A}_{\mathcal{H}}$ arising from the synchronization between the two h-actions cannot be matched by $(P_1 \parallel_{\{h\}} P_2) \setminus \mathcal{A}_{\mathcal{H}}$ in the branching bisimulation game. As a matter of fact, the only two possibilities are $(P_1 \parallel_{\{h\}} P_2) \setminus \mathcal{A}_{\mathcal{H}} \stackrel{\tau^*}{\Longrightarrow} (P_1 \parallel_{\{h\}} P_2) \setminus \mathcal{A}_{\mathcal{H}} \stackrel{\tau}{\longrightarrow} (\underline{0} \parallel_{\{h\}} P_2) \setminus \mathcal{A}_{\mathcal{H}} \stackrel{\tau}{\longrightarrow} (\underline{0} \parallel_{\{h\}} \underline{0}) \setminus \mathcal{A}_{\mathcal{H}}$ as well as $(P_1 \parallel_{\{h\}} P_2) \setminus \mathcal{A}_{\mathcal{H}} \stackrel{\tau^*}{\Longrightarrow} (P_1 \parallel_{\{h\}} P_2) \setminus \mathcal{A}_{\mathcal{H}} \stackrel{\tau}{\longrightarrow} (P_1 \parallel_{\{h\}} \underline{0}) \setminus \mathcal{A}_{\mathcal{H}} \stackrel{\tau}{\longrightarrow} (\underline{0} \parallel_{\{h\}} \underline{0}) \setminus \mathcal{A}_{\mathcal{H}}$ but neither

$(\underline{0} \parallel_{\{h\}} P_2) \setminus \mathcal{A}_\mathcal{H}$ nor $(P_1 \parallel_{\{h\}} \underline{0}) \setminus \mathcal{A}_\mathcal{H}$ is branching bisimilar to $(P_1 \parallel_{\{h\}} P_2) \setminus \mathcal{A}_\mathcal{H}$ when $l_1 \neq l_2$. Note that $(P_1 \parallel_{\{h\}} P_2) / \mathcal{A}_\mathcal{H} \approx (P_1 \parallel_{\{h\}} P_2) \setminus \mathcal{A}_\mathcal{H}$ because $(P_1 \parallel_{\{h\}} P_2) / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} (\underline{0} \parallel_{\{h\}} \underline{0}) / \mathcal{A}_\mathcal{H}$ is matched by $(P_1 \parallel_{\{h\}} P_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\underline{0} \parallel_{\{h\}} \underline{0}) \setminus \mathcal{A}_\mathcal{H}$. However, it is not only a matter of the higher discriminating power of $\approx_{\rm b}$ with respect to $\approx$. If we used the CCS parallel composition operator [Mil89], which turns into τ the synchronization of two actions thus combining communication with hiding, then the parallel composition of P_1 and P_2 with restriction on $\mathcal{A}_\mathcal{H}$ would be able to respond, in the branching bisimulation game, with a single τ-transition reaching the parallel composition of $\underline{0}$ and $\underline{0}$ with restriction on $\mathcal{A}_\mathcal{H}$.

To establish compositionality, we first prove some ancillary results about parallel composition, restriction, and hiding under SBrSNNI and SBrNDC.

Lemma 4.4. *Let $P_1, P_2, P \in \mathbb{P}$. Then:*

(1) *If $P_1, P_2 \in$ SBrSNNI and $L \subseteq \mathcal{A}_\mathcal{L}$, then $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} (R_1 \parallel_L R_2) / \mathcal{A}_\mathcal{H}$ for all $Q_1, R_1 \in reach(P_1)$ and $Q_2, R_2 \in reach(P_2)$ such that $Q_1 \parallel_L Q_2, R_1 \parallel_L R_2 \in reach(P_1 \parallel_L P_2)$, $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_1 / \mathcal{A}_\mathcal{H}$, and $Q_2 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_2 / \mathcal{A}_\mathcal{H}$.*

(2) *If $P \in$ SBrSNNI and $L \subseteq \mathcal{A}$, then $(Q / \mathcal{A}_\mathcal{H}) \setminus L \approx_{\rm b} (R \setminus L) / \mathcal{A}_\mathcal{H}$ for all $Q, R \in reach(P)$ such that $Q / \mathcal{A}_\mathcal{H} \approx_{\rm b} R \setminus \mathcal{A}_\mathcal{H}$.*

(3) *If $P_1, P_2 \in$ SBrNDC and $L \subseteq \mathcal{A}$, then $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} (R_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}$ for all $Q_1, R_1 \in reach(P_1)$ and $Q_2, R_2 \in reach(P_2)$ such that $Q_1 \parallel_L Q_2, R_1 \parallel_L R_2 \in reach(P_1 \parallel_L P_2)$, $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_1 \setminus \mathcal{A}_\mathcal{H}$ and $Q_2 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_2 \setminus \mathcal{A}_\mathcal{H}$.*

Proof. Let $\mathcal{B}$ be a symmetric relation containing all the pairs of processes that have to be shown to be branching bisimilar according to the property considered among the three stated above:

(1) Starting from $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ and $(R_1 \parallel_L R_2) / \mathcal{A}_\mathcal{H}$ related by $\mathcal{B}$, so that $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_1 / \mathcal{A}_\mathcal{H}$ and $Q_2 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_2 / \mathcal{A}_\mathcal{H}$, in the branching bisimulation game there are twelve cases based on the operational semantic rules in Table 1. In the first five cases, it is $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ to move first:
   - If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \xrightarrow{l} (Q_1' \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ with $Q_1 \xrightarrow{l} Q_1'$ and $l \notin L$, then $Q_1 \setminus \mathcal{A}_\mathcal{H} \xrightarrow{l} Q_1' \setminus \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$. From $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_1 / \mathcal{A}_\mathcal{H}$ it follows that there exist $\bar{R}_1$ and R_1' such that $R_1 / \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{R}_1 / \mathcal{A}_\mathcal{H} \xrightarrow{l} R_1' / \mathcal{A}_\mathcal{H}$ with $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} \bar{R}_1 / \mathcal{A}_\mathcal{H}$ and $Q_1' \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_1' / \mathcal{A}_\mathcal{H}$. Since synchronization does not apply to τ and l, it follows that $(R_1 \parallel_L R_2) / \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{R}_1 \parallel_L R_2) / \mathcal{A}_\mathcal{H} \xrightarrow{l} (R_1' \parallel_L R_2) / \mathcal{A}_\mathcal{H}$ with $((Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (\bar{R}_1 \parallel_L R_2) / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((Q_1' \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (R_1' \parallel_L R_2) / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
   - If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \xrightarrow{l} (Q_1 \parallel_L Q_2') \setminus \mathcal{A}_\mathcal{H}$ with $Q_2 \xrightarrow{l} Q_2'$ and $l \notin L$, then the proof is similar to the one of the previous case.
   - If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \xrightarrow{l} (Q_1' \parallel_L Q_2') \setminus \mathcal{A}_\mathcal{H}$ with $Q_i \xrightarrow{l} Q_i'$ for $i \in \{1, 2\}$ and $l \in L$, then $Q_i \setminus \mathcal{A}_\mathcal{H} \xrightarrow{l} Q_i' \setminus \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$. From $Q_i \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_i / \mathcal{A}_\mathcal{H}$ it follows that there exist $\bar{R}_i$ and R_i' such that $R_i / \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{R}_i / \mathcal{A}_\mathcal{H} \xrightarrow{l} R_i' / \mathcal{A}_\mathcal{H}$ with $Q_i \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} \bar{R}_i / \mathcal{A}_\mathcal{H}$ and $Q_i' \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_i' / \mathcal{A}_\mathcal{H}$. Since synchronization does not apply to τ, it follows that $(R_1 \parallel_L R_2) / \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{R}_1 \parallel_L \bar{R}_2) / \mathcal{A}_\mathcal{H} \xrightarrow{l} (R_1' \parallel_L R_2') / \mathcal{A}_\mathcal{H}$ with $((Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (\bar{R}_1 \parallel_L \bar{R}_2) / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((Q_1' \parallel_L Q_2') \setminus \mathcal{A}_\mathcal{H}, (R_1' \parallel_L R_2') / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
   - If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \xrightarrow{\tau} (Q_1' \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ with $Q_1 \xrightarrow{\tau} Q_1'$, then $Q_1 \setminus \mathcal{A}_\mathcal{H} \xrightarrow{\tau} Q_1' \setminus \mathcal{A}_\mathcal{H}$ as $\tau \notin \mathcal{A}_\mathcal{H}$. From $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_1 / \mathcal{A}_\mathcal{H}$ it follows that either $Q_1' \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} R_1 / \mathcal{A}_\mathcal{H}$, or there

exist $\bar{R}_1$ and R'_1 such that $R_1 \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{R}_1 \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} R'_1 \,/\, \mathcal{A}_\mathcal{H}$ with $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{R}_1 \,/\, \mathcal{A}_\mathcal{H}$ and $Q'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R'_1 \,/\, \mathcal{A}_\mathcal{H}$. In the former subcase $(R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H}$ is allowed to stay idle with $((Q'_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H}) \in \mathcal{B}$, while in the latter subcase, since synchronization does not apply to τ, it follows that $(R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{R}_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (R'_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H}$ with $((Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (\bar{R}_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((Q'_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (R'_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.

- If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (Q_1 \parallel_L Q'_2) \setminus \mathcal{A}_\mathcal{H}$ with $Q_2 \stackrel{\tau}{\longrightarrow} Q'_2$, then the proof is similar to the one of the previous case.

In the other seven cases, instead, it is $(R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H}$ to move first:

- If $(R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (R'_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H}$ with $R_1 \stackrel{l}{\longrightarrow} R'_1$ and $l \notin L$, then $R_1 \,/\, \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} R'_1 \,/\, \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$. From $R_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q_1 \setminus \mathcal{A}_\mathcal{H}$ it follows that there exist $\bar{Q}_1$ and Q'_1 such that $Q_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} Q'_1 \setminus \mathcal{A}_\mathcal{H}$ with $R_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{Q}_1 \setminus \mathcal{A}_\mathcal{H}$ and $R'_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q'_1 \setminus \mathcal{A}_\mathcal{H}$. Since synchronization does not apply to τ and l, it follows that $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{Q}_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (Q'_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ with $((R_1 \parallel_L R_2)/\mathcal{A}_\mathcal{H}, (\bar{Q}_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((R'_1 \parallel_L R_2)/\mathcal{A}_\mathcal{H}, (Q'_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $(R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (R_1 \parallel_L R'_2) \,/\, \mathcal{A}_\mathcal{H}$ with $R_2 \stackrel{l}{\longrightarrow} R'_2$ and $l \notin L$, then the proof is similar to the one of the previous case.
- If $(R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (R'_1 \parallel_L R'_2) \,/\, \mathcal{A}_\mathcal{H}$ with $R_i \stackrel{l}{\longrightarrow} R'_i$ for $i \in \{1, 2\}$ and $l \in L$, then $R_i \,/\, \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} R'_i \,/\, \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$. From $R_i \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q_i \setminus \mathcal{A}_\mathcal{H}$ it follows that there exist $\bar{Q}_i$ and Q'_i such that $Q_i \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_i \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} Q'_i \setminus \mathcal{A}_\mathcal{H}$ with $R_i \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{Q}_i \setminus \mathcal{A}_\mathcal{H}$ and $R'_i \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q'_i \setminus \mathcal{A}_\mathcal{H}$. Since synchronization does not apply to τ, it follows that $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{Q}_1 \parallel_L \bar{Q}_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (Q'_1 \parallel_L Q'_2) \setminus \mathcal{A}_\mathcal{H}$ with $((R_1 \parallel_L R_2)/\mathcal{A}_\mathcal{H}, (\bar{Q}_1 \parallel_L \bar{Q}_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((R'_1 \parallel_L R'_2)/\mathcal{A}_\mathcal{H}, (Q'_1 \parallel_L Q'_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $(R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (R'_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H}$ with $R_1 \stackrel{\tau}{\longrightarrow} R'_1$, then $R_1 \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} R'_1 \,/\, \mathcal{A}_\mathcal{H}$ as $\tau \notin \mathcal{A}_\mathcal{H}$. From $R_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q_1 \setminus \mathcal{A}_\mathcal{H}$ it follows that either $R'_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q_1 \setminus \mathcal{A}_\mathcal{H}$, or there exist $\bar{Q}_1$ and Q'_1 such that $Q_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} Q'_1 \setminus \mathcal{A}_\mathcal{H}$ with $R_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{Q}_1 \setminus \mathcal{A}_\mathcal{H}$ and $R'_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q'_1 \setminus \mathcal{A}_\mathcal{H}$. In the former subcase $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ is allowed to stay idle with $((R'_1 \parallel_L R_2)/\mathcal{A}_\mathcal{H}, (Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$, while in the latter subcase, since synchronization does not apply to τ, it follows that $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{Q}_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (Q'_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ with $((R_1 \parallel_L R_2)/\mathcal{A}_\mathcal{H}, (\bar{Q}_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((R'_1 \parallel_L R_2)/\mathcal{A}_\mathcal{H}, (Q'_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $(R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (R_1 \parallel_L R'_2) \,/\, \mathcal{A}_\mathcal{H}$ with $R_2 \stackrel{\tau}{\longrightarrow} R'_2$, then the proof is similar to the one of the previous case.
- If $(R_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (R'_1 \parallel_L R_2) \,/\, \mathcal{A}_\mathcal{H}$ with $R_1 \stackrel{h}{\longrightarrow} R'_1$ and $h \notin L$, then $R_1 \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} R'_1 \,/\, \mathcal{A}_\mathcal{H}$ as $h \in \mathcal{A}_\mathcal{H}$. From $R_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q_1 \setminus \mathcal{A}_\mathcal{H}$ it follows that either $R'_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q_1 \setminus \mathcal{A}_\mathcal{H}$, or there exist $\bar{Q}_1$ and Q'_1 such that $Q_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{Q}_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} Q'_1 \setminus \mathcal{A}_\mathcal{H}$ with $R_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{Q}_1 \setminus \mathcal{A}_\mathcal{H}$ and $R'_1 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q'_1 \setminus \mathcal{A}_\mathcal{H}$. In the former subcase $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ is allowed to stay idle with $((R'_1 \parallel_L R_2)/\mathcal{A}_\mathcal{H}, (Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$, while in the latter subcase, since synchronization does not apply to τ, it follows that $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{Q}_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (Q'_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ with $((R_1 \parallel_L R_2)/\mathcal{A}_\mathcal{H}, (\bar{Q}_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((R'_1 \parallel_L R_2)/\mathcal{A}_\mathcal{H}, (Q'_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.

- If $(R_1 \parallel_L R_2) / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} (R_1 \parallel_L R_2') / \mathcal{A}_\mathcal{H}$ with $R_2 \xrightarrow{h} R_2'$ and $h \notin L$, then the proof is similar to the one of the previous case.

(2) Starting from $(Q / \mathcal{A}_\mathcal{H}) \setminus L$ and $(R \setminus L) / \mathcal{A}_\mathcal{H}$ related by $\mathcal{B}$, so that $Q / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R \setminus \mathcal{A}_\mathcal{H}$, in the branching bisimulation game there are six cases based on the operational semantic rules in Table 1. In the first three cases, it is $(Q / \mathcal{A}_\mathcal{H}) \setminus L$ to move first:

- If $(Q / \mathcal{A}_\mathcal{H}) \setminus L \xrightarrow{l} (Q' / \mathcal{A}_\mathcal{H}) \setminus L$ with $Q \xrightarrow{l} Q'$ and $l \notin L$, then $Q / \mathcal{A}_\mathcal{H} \xrightarrow{l} Q' / \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$. From $Q / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R \setminus \mathcal{A}_\mathcal{H}$ it follows that there exist $\bar{R}$ and R' such that $R \setminus \mathcal{A}_\mathcal{H} \xRightarrow{\tau^*} \bar{R} \setminus \mathcal{A}_\mathcal{H} \xrightarrow{l} R' \setminus \mathcal{A}_\mathcal{H}$ with $Q / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{R} \setminus \mathcal{A}_\mathcal{H}$ and $Q' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R' \setminus \mathcal{A}_\mathcal{H}$. Since neither the restriction operator nor the hiding operator applies to τ and l, it follows that $(R \setminus L) / \mathcal{A}_\mathcal{H} \xRightarrow{\tau^*} (\bar{R} \setminus L) / \mathcal{A}_\mathcal{H} \xrightarrow{l} (R' \setminus L) / \mathcal{A}_\mathcal{H}$ with $((Q / \mathcal{A}_\mathcal{H}) \setminus L, (\bar{R} \setminus L) / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((Q' / \mathcal{A}_\mathcal{H}) \setminus L, (R' \setminus L) / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $(Q / \mathcal{A}_\mathcal{H}) \setminus L \xrightarrow{\tau} (Q' / \mathcal{A}_\mathcal{H}) \setminus L$ with $Q \xrightarrow{\tau} Q'$, then $Q / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} Q' / \mathcal{A}_\mathcal{H}$ as $\tau \notin \mathcal{A}_\mathcal{H}$. From $Q / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R \setminus \mathcal{A}_\mathcal{H}$ it follows that either $Q' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R \setminus \mathcal{A}_\mathcal{H}$, or there exist $\bar{R}$ and R' such that $R \setminus \mathcal{A}_\mathcal{H} \xRightarrow{\tau^*} \bar{R} \setminus \mathcal{A}_\mathcal{H} \xrightarrow{\tau} R' \setminus \mathcal{A}_\mathcal{H}$ with $Q / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{R} \setminus \mathcal{A}_\mathcal{H}$ and $Q' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R' \setminus \mathcal{A}_\mathcal{H}$. In the former subcase $(R \setminus L) / \mathcal{A}_\mathcal{H}$ is allowed to stay idle with $((Q' / \mathcal{A}_\mathcal{H}) \setminus L, (R \setminus L) / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$, while in the latter subcase, since neither the restriction operator nor the hiding operator applies to τ, it follows that $(R \setminus L) / \mathcal{A}_\mathcal{H} \xRightarrow{\tau^*} (\bar{R} \setminus L) / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} (R' \setminus L) / \mathcal{A}_\mathcal{H}$ with $((Q / \mathcal{A}_\mathcal{H}) \setminus L, (\bar{R} \setminus L) / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((Q' / \mathcal{A}_\mathcal{H}) \setminus L, (R' \setminus L) / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $(Q / \mathcal{A}_\mathcal{H}) \setminus L \xrightarrow{\tau} (Q' / \mathcal{A}_\mathcal{H}) \setminus L$ with $Q \xrightarrow{h} Q'$, then $Q / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} Q' / \mathcal{A}_\mathcal{H}$ as $h \in \mathcal{A}_\mathcal{H}$ and the rest of the proof is similar to the one of the previous case.

In the other three cases, instead, it is $(R \setminus L) / \mathcal{A}_\mathcal{H}$ to move first:

- If $(R \setminus L) / \mathcal{A}_\mathcal{H} \xrightarrow{l} (R' \setminus L) / \mathcal{A}_\mathcal{H}$ with $R \xrightarrow{l} R'$ and $l \notin L$, then $R \setminus \mathcal{A}_\mathcal{H} \xrightarrow{l} R' \setminus \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$. From $R \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q / \mathcal{A}_\mathcal{H}$ it follows that there exist $\bar{Q}$ and Q' such that $Q / \mathcal{A}_\mathcal{H} \xRightarrow{\tau^*} \bar{Q} / \mathcal{A}_\mathcal{H} \xrightarrow{l} Q' / \mathcal{A}_\mathcal{H}$ with $R \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{Q} / \mathcal{A}_\mathcal{H}$ and $R' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q' / \mathcal{A}_\mathcal{H}$. Since the restriction operator does not apply to τ and l, it follows that $(Q / \mathcal{A}_\mathcal{H}) \setminus L \xRightarrow{\tau^*} (\bar{Q} / \mathcal{A}_\mathcal{H}) \setminus L \xrightarrow{l} (Q' / \mathcal{A}_\mathcal{H}) \setminus L$ with $((R \setminus L) / \mathcal{A}_\mathcal{H}, (\bar{Q} / \mathcal{A}_\mathcal{H}) \setminus L) \in \mathcal{B}$ and $((R' \setminus L) / \mathcal{A}_\mathcal{H}, (Q' / \mathcal{A}_\mathcal{H}) \setminus L) \in \mathcal{B}$.
- If $(R \setminus L) / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} (R' \setminus L) / \mathcal{A}_\mathcal{H}$ with $R \xrightarrow{\tau} R'$, then $R \setminus \mathcal{A}_\mathcal{H} \xrightarrow{\tau} R' \setminus \mathcal{A}_\mathcal{H}$ as $\tau \notin \mathcal{A}_\mathcal{H}$. From $R \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q / \mathcal{A}_\mathcal{H}$ it follows that either $R' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q / \mathcal{A}_\mathcal{H}$, or there exist $\bar{Q}$ and Q' such that $Q / \mathcal{A}_\mathcal{H} \xRightarrow{\tau^*} \bar{Q} / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} Q' / \mathcal{A}_\mathcal{H}$ with $R \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{Q} / \mathcal{A}_\mathcal{H}$ and $R' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q' / \mathcal{A}_\mathcal{H}$. In the former subcase $(Q / \mathcal{A}_\mathcal{H}) \setminus L$ is allowed to stay idle with $((R' \setminus L) / \mathcal{A}_\mathcal{H}, (Q / \mathcal{A}_\mathcal{H}) \setminus L) \in \mathcal{B}$, while in the latter subcase, since the restriction operator does not apply to τ, it follows that $(Q / \mathcal{A}_\mathcal{H}) \setminus L \xRightarrow{\tau^*} (\bar{Q} / \mathcal{A}_\mathcal{H}) \setminus L \xrightarrow{\tau} (Q' / \mathcal{A}_\mathcal{H}) \setminus L$ with $((R \setminus L) / \mathcal{A}_\mathcal{H}, (\bar{Q} / \mathcal{A}_\mathcal{H}) \setminus L) \in \mathcal{B}$ and $((R' \setminus L) / \mathcal{A}_\mathcal{H}, (Q' / \mathcal{A}_\mathcal{H}) \setminus L) \in \mathcal{B}$.
- If $(R \setminus L) / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} (R' \setminus L) / \mathcal{A}_\mathcal{H}$ with $R \xrightarrow{h} R'$ and $h \notin L$, then $R / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} R' / \mathcal{A}_\mathcal{H}$ as $h \in \mathcal{A}_\mathcal{H}$ (note that $R \setminus \mathcal{A}_\mathcal{H}$ cannot perform h). From $R / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R \setminus \mathcal{A}_\mathcal{H}$ – as $P \in$ SBrSNNI and $R \in \mathit{reach}(P)$ – and $R \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q / \mathcal{A}_\mathcal{H}$ it follows that either $R' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q / \mathcal{A}_\mathcal{H}$ and hence $R' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q / \mathcal{A}_\mathcal{H}$ – as $R' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R' \setminus \mathcal{A}_\mathcal{H}$ due to $P \in$ SBrSNNI and $R' \in \mathit{reach}(P)$ – or there exist $\bar{Q}$ and Q' such that $Q / \mathcal{A}_\mathcal{H} \xRightarrow{\tau^*} \bar{Q} / \mathcal{A}_\mathcal{H} \xrightarrow{\tau} Q' / \mathcal{A}_\mathcal{H}$ with $R / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{Q} / \mathcal{A}_\mathcal{H}$ and $R' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q' / \mathcal{A}_\mathcal{H}$ and hence $R \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{Q} / \mathcal{A}_\mathcal{H}$ and $R' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} Q' / \mathcal{A}_\mathcal{H}$. In the former subcase $(Q / \mathcal{A}_\mathcal{H}) \setminus L$ is allowed to stay idle with $((R' \setminus L) / \mathcal{A}_\mathcal{H}, (Q / \mathcal{A}_\mathcal{H}) \setminus L) \in \mathcal{B}$, while in

the latter subcase, since the restriction operator does not apply to τ, it follows that $(Q \,/\, \mathcal{A}_\mathcal{H}) \setminus L \stackrel{\tau^*}{\Longrightarrow} (\bar{Q} \,/\, \mathcal{A}_\mathcal{H}) \setminus L \stackrel{\tau}{\longrightarrow} (Q' \,/\, \mathcal{A}_\mathcal{H}) \setminus L$ with $((R \setminus L) \,/\, \mathcal{A}_\mathcal{H}, (\bar{Q} \,/\, \mathcal{A}_\mathcal{H}) \setminus L) \in \mathcal{B}$ and $((R' \setminus L) \,/\, \mathcal{A}_\mathcal{H}, (Q' \,/\, \mathcal{A}_\mathcal{H}) \setminus L) \in \mathcal{B}$.

(3) Starting from $((Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (R_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$, so that $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R_1 \setminus \mathcal{A}_\mathcal{H}$ and $Q_2 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R_2 \setminus \mathcal{A}_\mathcal{H}$, in the branching bisimulation game there are five cases based on the operational semantic rules in Table 1:
- If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (Q_1' \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ with $Q_1 \stackrel{l}{\longrightarrow} Q_1'$ and $l \notin L$, then $Q_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} Q_1' \setminus \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$. From $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R_1 \setminus \mathcal{A}_\mathcal{H}$ it follows that there exist $\bar{R}_1$ and R_1' such that $R_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{R}_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} R_1' \setminus \mathcal{A}_\mathcal{H}$ with $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{R}_1 \setminus \mathcal{A}_\mathcal{H}$ and $Q_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R_1' \setminus \mathcal{A}_\mathcal{H}$. Since synchronization does not apply to τ, it follows that $(R_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{R}_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (R_1' \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}$ with $((Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (\bar{R}_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((Q_1' \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (R_1' \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (Q_1 \parallel_L Q_2') \setminus \mathcal{A}_\mathcal{H}$ with $Q_2 \stackrel{l}{\longrightarrow} Q_2'$ and $l \notin L$, then the proof is similar to the one of the previous case.
- If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (Q_1' \parallel_L Q_2') \setminus \mathcal{A}_\mathcal{H}$ with $Q_i \stackrel{l}{\longrightarrow} Q_i'$ for $i \in \{1,2\}$ and $l \in L$, then $Q_i \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} Q_i' \setminus \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$. From $Q_i \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R_i \setminus \mathcal{A}_\mathcal{H}$ it follows that there exist $\bar{R}_i$ and R_i' such that $R_i \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{R}_i \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} R_i' \setminus \mathcal{A}_\mathcal{H}$ with $Q_i \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{R}_i \setminus \mathcal{A}_\mathcal{H}$ and $Q_i' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R_i' \setminus \mathcal{A}_\mathcal{H}$. Since synchronization does not apply to τ, it follows that $(R_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{R}_1 \parallel_L \bar{R}_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} (R_1' \parallel_L R_2') \setminus \mathcal{A}_\mathcal{H}$ with $((Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (\bar{R}_1 \parallel_L \bar{R}_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((Q_1' \parallel_L Q_2') \setminus \mathcal{A}_\mathcal{H}, (R_1' \parallel_L R_2') \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (Q_1' \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}$ with $Q_1 \stackrel{\tau}{\longrightarrow} Q_1'$, then $Q_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} Q_1' \setminus \mathcal{A}_\mathcal{H}$ as $\tau \notin \mathcal{A}_\mathcal{H}$. From $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R_1 \setminus \mathcal{A}_\mathcal{H}$ it follows that either $Q_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R_1 \setminus \mathcal{A}_\mathcal{H}$, or there exist $\bar{R}_1$ and R_1' such that $R_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{R}_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} R_1' \setminus \mathcal{A}_\mathcal{H}$ with $Q_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{R}_1 \setminus \mathcal{A}_\mathcal{H}$ and $Q_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} R_1' \setminus \mathcal{A}_\mathcal{H}$. In the former subcase $(R_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}$ is allowed to stay idle with $((Q_1' \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (R_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$, while in the latter subcase, since synchronization does not apply to τ, it follows that $(R_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} (\bar{R}_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (R_1' \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}$ with $((Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (\bar{R}_1 \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $((Q_1' \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H}, (R_1' \parallel_L R_2) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $(Q_1 \parallel_L Q_2) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} (Q_1 \parallel_L Q_2') \setminus \mathcal{A}_\mathcal{H}$ with $Q_2 \stackrel{\tau}{\longrightarrow} Q_2'$, then the proof is similar to the one of the previous case. □

Theorem 4.5. *Let $P, P_1, P_2 \in \mathbb{P}$ and $\mathcal{P} \in \{\mathrm{SBrSNNI}, \mathrm{P\_BrNDC}, \mathrm{SBrNDC}\}$. Then:*

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

Proof. We first prove the four results for SBrSNNI, from which it will follow that they hold for P_BrNDC too by virtue of the forthcoming Theorem 4.8:

(1) Given an arbitrary $P \in \mathrm{SBrSNNI}$ and an arbitrary $a \in \mathcal{A}_\mathcal{L} \cup \{\tau\}$, from $P \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P \,/\, \mathcal{A}_\mathcal{H}$ we derive that $a \,.\, (P \setminus \mathcal{A}_\mathcal{H}) \approx_\mathrm{b} a \,.\, (P \,/\, \mathcal{A}_\mathcal{H})$ because $\approx_\mathrm{b}$ is a congruence with respect to action prefix [GW96], from which it follows that $(a \,.\, P) \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} (a \,.\, P) \,/\, \mathcal{A}_\mathcal{H}$, i.e., $a \,.\, P \in \mathrm{BrSNNI}$, because $a \notin \mathcal{A}_\mathcal{H}$. To conclude the proof, it suffices to observe that all

the processes reachable from $a\,.\,P$ after performing a are processes reachable from P, which are known to be BrSNNI.

(2) Given two arbitrary $P_1, P_2 \in$ SBrSNNI and an arbitrary $L \subseteq \mathcal{A}_{\mathcal{L}}$, the result follows from Lemma 4.4(1) by taking Q_1 identical to R_1 and Q_2 identical to R_2.

(3) Given an arbitrary $P \in$ SBrSNNI and an arbitrary $L \subseteq \mathcal{A}$, the result follows from Lemma 4.4(2) by taking Q identical to R – which will be denoted by P' – because:
- $(P' \setminus L) \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} (P' \setminus \mathcal{A}_{\mathcal{H}}) \setminus L$ as the order in which restriction sets are considered is unimportant.
- $(P' \setminus \mathcal{A}_{\mathcal{H}}) \setminus L \approx_{\mathrm{b}} (P' \,/\, \mathcal{A}_{\mathcal{H}}) \setminus L$ due to $P' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P' \,/\, \mathcal{A}_{\mathcal{H}}$ – as $P \in$ SBrSNNI and $P' \in \mathit{reach}(P)$ – and $\approx_{\mathrm{b}}$ being a congruence with respect to the restriction operator due to Lemma 4.2.
- $(P' \,/\, \mathcal{A}_{\mathcal{H}}) \setminus L \approx_{\mathrm{b}} (P' \setminus L) \,/\, \mathcal{A}_{\mathcal{H}}$ as shown in Lemma 4.4(2).
- From the transitivity of $\approx_{\mathrm{b}}$ we obtain that $(P' \setminus L) \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} (P' \setminus L) \,/\, \mathcal{A}_{\mathcal{H}}$.

(4) Given an arbitrary $P \in$ SBrSNNI and an arbitrary $L \subseteq \mathcal{A}_{\mathcal{L}}$, for every $P' \in \mathit{reach}(P)$ it holds that $P' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P' \,/\, \mathcal{A}_{\mathcal{H}}$, from which we derive that $(P' \setminus \mathcal{A}_{\mathcal{H}}) \,/\, L \approx_{\mathrm{b}} (P' / \mathcal{A}_{\mathcal{H}}) \,/\, L$ because $\approx_{\mathrm{b}}$ is a congruence with respect to the hiding operator due to Lemma 4.2. Since $L \cap \mathcal{A}_{\mathcal{H}} = \emptyset$, we have that $(P' \setminus \mathcal{A}_{\mathcal{H}}) \,/\, L$ is isomorphic to $(P' \,/\, L) \setminus \mathcal{A}_{\mathcal{H}}$ and $(P' \,/\, \mathcal{A}_{\mathcal{H}}) \,/\, L$ is isomorphic to $(P' \,/\, L) \,/\, \mathcal{A}_{\mathcal{H}}$, hence $(P' \,/\, L) \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} (P' \,/\, L) \,/\, \mathcal{A}_{\mathcal{H}}$, i.e., $P' \,/\, L$ is BrSNNI.

We then prove the four results for SBrNDC:

(1) Given an arbitrary $P \in$ SBrNDC and an arbitrary $a \in \mathcal{A}_\tau \setminus \mathcal{A}_{\mathcal{H}}$, it trivially holds that $a\,.\,P \in$ SBrNDC because a is not high and all the processes reachable from $a\,.\,P$ after performing a are processes reachable from P, which is known to be SBrNDC.

(2) Given two arbitrary $P_1, P_2 \in$ SBrNDC and an arbitrary $L \subseteq \mathcal{A}$, the result follows from Lemma 4.4(3) as can be seen by observing that whenever $P_1' \parallel_L P_2' \xrightarrow{h} P_1'' \parallel_L P_2''$ for $P_1' \parallel_L P_2' \in \mathit{reach}(P_1 \parallel_L P_2)$:
- If $P_1' \xrightarrow{h} P_1''$, $P_2'' = P_2'$, and $h \notin L$, then from $P_1 \in$ SBrNDC it follows that $P_1' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P_1'' \setminus \mathcal{A}_{\mathcal{H}}$ and hence $(P_1' \parallel_L P_2') \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} (P_1'' \parallel_L P_2'') \setminus \mathcal{A}_{\mathcal{H}}$ as $P_2' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P_2'' \setminus \mathcal{A}_{\mathcal{H}}$.
- If $P_2' \xrightarrow{h} P_2''$, $P_1'' = P_1'$, and $h \notin L$, then from $P_2 \in$ SBrNDC it follows that $P_2' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P_2'' \setminus \mathcal{A}_{\mathcal{H}}$ and hence $(P_1' \parallel_L P_2') \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} (P_1'' \parallel_L P_2'') \setminus \mathcal{A}_{\mathcal{H}}$ as $P_1' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P_1'' \setminus \mathcal{A}_{\mathcal{H}}$.
- If $P_1' \xrightarrow{h} P_1''$, $P_2' \xrightarrow{h} P_2''$, and $h \in L$, then from $P_1, P_2 \in$ SBrNDC it follows that $P_1' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P_1'' \setminus \mathcal{A}_{\mathcal{H}}$ and $P_2' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P_2'' \setminus \mathcal{A}_{\mathcal{H}}$, which in turn entail that $(P_1' \parallel_L P_2') \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} (P_1'' \parallel_L P_2'') \setminus \mathcal{A}_{\mathcal{H}}$.

(3) Given an arbitrary $P \in$ SBrNDC and an arbitrary $L \subseteq \mathcal{A}$, for every $P' \in \mathit{reach}(P)$ and for every P'' such that $P' \xrightarrow{h} P''$ it holds that $P' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P'' \setminus \mathcal{A}_{\mathcal{H}}$, from which we derive that $(P' \setminus \mathcal{A}_{\mathcal{H}}) \setminus L \approx_{\mathrm{b}} (P'' \setminus \mathcal{A}_{\mathcal{H}}) \setminus L$ because $\approx_{\mathrm{b}}$ is a congruence with respect to the restriction operator due to Lemma 4.2. Since $(P' \setminus \mathcal{A}_{\mathcal{H}}) \setminus L$ is isomorphic to $(P' \setminus L) \setminus \mathcal{A}_{\mathcal{H}}$ and $(P'' \setminus \mathcal{A}_{\mathcal{H}}) \setminus L$ is isomorphic to $(P'' \setminus L) \setminus \mathcal{A}_{\mathcal{H}}$, we have that $(P' \setminus L) \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} (P'' \setminus L) \setminus \mathcal{A}_{\mathcal{H}}$.

(4) Given an arbitrary $P \in$ SBrNDC and an arbitrary $L \subseteq \mathcal{A}_{\mathcal{L}}$, for every $P' \in \mathit{reach}(P)$ and for every P'' such that $P' \xrightarrow{h} P''$ it holds that $P' \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} P'' \setminus \mathcal{A}_{\mathcal{H}}$, from which we derive that $(P' \setminus \mathcal{A}_{\mathcal{H}}) \,/\, L \approx_{\mathrm{b}} (P'' \setminus \mathcal{A}_{\mathcal{H}}) \,/\, L$ because $\approx_{\mathrm{b}}$ is a congruence with respect to the hiding operator due to Lemma 4.2. Since $L \cap \mathcal{A}_{\mathcal{H}} = \emptyset$, we have that $(P' \setminus \mathcal{A}_{\mathcal{H}}) \,/\, L$ is isomorphic to $(P' \,/\, L) \setminus \mathcal{A}_{\mathcal{H}}$ and $(P'' \setminus \mathcal{A}_{\mathcal{H}}) \,/\, L$ is isomorphic to $(P'' \,/\, L) \setminus \mathcal{A}_{\mathcal{H}}$, hence $(P' \,/\, L) \setminus \mathcal{A}_{\mathcal{H}} \approx_{\mathrm{b}} (P'' \,/\, L) \setminus \mathcal{A}_{\mathcal{H}}$. □

4.2. **Taxonomy of Security Properties.** The relationships among the various $\approx_{\rm b}$-based noninterference properties turn out to follow the same pattern as Theorem 2.6.

In [EAB23] some parts of the proof of the forthcoming Theorem 4.8 – as well as some parts of the proof of the previous Theorem 4.5 – proceeded by induction on the depth of the labeled transition system underlying the process under examination. Now that the language includes recursion, which may introduce cycles, we have to follow a different proof technique, which relies on the notion of branching bisimulation up to $\approx_{\rm b}$ of [Gla93] recalled below.

Definition 4.6. A symmetric binary relation $\mathcal{B}$ over $\mathbb{P}$ is a *branching bisimulation up to* $\approx_{\rm b}$ iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:

- for each $P_1 \stackrel{\tau^*}{\Longrightarrow} \bar{P}_1 \stackrel{a}{\longrightarrow} P_1'$ with $P_1 \approx_{\rm b} \bar{P}_1$:
  - either $a = \tau$ and $\bar{P}_1 \approx_{\rm b} P_1'$;
  - or there exists $P_2 \stackrel{\tau^*}{\Longrightarrow} \bar{P}_2 \stackrel{a}{\longrightarrow} P_2'$ such that $\bar{P}_1 \approx_{\rm b} \mathcal{B} \approx_{\rm b} \bar{P}_2$ and $P_1' \approx_{\rm b} \mathcal{B} \approx_{\rm b} P_2'$. ■

In the definition above, $\approx_{\rm b} \mathcal{B} \approx_{\rm b}$ stands for the composition of the three mentioned relations. Moreover, in the case that $a = \tau$ and $\bar{P}_1 \approx_{\rm b} P_1'$, since the considered relations are symmetric and $\approx_{\rm b}$ is also transitive and reflexive, it holds that $P_1' \approx_{\rm b} \bar{P}_1 \approx_{\rm b} P_1 \; \mathcal{B} \; P_2 \approx_{\rm b} P_2$, i.e., $P_1' \approx_{\rm b} \mathcal{B} \approx_{\rm b} P_2$. As shown in [Gla93], if $\mathcal{B}$ is a branching bisimulation up to $\approx_{\rm b}$ and $(P_1, P_2) \in \mathcal{B}$, then $P_1 \approx_{\rm b} P_2$ because $\approx_{\rm b} \mathcal{B} \approx_{\rm b}$ turns out to be a branching bisimulation; the advantage of working with $\mathcal{B}$ is that it needs to include fewer pairs. While in [FG01] weak bisimulation up to $\approx$ [SM92] has been exploited several times to prove various results, here branching bisimulation up to $\approx_{\rm b}$ is employed only to show that SBrNDC $\subset$ SBrSNNI.

To study the taxonomy of the noninterference properties in Definition 4.1, we first prove some further ancillary results about parallel composition, restriction, and hiding under SBrSNNI and SBrNDC.

Lemma 4.7. *Let $P, P_1, P_2 \in \mathbb{P}$. Then:*

(1) *If $P \in$ SBrNDC and $P' \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} P'' \,/\, \mathcal{A}_\mathcal{H}$ for $P' \in reach(P)$, then $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}'' \setminus \mathcal{A}_\mathcal{H}$ with $P'' \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} \hat{P}'' \setminus \mathcal{A}_\mathcal{H}$.*
(2) *If $P_1, P_2 \in$ SBrNDC and $P_1 \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} P_2 \setminus \mathcal{A}_\mathcal{H}$, then $P_1 \,/\, \mathcal{A}_\mathcal{H} \approx_{\rm b} P_2 \,/\, \mathcal{A}_\mathcal{H}$.*
(3) *If $P_2 \in$ SBrSNNI and $L \subseteq \mathcal{A}_\mathcal{H}$, then $P_1' \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} ((P_2' \parallel_L Q) \,/\, L) \setminus \mathcal{A}_\mathcal{H}$ for all $Q \in \mathbb{P}$ having only actions in $\mathcal{A}_\mathcal{H}$ and for all $P_1' \in reach(P_1)$ and $P_2' \in reach(P_2)$ such that $P_1' \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} P_2' \,/\, \mathcal{A}_\mathcal{H}$.*

Proof. Let $\mathcal{B}$ be a symmetric relation containing all the pairs of processes that have to be shown to be branching bisimilar according to the property considered between the last two stated above:

(1) We proceed by induction on the number $n \in \mathbb{N}$ of τ-transitions in $P' \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} P'' \,/\, \mathcal{A}_\mathcal{H}$:
   - If $n = 0$ then $P' \,/\, \mathcal{A}_\mathcal{H}$ stays idle and $P'' \,/\, \mathcal{A}_\mathcal{H}$ is $P' \,/\, \mathcal{A}_\mathcal{H}$. Likewise, $P' \setminus \mathcal{A}_\mathcal{H}$ can stay idle, i.e., $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} P' \setminus \mathcal{A}_\mathcal{H}$, with $P' \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} P' \setminus \mathcal{A}_\mathcal{H}$ as $\approx_{\rm b}$ is reflexive.
   - Let $n > 0$ and $P_0' \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_1' \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} \ldots \stackrel{\tau}{\longrightarrow} P_{n-1}' \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_n' \,/\, \mathcal{A}_\mathcal{H}$ where $P_0' \,/\, \mathcal{A}_\mathcal{H}$ is $P' \,/\, \mathcal{A}_\mathcal{H}$ and $P_n' \,/\, \mathcal{A}_\mathcal{H}$ is $P'' \,/\, \mathcal{A}_\mathcal{H}$. From the induction hypothesis it follows that $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}_{n-1}' \setminus \mathcal{A}_\mathcal{H}$ with $P_{n-1}' \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} \hat{P}_{n-1}' \setminus \mathcal{A}_\mathcal{H}$. As far as the n-th τ-transition $P_{n-1}' \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_n' \,/\, \mathcal{A}_\mathcal{H}$ is concerned, there are two cases depending on whether it is originated from $P_{n-1}' \stackrel{\tau}{\longrightarrow} P_n'$ or $P_{n-1}' \stackrel{h}{\longrightarrow} P_n'$:
     - If $P_{n-1}' \stackrel{\tau}{\longrightarrow} P_n'$ then $P_{n-1}' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_n' \setminus \mathcal{A}_\mathcal{H}$. Since $P_{n-1}' \setminus \mathcal{A}_\mathcal{H} \approx_{\rm b} \hat{P}_{n-1}' \setminus \mathcal{A}_\mathcal{H}$, it follows that:

* Either $P'_n \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}'_{n-1} \setminus \mathcal{A}_\mathcal{H}$, in which case $\hat{P}'_{n-1} \setminus \mathcal{A}_\mathcal{H}$ stays idle and we are done because $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}'_{n-1} \setminus \mathcal{A}_\mathcal{H}$.
* Or $\hat{P}'_{n-1} \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}''_{n-1} \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} \hat{P}'_n \setminus \mathcal{A}_\mathcal{H}$ with $P'_{n-1} \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}''_{n-1} \setminus \mathcal{A}_\mathcal{H}$ and $P'_n \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}'_n \setminus \mathcal{A}_\mathcal{H}$, in which case we are done because $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}'_n \setminus \mathcal{A}_\mathcal{H}$.

– If $P'_{n-1} \stackrel{h}{\longrightarrow} P'_n$ then from $P \in \mathrm{SBrNDC}$ it follows that $P'_{n-1} \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_n \setminus \mathcal{A}_\mathcal{H}$. Since $P'_{n-1} \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}'_{n-1} \setminus \mathcal{A}_\mathcal{H}$ and $\approx_\mathrm{b}$ is symmetric and transitive, we obtain $P'_n \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}'_{n-1} \setminus \mathcal{A}_\mathcal{H}$. Thus we are done because $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}'_{n-1} \setminus \mathcal{A}_\mathcal{H}$.

(2) Starting from $(P_1 \,/\, \mathcal{A}_\mathcal{H}, P_2 \,/\, \mathcal{A}_\mathcal{H}) \in \mathcal{B}$, so that $P_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2 \setminus \mathcal{A}_\mathcal{H}$, in the branching bisimulation game there are three cases based on the operational semantic rules in Table 1:

- If $P_1 \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P'_1 \,/\, \mathcal{A}_\mathcal{H}$ with $P_1 \stackrel{h}{\longrightarrow} P'_1$, then $P_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_1 \setminus \mathcal{A}_\mathcal{H}$ as $h \in \mathcal{A}_\mathcal{H}$ and $P_1 \in \mathrm{SBrNDC}$. Since $P_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2 \setminus \mathcal{A}_\mathcal{H}$, so that $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2 \setminus \mathcal{A}_\mathcal{H}$ as $\approx_\mathrm{b}$ is symmetric and transitive, and $P'_1, P_2 \in \mathrm{SBrNDC}$, it follows that $P_2 \,/\, \mathcal{A}_\mathcal{H}$ is allowed to stay idle with $(P'_1 \,/\, \mathcal{A}_\mathcal{H}, P_2 \,/\, \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $P_1 \,/\, \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} P'_1 \,/\, \mathcal{A}_\mathcal{H}$ with $P_1 \stackrel{l}{\longrightarrow} P'_1$, then $P_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} P'_1 \setminus \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$. From $P_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2 \setminus \mathcal{A}_\mathcal{H}$ it follows that there exist $\bar{P}_2$ and P'_2 such that $P_2 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}_2 \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} P'_2 \setminus \mathcal{A}_\mathcal{H}$ with $P_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}_2 \setminus \mathcal{A}_\mathcal{H}$ and $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \setminus \mathcal{A}_\mathcal{H}$. Thus $P_2 \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}_2 \,/\, \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} P'_2 \,/\, \mathcal{A}_\mathcal{H}$. Since $P_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}_2 \setminus \mathcal{A}_\mathcal{H}$ with $P_1, \bar{P}_2 \in \mathrm{SBrNDC}$ and $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \setminus \mathcal{A}_\mathcal{H}$ with $P'_1, P'_2 \in \mathrm{SBrNDC}$, we have $(P_1 \,/\, \mathcal{A}_\mathcal{H}, \bar{P}_2 \,/\, \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $(P'_1 \,/\, \mathcal{A}_\mathcal{H}, P'_2 \,/\, \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $P_1 \,/\, \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P'_1 \,/\, \mathcal{A}_\mathcal{H}$ with $P_1 \stackrel{\tau}{\longrightarrow} P'_1$, then the proof is similar to the previous one, with the additional possibility that, in response to $P_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P'_1 \setminus \mathcal{A}_\mathcal{H}$, $P_2 \setminus \mathcal{A}_\mathcal{H}$ stays idle with $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2 \setminus \mathcal{A}_\mathcal{H}$, so that $P_2 \,/\, \mathcal{A}_\mathcal{H}$ stays idle too with $(P'_1 \,/\, \mathcal{A}_\mathcal{H}, P_2 \,/\, \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ because $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2 \setminus \mathcal{A}_\mathcal{H}$ and $P'_1, P_2 \in \mathrm{SBrNDC}$.

(3) Starting from $P'_1 \setminus \mathcal{A}_\mathcal{H}$ and $((P'_2 \parallel_L Q) \,/\, L) \setminus \mathcal{A}_\mathcal{H}$ related by $\mathcal{B}$, so that $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \,/\, \mathcal{A}_\mathcal{H}$, in the branching bisimulation game there are six cases based on the operational semantic rules in Table 1. In the first two cases, it is $P'_1 \setminus \mathcal{A}_\mathcal{H}$ to move first:

- If $P'_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} P''_1 \setminus \mathcal{A}_\mathcal{H}$ we observe that from $P'_2 \in \mathit{reach}(P_2)$ and $P_2 \in \mathrm{SBrSNNI}$ it follows that $P'_2 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \,/\, \mathcal{A}_\mathcal{H}$, so that $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \setminus \mathcal{A}_\mathcal{H}$, i.e., $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \setminus \mathcal{A}_\mathcal{H}$, as $\approx_\mathrm{b}$ is symmetric and transitive. As a consequence, since $l \neq \tau$ there exist $\bar{P}'_2$ and P''_2 such that $P'_2 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}'_2 \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} P''_2 \setminus \mathcal{A}_\mathcal{H}$ with $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}'_2 \setminus \mathcal{A}_\mathcal{H}$ and $P''_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P''_2 \setminus \mathcal{A}_\mathcal{H}$. Thus, $((P'_2 \parallel_L Q) \,/\, L) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} ((\bar{P}'_2 \parallel_L Q) \,/\, L) \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} ((P''_2 \parallel_L Q) \,/\, L) \setminus \mathcal{A}_\mathcal{H}$ with $(P'_1 \setminus \mathcal{A}_\mathcal{H}, ((\bar{P}'_2 \parallel_L Q) \,/\, L) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $P'_1 \in \mathit{reach}(P_1)$, $\bar{P}'_2 \in \mathit{reach}(P_2)$, and $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}'_2 \,/\, \mathcal{A}_\mathcal{H}$ as $P_2 \in \mathrm{SBrSNNI}$ – and $(P''_1 \setminus \mathcal{A}_\mathcal{H}, ((P''_2 \parallel_L Q) \,/\, L) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $P''_1 \in \mathit{reach}(P_1)$, $P''_2 \in \mathit{reach}(P_2)$, and $P''_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P''_2 \,/\, \mathcal{A}_\mathcal{H}$ as $P_2 \in \mathrm{SBrSNNI}$.
- If $P'_1 \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P''_1 \setminus \mathcal{A}_\mathcal{H}$ there are two subcases:
  – If $P''_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \,/\, \mathcal{A}_\mathcal{H}$ then $(P'_2 \parallel_L Q) \,/\, L) \setminus \mathcal{A}_\mathcal{H}$ is allowed to stay idle with $(P''_1 \setminus \mathcal{A}_\mathcal{H}, ((P'_2 \parallel_L Q) \,/\, L) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ because $P''_1 \in \mathit{reach}(P_1)$ and $P'_2 \in \mathit{reach}(P_2)$.
  – If $P''_1 \setminus \mathcal{A}_\mathcal{H} \not\approx_\mathrm{b} P'_2 \,/\, \mathcal{A}_\mathcal{H}$ we observe that from $P'_2 \in \mathit{reach}(P_2)$ and $P_2 \in \mathrm{SBrSNNI}$ it follows that $P'_2 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \,/\, \mathcal{A}_\mathcal{H}$, so that on the one hand $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \setminus \mathcal{A}_\mathcal{H}$, i.e., $P'_1 \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \setminus \mathcal{A}_\mathcal{H}$, while on the other hand $P''_1 \setminus \mathcal{A}_\mathcal{H} \not\approx_\mathrm{b} P'_2 \,/\, \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'_2 \setminus \mathcal{A}_\mathcal{H}$, i.e., $P''_1 \setminus \mathcal{A}_\mathcal{H} \not\approx_\mathrm{b} P'_2 \setminus \mathcal{A}_\mathcal{H}$. As a consequence, there exist $\bar{P}'_2$ and P''_2 such that

$P_2' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}_2' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_2'' \setminus \mathcal{A}_\mathcal{H}$ with $P_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}_2' \setminus \mathcal{A}_\mathcal{H}$ and $P_1'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2'' \setminus \mathcal{A}_\mathcal{H}$. Therefore, $((P_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} ((\bar{P}_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} ((P_2'' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}$ with $(P_1' \setminus \mathcal{A}_\mathcal{H}, ((\bar{P}_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $P_1' \in \mathit{reach}(P_1)$, $\bar{P}_2' \in \mathit{reach}(P_2)$, and $P_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}_2' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI – and $(P_1'' \setminus \mathcal{A}_\mathcal{H}, ((P_2'' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $P_1'' \in \mathit{reach}(P_1)$, $P_2'' \in \mathit{reach}(P_2)$, and $P_1'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2'' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI.

In the other four cases, instead, it is $((P_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}$ to move first:

- If $((P_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} ((P_2'' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}$ with $P_2' \stackrel{l}{\longrightarrow} P_2''$ so that $P_2' \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} P_2'' \setminus \mathcal{A}_\mathcal{H}$ as $l \notin \mathcal{A}_\mathcal{H}$, we observe that from $P_2' \in \mathit{reach}(P_2)$ and $P_2 \in$ SBrSNNI it follows that $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2' / \mathcal{A}_\mathcal{H}$, so that $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$, i.e., $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$. Consequently, since $l \neq \tau$ there exist $\bar{P}_1'$ and P_1'' such that $P_1' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}_1' \setminus \mathcal{A}_\mathcal{H} \stackrel{l}{\longrightarrow} P_1'' \setminus \mathcal{A}_\mathcal{H}$ with $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}_1' \setminus \mathcal{A}_\mathcal{H}$ and $P_2'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1'' \setminus \mathcal{A}_\mathcal{H}$. Therefore, $(((P_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}, \bar{P}_1' \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $\bar{P}_1' \in \mathit{reach}(P_1)$, $P_2' \in \mathit{reach}(P_2)$, and $\bar{P}_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI – and $(((P_2'' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}, P_1'' \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $P_1'' \in \mathit{reach}(P_1)$, $P_2'' \in \mathit{reach}(P_2)$, and $P_1'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2'' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI.
- If $((P_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} ((P_2'' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}$ with $P_2' \stackrel{\tau}{\longrightarrow} P_2''$ so that $P_2' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_2'' \setminus \mathcal{A}_\mathcal{H}$ as $\tau \notin \mathcal{A}_\mathcal{H}$, we observe that from $P_2' \in \mathit{reach}(P_2)$ and $P_2 \in$ SBrSNNI it follows that $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2' / \mathcal{A}_\mathcal{H}$, so that $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$, i.e., $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$. There are two subcases:
  - If $P_2'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$ then $P_1' \setminus \mathcal{A}_\mathcal{H}$ is allowed to stay idle with $(((P_2'' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}, P_1' \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ because $P_1' \in \mathit{reach}(P_1)$, $P_2'' \in \mathit{reach}(P_2)$, and $P_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2'' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI.
  - If $P_2'' \setminus \mathcal{A}_\mathcal{H} \not\approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$ then there exist $\bar{P}_1'$ and P_1'' such that $P_1' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}_1' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_1'' \setminus \mathcal{A}_\mathcal{H}$ with $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}_1' \setminus \mathcal{A}_\mathcal{H}$ and $P_2'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1'' \setminus \mathcal{A}_\mathcal{H}$. Therefore, $(((P_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}, \bar{P}_1' \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $\bar{P}_1' \in \mathit{reach}(P_1)$, $P_2' \in \mathit{reach}(P_2)$, and $\bar{P}_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI – and $(((P_2'' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}, P_1'' \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $P_1'' \in \mathit{reach}(P_1)$, $P_2'' \in \mathit{reach}(P_2)$, and $P_1'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2'' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI.
- If $((P_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} ((P_2' \parallel_L Q') / L) \setminus \mathcal{A}_\mathcal{H}$ with $Q \stackrel{\tau}{\longrightarrow} Q'$, then trivially $(((P_2' \parallel_L Q') / L) \setminus \mathcal{A}_\mathcal{H}, P_1' \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$.
- If $((P_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} ((P_2'' \parallel_L Q' / L) \setminus \mathcal{A}_\mathcal{H})$ with $P_2' \stackrel{h}{\longrightarrow} P_2''$ – so that $P_2' / \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_2'' / \mathcal{A}_\mathcal{H}$ as $h \in \mathcal{A}_\mathcal{H}$ – and $Q \stackrel{h}{\longrightarrow} Q'$ for $h \in L$, we observe that from $P_2', P_2'' \in \mathit{reach}(P_2)$ and $P_2 \in$ SBrSNNI it follows that $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2' / \mathcal{A}_\mathcal{H}$ and $P_2'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2'' / \mathcal{A}_\mathcal{H}$, so that $P_2' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_2'' \setminus \mathcal{A}_\mathcal{H}$ and $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$, i.e., $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$. There are two subcases:
  - If $P_2'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$ then $P_1' \setminus \mathcal{A}_\mathcal{H}$ is allowed to stay idle with $(((P_2'' \parallel_L Q') / L) \setminus \mathcal{A}_\mathcal{H}, P_1' \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ because $P_1' \in \mathit{reach}(P_1)$, $P_2'' \in \mathit{reach}(P_2)$, and $P_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2'' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI.
  - If $P_2'' \setminus \mathcal{A}_\mathcal{H} \not\approx_\mathrm{b} P_1' \setminus \mathcal{A}_\mathcal{H}$ then there exist $\bar{P}_1'$ and P_1'' such that $P_1' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}_1' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P_1'' \setminus \mathcal{A}_\mathcal{H}$ with $P_2' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}_1' \setminus \mathcal{A}_\mathcal{H}$ and $P_2'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_1'' \setminus \mathcal{A}_\mathcal{H}$. Therefore, $(((P_2' \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}, \bar{P}_1' \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $\bar{P}_1' \in \mathit{reach}(P_1)$, $P_2' \in \mathit{reach}(P_2)$, and $\bar{P}_1' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI – and $(((P_2'' \parallel_L Q') / L) \setminus \mathcal{A}_\mathcal{H}, P_1'' \setminus \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ – because $P_1'' \in \mathit{reach}(P_1)$, $P_2'' \in \mathit{reach}(P_2)$, and $P_1'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P_2'' / \mathcal{A}_\mathcal{H}$ as $P_2 \in$ SBrSNNI. □

Theorem 4.8. SBrNDC $\subset$ SBrSNNI = P_BrNDC $\subset$ BrNDC $\subset$ BrSNNI.

Proof. Let us examine each relationship separately:

- SBrNDC $\subset$ SBrSNNI. Given $P \in$ SBrNDC, the result follows by proving that the symmetric relation $\mathcal{B} = \{(P' \setminus \mathcal{A}_\mathcal{H}, P' / \mathcal{A}_\mathcal{H}), (P' / \mathcal{A}_\mathcal{H}, P' \setminus \mathcal{A}_\mathcal{H}) \mid P' \in \mathit{reach}(P)\}$ is a branching bisimulation up to $\approx_\mathrm{b}$. Starting from $P' \setminus \mathcal{A}_\mathcal{H}$ and $P' / \mathcal{A}_\mathcal{H}$ related by $\mathcal{B}$, in the up-to branching bisimulation game there are three cases based on the operational semantic rules in Table 1. In the first case, it is $P' \setminus \mathcal{A}_\mathcal{H}$ to move first:
  - If $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}' \setminus \mathcal{A}_\mathcal{H} \stackrel{a}{\longrightarrow} P'' \setminus \mathcal{A}_\mathcal{H}$ with $\bar{P}' \stackrel{a}{\longrightarrow} P''$ and $a \in \mathcal{A}_\mathcal{L} \cup \{\tau\}$, then $P'/\mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}'/\mathcal{A}_\mathcal{H} \stackrel{a}{\longrightarrow} P''/\mathcal{A}_\mathcal{H}$ as $\tau, a \notin \mathcal{A}_\mathcal{H}$, with $(\bar{P}' \setminus \mathcal{A}_\mathcal{H}, \bar{P}' / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ and $(P'' \setminus \mathcal{A}_\mathcal{H}, P'' / \mathcal{A}_\mathcal{H}) \in \mathcal{B}$ so that $\bar{P}' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}' \setminus \mathcal{A}_\mathcal{H}$ $\mathcal{B}$ $\bar{P}' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}' / \mathcal{A}_\mathcal{H}$ and $P'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'' \setminus \mathcal{A}_\mathcal{H}$ $\mathcal{B}$ $P'' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'' / \mathcal{A}_\mathcal{H}$.

  In the other two cases, instead, it is $P' / \mathcal{A}_\mathcal{H}$ to move first (note that possible τ-transitions arising from high actions in P' can no longer be executed when starting from $P' \setminus \mathcal{A}_\mathcal{H}$):
  - If $P' / \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}' / \mathcal{A}_\mathcal{H} \stackrel{a}{\longrightarrow} P'' / \mathcal{A}_\mathcal{H}$ with $\bar{P}' \stackrel{a}{\longrightarrow} P''$ and $a \in \mathcal{A}_\mathcal{L} \cup \{\tau\}$, then $\bar{P}' \setminus \mathcal{A}_\mathcal{H} \stackrel{a}{\longrightarrow} P'' \setminus \mathcal{A}_\mathcal{H}$ as $a \notin \mathcal{A}_\mathcal{H}$. Since $P' / \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}' / \mathcal{A}_\mathcal{H}$ implies $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}' \setminus \mathcal{A}_\mathcal{H}$ with $\bar{P}' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}' \setminus \mathcal{A}_\mathcal{H}$ by virtue of Lemma 4.7(1), from $\bar{P}' \setminus \mathcal{A}_\mathcal{H} \stackrel{a}{\longrightarrow} P'' \setminus \mathcal{A}_\mathcal{H}$ it follows that:
    - Either $a = \tau$ and $P'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}' \setminus \mathcal{A}_\mathcal{H}$, hence $\bar{P}' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'' \setminus \mathcal{A}_\mathcal{H}$ as $\approx_\mathrm{b}$ is symmetric and transitive. From Lemma 4.7(2) it then follows that $\bar{P}' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'' / \mathcal{A}_\mathcal{H}$ because $\bar{P}', P'' \in$ SBrNDC as $\bar{P}', P'' \in \mathit{reach}(P)$ and $P \in$ SBrNDC. Therefore $P' \setminus \mathcal{A}_\mathcal{H}$ can stay idle in the up-to branching bisimulation game.
    - Or $\hat{P}' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}'' \setminus \mathcal{A}_\mathcal{H} \stackrel{a}{\longrightarrow} \hat{P}''' \setminus \mathcal{A}_\mathcal{H}$ with $\bar{P}' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}'' \setminus \mathcal{A}_\mathcal{H}$ and $P'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}''' \setminus \mathcal{A}_\mathcal{H}$. From $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}' \setminus \mathcal{A}_\mathcal{H}$ it follows that $P' \setminus \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \hat{P}'' \setminus \mathcal{A}_\mathcal{H} \stackrel{a}{\longrightarrow} \hat{P}''' \setminus \mathcal{A}_\mathcal{H}$ with $\bar{P}' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \bar{P}' / \mathcal{A}_\mathcal{H}$ $\mathcal{B}$ $\bar{P}' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}'' \setminus \mathcal{A}_\mathcal{H}$ and $P'' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'' / \mathcal{A}_\mathcal{H}$ $\mathcal{B}$ $P'' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} \hat{P}''' \setminus \mathcal{A}_\mathcal{H}$.
  - If $P' / \mathcal{A}_\mathcal{H} \stackrel{\tau^*}{\Longrightarrow} \bar{P}' / \mathcal{A}_\mathcal{H} \stackrel{\tau}{\longrightarrow} P'' / \mathcal{A}_\mathcal{H}$ with $\bar{P}' \stackrel{h}{\longrightarrow} P''$, we observe that $\bar{P}' \setminus \mathcal{A}_\mathcal{H}$ cannot perform any h-action as $h \in \mathcal{A}_\mathcal{H}$, nor we know whether it can perform a τ-action – moreover $(P'' / \mathcal{A}_\mathcal{H}, P' \setminus \mathcal{A}_\mathcal{H}) \notin \mathcal{B}$ when P'' is different from P', hence the need of resorting to the up-to technique. However, from $\bar{P}' \in \mathit{reach}(P)$ and $P \in$ SBrNDC it follows that $\bar{P}' \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'' \setminus \mathcal{A}_\mathcal{H}$, hence $\bar{P}' / \mathcal{A}_\mathcal{H} \approx_\mathrm{b} P'' / \mathcal{A}_\mathcal{H}$ by virtue of Lemma 4.7(2) because $\bar{P}', P'' \in$ SBrNDC. Therefore $P' \setminus \mathcal{A}_\mathcal{H}$ can stay idle in the up-to branching bisimulation game.
- SBrSNNI = P_BrNDC. We first prove that P_BrNDC $\subseteq$ SBrSNNI. If $P \in$ P_BrNDC then $P' \in$ BrNDC for every $P' \in \mathit{reach}(P)$. Since BrNDC $\subset$ BrSNNI as we will show in the last part of the proof of this theorem, $P' \in$ BrSNNI for every $P' \in \mathit{reach}(P)$, i.e., $P \in$ SBrSNNI.

  The fact that SBrSNNI $\subseteq$ P_BrNDC follows from Lemma 4.7(3) by taking P'_1 identical to P'_2 and both reachable from $P \in$ SBrSNNI.
- SBrSNNI $\subset$ BrNDC. If $P \in$ SBrSNNI = P_BrNDC then it immediately follows that $P \in$ BrNDC.
- BrNDC $\subset$ BrSNNI. If $P \in$ BrNDC, i.e., $P \setminus \mathcal{A}_\mathcal{H} \approx_\mathrm{b} (P \parallel_L Q) / L) \setminus \mathcal{A}_\mathcal{H}$ for all $Q \in \mathbb{P}$ such that every $Q' \in \mathit{reach}(Q)$ has only actions in $\mathcal{A}_\mathcal{H}$ and for all $L \subseteq \mathcal{A}_\mathcal{H}$, then we can consider in particular $\hat{Q}$ capable of stepwise mimicking the high-level behavior of P, in the sense that $\hat{Q}$ is able to synchronize with all the high-level actions executed by P

and its reachable processes, along with $\hat{L} = \mathcal{A}_{\mathcal{H}}$. As a consequence $(P \parallel_{\hat{L}} \hat{Q}) / \hat{L}) \setminus \mathcal{A}_{\mathcal{H}}$ is isomorphic to $P / \mathcal{A}_{\mathcal{H}}$, hence $P \setminus \mathcal{A}_{\mathcal{H}} \approx_{\rm b} P / \mathcal{A}_{\mathcal{H}}$, i.e., $P \in$ BrSNNI. □

All the inclusions above are strict as we now show:

- The process $\tau . l . \underline{0} + l . l . \underline{0} + h . l . \underline{0}$ is SBrSNNI (resp. P_BrNDC) because $(\tau . l . \underline{0} + l . l . \underline{0} + h . l . \underline{0}) \setminus \{h\} \approx_{\rm b} (\tau . l . \underline{0} + l . l . \underline{0} + h . l . \underline{0}) / \{h\}$ and action h is enabled only by the initial process so every reachable process is BrSNNI (resp. BrNDC). It is not SBrNDC because the low-level view of the process reached after action h, i.e., $(l . \underline{0}) \setminus \{h\}$, is not branching bisimilar to $(\tau . l . \underline{0} + l . l . \underline{0} + h . l . \underline{0}) \setminus \{h\}$.
- The process $l . \underline{0} + l . l . \underline{0} + l . h . l . \underline{0}$ is BrNDC because, whether there are synchronizations with high-level actions or not, the overall process can always perform either an l-action or a sequence of two l-actions without incurring any problematic branching. The process is not SBrSNNI (resp. P_BrNDC) because the reachable process $h . l . \underline{0}$ is not BrSNNI (resp. BrNDC).
- The process $l . \underline{0} + h . h . l . \underline{0}$ is BrSNNI as $(l . \underline{0} + h . h . l . \underline{0}) \setminus \{h\} \approx_{\rm b} (l . \underline{0} + h . h . l . \underline{0}) / \{h\}$. It is not BrNDC due to $(((l . \underline{0} + h . h . l . \underline{0}) \parallel_{\{h\}} (h . \underline{0})) / \{h\}) \setminus \{h\} \not\approx_{\rm b} (l . \underline{0} + h . h . l . \underline{0}) \setminus \{h\}$ because the former behaves as $l . \underline{0} + \tau . \underline{0}$ while the latter behaves as $l . \underline{0}$.

We further observe that each of the $\approx_{\rm b}$-based noninterference properties listed in Definition 4.1 implies the corresponding $\approx$-based noninterference property listed in Definition 2.5. This is simply due to the fact that $\approx_{\rm b}$ is finer than $\approx$ [GW96].

Theorem 4.9. *The following inclusions hold:*

(1) BrSNNI ⊂ BSNNI.
(2) BrNDC ⊂ BNDC.
(3) SBrSNNI ⊂ SBSNNI.
(4) P_BrNDC ⊂ P_BNDC.
(5) SBrNDC ⊂ SBNDC. ■

All the inclusions above are strict by virtue of the following result; for an example of P_1 and P_2 below, see Figure 1.

Theorem 4.10. *Let $P_1, P_2 \in \mathbb{P}$ be such that $P_1 \approx P_2$ but $P_1 \not\approx_{\rm b} P_2$. If no high-level actions occur in P_1 and P_2, then $Q \in \{P_1 + h . P_2, P_2 + h . P_1\}$ is such that:*

(1) $Q \in$ BSNNI *but* $Q \notin$ BrSNNI.
(2) $Q \in$ BNDC *but* $Q \notin$ BrNDC.
(3) $Q \in$ SBSNNI *but* $Q \notin$ SBrSNNI.
(4) $Q \in$ P_BNDC *but* $Q \notin$ P_BrNDC.
(5) $Q \in$ SBNDC *but* $Q \notin$ SBrNDC.

Proof. Let Q be $P_1 + h . P_2$ (the proof is similar for Q equal to $P_2 + h . P_1$) and observe that no high-level actions occur in every process reachable from Q except Q itself:

(1) Let $\mathcal{B}$ be a weak bisimulation witnessing $P_1 \approx P_2$. Then $Q \in$ BSNNI because the symmetric relation $\mathcal{B}' = \mathcal{B} \cup \{(Q \setminus \mathcal{A}_{\mathcal{H}}, Q / \mathcal{A}_{\mathcal{H}}), (Q / \mathcal{A}_{\mathcal{H}}, Q \setminus \mathcal{A}_{\mathcal{H}})\}$ turns out to be a weak bisimulation too. The only interesting case is the one where $Q / \mathcal{A}_{\mathcal{H}}$, which is isomorphic to $P_1 + \tau . P_2$, performs a τ-action toward $P_2 / \mathcal{A}_{\mathcal{H}}$, which is isomorphic to P_2. In that case $Q \setminus \mathcal{A}_{\mathcal{H}}$, which is isomorphic to P_1, can respond by staying idle, because $(P_2, P_1) \in \mathcal{B}$ and hence $(P_2, P_1) \in \mathcal{B}'$.
On the other hand, $Q \notin$ BrSNNI because $P_2 \not\approx_{\rm b} P_1$ in the same situation as before.

(2) Since $Q \in$ BSNNI and no high-level actions occur in every process reachable from Q other than Q, it holds that $Q \in$ SBSNNI and hence $Q \in$ BNDC by virtue of Theorem 2.6. On the other hand, from $Q \notin$ BrSNNI it follows that $Q \notin$ BrNDC by virtue of Theorem 4.8.
(3) We already know from the previous case that $Q \in$ SBSNNI.
On the other hand, from $Q \notin$ BrSNNI it follows that $Q \notin$ SBrSNNI by virtue of Theorem 4.8.
(4) A straightforward consequence of P_BNDC = SBSNNI (Theorem 2.6) and P_BrNDC = SBrSNNI (Theorem 4.8).
(5) Since the only high-level action occurring in Q is h, in the proof of $Q \in$ SBNDC the only interesting case is the transition $Q \xrightarrow{h} P_2$, for which it holds that $Q \setminus \mathcal{A}_{\mathcal{H}} \approx P_2 \setminus \mathcal{A}_{\mathcal{H}}$ because the former is isomorphic to P_1, the latter is isomorphic to P_2, and $P_1 \approx P_2$.
On the other hand, $Q \notin$ SBrNDC because $P_1 \not\approx_{\mathrm{b}} P_2$ in the same situation as before. □

An alternative strategy to explore the differences between $\approx$ and $\approx_{\mathrm{b}}$ with respect to B/BrSNNI and SB/BrSNNI consists of considering the two τ-axioms $\tau \,.\, x + x = \tau \,.\, x$ and $a \,.\, (\tau \,.\, x + y) + a \,.\, x = a \,.\, (\tau \,.\, x + y)$ for $\approx$ [Mil89]. The strategy is inspired by the initial remarks in [GW96], where it is noted that the two aforementioned axioms are not valid for $\approx_{\mathrm{b}}$ and are responsible for the lack of distinguishing power of $\approx$ over τ-branching processes. For each axiom, the strategy is based on constructing a pair of new processes from the ones equated in the axiom, such that they are weakly bisimilar by construction but not branching bisimilar. Then from this 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 constructed processes.

Fact 4.11. From $\tau \,.\, x + x = \tau \,.\, x$ it is possible to construct $P \in \mathbb{P}$ such that $P \in$ BSNNI but $P \notin$ BrSNNI and $P \in$ SBSNNI but $P \notin$ SBrSNNI.

Proof. In $\tau \,.\, x + x = \tau \,.\, x$ let us instantiate x as $\tau \,.\, l_1 \,.\, \underline{0} + \tau \,.\, l_2 \,.\, \underline{0}$ and then add $+ l_3 \,.\, \underline{0}$ to both sides of the equation thus obtaining $\tau \,.\, (\tau \,.\, l_1 \,.\, \underline{0} + \tau \,.\, l_2 \,.\, \underline{0}) + (\tau \,.\, l_1 \,.\, \underline{0} + \tau \,.\, l_2 \,.\, \underline{0}) + l_3 \,.\, \underline{0} = \tau \,.\, (\tau \,.\, l_1 \,.\, \underline{0} + \tau \,.\, l_2 \,.\, \underline{0}) + l_3 \,.\, \underline{0}$, which are related by weak bisimilarity but not by branching bisimilarity. Now let us define process P as $\tau \,.\, (\tau \,.\, l_1 \,.\, \underline{0} + \tau \,.\, l_2 \,.\, \underline{0}) + (h \,.\, l_1 \,.\, \underline{0} + h \,.\, l_2 \,.\, \underline{0}) + l_3 \,.\, \underline{0}$, for which it holds that $P/\mathcal{A}_{\mathcal{H}}$ and $P \setminus \mathcal{A}_{\mathcal{H}}$ are isomorphic to the two sides of the equation, respectively. By construction, it immediately follows that P is BSNNI but not BrSNNI.
Since the only high-level action is performed by P itself, which is BSNNI, for every other process P' reachable from P it holds that $P' \setminus \mathcal{A}_{\mathcal{H}}$ is isomorphic to $P' \,/\, \mathcal{A}_{\mathcal{H}}$, hence $P \in$ SBSNNI but $P \notin$ SBrSNNI. □

Fact 4.12. From $a \,.\, (\tau \,.\, x + y) + a \,.\, x = a \,.\, (\tau \,.\, x + y)$ it is possible to construct $P \in \mathbb{P}$ such that $P \in$ BSNNI but $P \notin$ BrSNNI and $P \in$ SBSNNI but $P \notin$ SBrSNNI.

Proof. In $a \,.\, (\tau \,.\, x + y) + a \,.\, x = a \,.\, (\tau \,.\, x + y)$ let us instantiate a as τ, x as $l_1 \,.\, \underline{0}$, and y as $l_2 \,.\, \underline{0}$ and then add $+ l_3 \,.\, \underline{0}$ to both sides of the equation thus obtaining $\tau \,.\, (\tau \,.\, l_1 \,.\, \underline{0} + l_2 \,.\, \underline{0}) + \tau \,.\, l_1 \,.\, \underline{0} + l_3 \,.\, \underline{0} = \tau \,.\, (\tau \,.\, l_1 \,.\, \underline{0} + l_2 \,.\, \underline{0}) + l_3 \,.\, \underline{0}$, which are related by weak bisimilarity but not by branching bisimilarity. Now let us define process P as $\tau \,.\, (\tau \,.\, l_1 \,.\, \underline{0} + l_2 \,.\, \underline{0}) + h \,.\, l_1 \,.\, \underline{0} + l_3 \,.\, \underline{0}$, for which it holds that $P \,/\, \mathcal{A}_{\mathcal{H}}$ and $P \setminus \mathcal{A}_{\mathcal{H}}$ are isomorphic to the two sides of the equation, respectively. By construction, it immediately follows that P is BSNNI but not BrSNNI.
Since the only high-level action is performed by P itself, which is BSNNI, for every other process P' reachable from P it holds that $P' \setminus \mathcal{A}_{\mathcal{H}}$ is isomorphic to $P' \,/\, \mathcal{A}_{\mathcal{H}}$, hence $P \in$ SBSNNI but $P \notin$ SBrSNNI. □

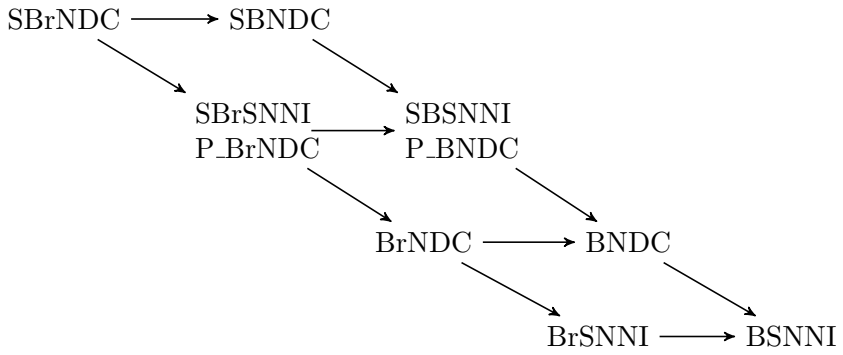

Figure 4: Taxonomy of security properties based on weak and branching bisimilarities

Based on the results in Theorems 2.6, 4.8, and 4.9, the diagram in Figure 4 summarizes the inclusions among the various noninterference properties, where $\mathcal{P} \to \mathcal{Q}$ means that $\mathcal{P}$ is strictly included in $\mathcal{Q}$. The missing arrows in the diagram, witnessing incomparability, are justified by the following counterexamples:

- SBNDC vs. SBrSNNI. The process $\tau \,.\, l \,.\, \underline{0} + l \,.\, l \,.\, \underline{0} + h \,.\, l \,.\, \underline{0}$ is BrSNNI as $\tau \,.\, l \,.\, \underline{0} + l \,.\, l \,.\, \underline{0} \approx_{\mathrm{b}} \tau \,.\, l \,.\, \underline{0} + l \,.\, l \,.\, \underline{0} + \tau \,.\, l \,.\, \underline{0}$. It is also SBrSNNI because every reachable process does not enable any more high-level actions. However, it is not SBNDC, because after executing the high-level action h it can perform a single l-action, while the original process with the restriction on high-level actions can go along a path where it performs two l-actions. On the other hand, the process Q mentioned in Theorem 4.10 is SBNDC but neither BrSNNI nor SBrSNNI.
- SBSNNI vs. BrNDC. The process $l \,.\, h \,.\, l \,.\, \underline{0} + l \,.\, \underline{0} + l \,.\, l \,.\, \underline{0}$ is BrSNNI as $l \,.\, \underline{0} + l \,.\, \underline{0} + l \,.\, l \,.\, \underline{0} \approx_{\mathrm{b}} l \,.\, \tau \,.\, l \,.\, \underline{0} + l \,.\, \underline{0} + l \,.\, l \,.\, \underline{0}$. In particular, the subprocesses $l \,.\, \tau \,.\, l \,.\, \underline{0}$ and $l \,.\, l \,.\, \underline{0}$ are equated by virtue of the other axiom of weak bisimilarity, $a \,.\, \tau \,.\, x = a \,.\, x$, which holds also for branching bisimilarity. The same process is BrNDC too as it includes only one high-level action, hence the only possible high-level strategy coincides with the check conducted by BrSNNI. However, the process is not SBSNNI because of the reachable process $h \,.\, l \,.\, \underline{0}$, which is not BSNNI. On the other hand, the process Q mentioned in Theorem 4.10 is SBSNNI but not BrSNNI and, therefore, cannot be BrNDC.
- BNDC vs. BrSNNI. The process $l \,.\, \underline{0} + h_1 \,.\, h_2 \,.\, l \,.\, \underline{0}$ is not BNDC (see Section 2.3), but it is BrSNNI as $l \,.\, \underline{0} \approx_{\mathrm{b}} l \,.\, \underline{0} + \tau \,.\, \tau \,.\, l \,.\, \underline{0}$. In contrast, the process Q mentioned in Theorem 4.10 is both BSNNI and BNDC, but not BrSNNI.

It is worth noting that the strongest property based on weak bisimilarity (SBNDC) and the weakest property based on branching bisimilarity (BrSNNI) are incomparable too. The former is a very restrictive property because it requires a local check every time a high-level action is performed, while the latter requires a check only on the initial state. On the other hand, as shown in Theorem 4.10 it is very easy to construct processes that are secure under properties based on $\approx$ but not on $\approx_{\mathrm{b}}$ due to the minimal number of high-level actions in Q.

5. Noninterference in Reversible Processes

As anticipated, we use reversible computing to motivate the study of branching-bisimilarity-based noninterference properties. To this aim, we now recall from [DMV90] back-and-forth bisimilarity and its relationship with standard forward-only bisimilarity.

An LTS represents a reversible process if each of its transitions is seen as bidirectional. This means that the action labeling every transition can be undone and then redone. When going backward, it is of paramount importance to respect causality. While this is straightforward for sequential processes, it is not obvious for concurrent ones, because the last performed action is the first one to be undone but this action may not necessarily be identifiable uniquely in the presence of concurrency.

Consider for example a process that can perform action a in parallel with action b. This process can be represented as a diamond-like LTS where from the initial state an a-transition and a b-transition depart, which are respectively followed by a b-transition and an a-transition both of which reach the final state. Suppose that action a completes before action b, so that the a-transition is executed before the b-transition. Once in the final state, either the b-transition is undone before the a-transition, or the a-transition is undone before the b-transition. Both options are causally consistent, as a and b are independent of each other, but only the former is history preserving too.

The history-preserving option is the one that was addressed in [DMV90] in order to study reversible processes in an interleaving setting. To accomplish this, strong and weak bisimulations were redefined as binary relations between histories, formalized below as runs, instead of states. The resulting behavioral equivalences are respectively called strong and weak back-and-forth bisimilarities in [DMV90].

Definition 5.1. Let $(\mathcal{S}, \mathcal{A}_\tau, \longrightarrow)$ be an LTS:

- A sequence $\xi = s \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \ldots s_{n-1} \xrightarrow{a_n} s_n$ is called a *path* from state $s \in \mathcal{S}$ of length $n \in \mathbb{N}$, where we let $\mathit{first}(\xi) = s$ and $\mathit{last}(\xi) = s_n$. We denote by $\mathit{Path}(s)$ the set of paths from state s, including the empty path indicated with ε.
- A pair $\rho = (s, \xi)$ is called a *run* from state $s \in \mathcal{S}$ iff $\xi \in \mathit{Path}(s)$, in which case we let $\mathit{path}(\rho) = \xi$, $\mathit{first}(\rho) = \mathit{first}(\xi)$, and $\mathit{last}(\rho) = \mathit{last}(\xi)$, with $\mathit{first}(\rho) = \mathit{last}(\rho) = s$ when $\xi = \varepsilon$. We denote by $\mathit{Run}(s)$ the set of runs from state s.
- Let $\rho = (s, \xi) \in \mathit{Run}(s)$ and $\rho' = (s', \xi') \in \mathit{Run}(s')$ for $s, s' \in \mathcal{S}$:
  - Their composition $\rho\rho' = (s, \xi\xi') \in \mathit{Run}(s)$ is defined iff $\mathit{last}(\rho) = \mathit{first}(\rho')$.
  - We write $\rho \xrightarrow{a} \rho'$ iff there exists $\bar\rho = (\bar s, \bar s \xrightarrow{a} s')$ with $\bar s = \mathit{last}(\rho)$ such that $\rho' = \rho\bar\rho$. ■

In the behavioral equivalences of [DMV90], for any given LTS the set $\mathcal{R}$ of its runs is considered in lieu of the set $\mathcal{S}$ of its states. Using runs instead of just paths is convenient in the case of an empty path so as to know the state under consideration. Given a pair of runs (ρ_1, ρ_2), in the two definitions below the forward clauses consider outgoing transitions whereas the backward clauses consider incoming transitions.

Definition 5.2. Let $(\mathcal{S}, \mathcal{A}_\tau, \longrightarrow)$ be an LTS and $s_1, s_2 \in \mathcal{S}$. We say that s_1 and s_2 are *strongly back-and-forth bisimilar*, written $s_1 \sim_{\rm bf} s_2$, iff $((s_1, \varepsilon), (s_2, \varepsilon)) \in \mathcal{B}$ for some strong back-and-forth bisimulation $\mathcal{B}$. A symmetric binary relation $\mathcal{B}$ over $\mathcal{R}$ is a *strong back-and-forth bisimulation* iff, whenever $(\rho_1, \rho_2) \in \mathcal{B}$, then:

- for each $\rho_1 \xrightarrow{a} \rho_1'$ there exists $\rho_2 \xrightarrow{a} \rho_2'$ such that $(\rho_1', \rho_2') \in \mathcal{B}$;
- for each $\rho_1' \xrightarrow{a} \rho_1$ there exists $\rho_2' \xrightarrow{a} \rho_2$ such that $(\rho_1', \rho_2') \in \mathcal{B}$. ■

Definition 5.3. Let $(\mathcal{S}, \mathcal{A}_\tau, \longrightarrow)$ be an LTS and $s_1, s_2 \in \mathcal{S}$. We say that s_1 and s_2 are *weakly back-and-forth bisimilar*, written $s_1 \approx_{\rm bf} s_2$, 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 $\mathcal{R}$ is a *weak back-and-forth bisimulation* iff, whenever $(\rho_1, \rho_2) \in \mathcal{B}$, then:

- for each $\rho_1 \stackrel{\tau}{\longrightarrow} \rho_1'$ there exists $\rho_2 \stackrel{\tau^*}{\Longrightarrow} \rho_2'$ such that $(\rho_1', \rho_2') \in \mathcal{B}$;
- for each $\rho_1' \stackrel{\tau}{\longrightarrow} \rho_1$ there exists $\rho_2' \stackrel{\tau^*}{\Longrightarrow} \rho_2$ such that $(\rho_1', \rho_2') \in \mathcal{B}$;
- for each $\rho_1 \stackrel{a}{\longrightarrow} \rho_1'$ with $a \in \mathcal{A}$ there exists $\rho_2 \stackrel{\tau^*}{\Longrightarrow} \stackrel{a}{\longrightarrow} \stackrel{\tau^*}{\Longrightarrow} \rho_2'$ such that $(\rho_1', \rho_2') \in \mathcal{B}$;
- for each $\rho_1' \stackrel{a}{\longrightarrow} \rho_1$ with $a \in \mathcal{A}$ there exists $\rho_2' \stackrel{\tau^*}{\Longrightarrow} \stackrel{a}{\longrightarrow} \stackrel{\tau^*}{\Longrightarrow} \rho_2$ such that $(\rho_1', \rho_2') \in \mathcal{B}$. ■

In [DMV90] it was shown that strong back-and-forth bisimilarity coincides with strong bisimilarity. Surprisingly, weak back-and-forth bisimilarity does not coincide with weak bisimilarity. Instead, it coincides with branching bisimilarity. For example, in Figure 1 it holds that $s_1 \not\approx_{\rm bf} s_2$ because in the forward direction $(s_1, \varepsilon) \stackrel{a}{\longrightarrow} (s_1, s_1 \stackrel{a}{\longrightarrow} s_1')$ is matched by $(s_2, \varepsilon) \stackrel{\tau}{\longrightarrow} (s_2, s_2 \stackrel{\tau}{\longrightarrow} s_2') \stackrel{a}{\longrightarrow} (s_2, s_2 \stackrel{\tau}{\longrightarrow} s_2' \stackrel{a}{\longrightarrow} s_2'')$, but then in the backward direction $(s_2, s_2 \stackrel{\tau}{\longrightarrow} s_2') \stackrel{a}{\longrightarrow} (s_2, s_2 \stackrel{\tau}{\longrightarrow} s_2' \stackrel{a}{\longrightarrow} s_2'')$ is not matched by $(s_1, \varepsilon) \stackrel{a}{\longrightarrow} (s_1, s_1 \stackrel{a}{\longrightarrow} s_1')$ because (s_1, ε) has an outgoing b-transition whilst $(s_2, s_2 \stackrel{\tau}{\longrightarrow} s_2')$ has not.

Theorem 5.4. *Let $(\mathcal{S}, \mathcal{A}_\tau, \longrightarrow)$ be an LTS and $s_1, s_2 \in \mathcal{S}$. Then:*

- $s_1 \sim_{\rm bf} s_2$ *iff* $s_1 \sim s_2$.
- $s_1 \approx_{\rm bf} s_2$ *iff* $s_1 \approx_{\rm b} s_2$. ■

As a consequence, the properties BrSNNI, BrNDC, SBrSNNI, P_BrDNC, and SBrNDC do not change if $\approx_{\rm b}$ is replaced by $\approx_{\rm bf}$. This allows us to study noninterference properties for reversible systems by using $\approx_{\rm b}$ in a standard process calculus like the one of Section 2.2, without having to decorate executed actions like in [PU07, BR23] or store them into stack-based memories like in [DK04].

6. Use Case: DBMS Authentication – Part II

The example provided in Section 3 is useful to illustrate the limitations of weak bisimilarity when investigating potential covert channels in reversible systems. In particular, it turns out that $Auth \setminus \mathcal{A}_\mathcal{H} \not\approx_{\rm b} Auth \,/\, \mathcal{A}_\mathcal{H}$, i.e., $Auth$ is not BrSNNI, and hence not even BrNDC, SBrSNNI, and SBrNDC by virtue of Theorem 4.8. As can be seen in Figure 3, the reason is that, if $Auth \,/\, \mathcal{A}_\mathcal{H}$ performs the leftmost τ-action and hence moves to state r_3', from which the only executable action is l_{sso}, then according to the definition of branching bisimilarity $Auth \setminus \mathcal{A}_\mathcal{H}$ can:

- either stay idle, but from that state $Auth \setminus \mathcal{A}_\mathcal{H}$ can then perform actions other than l_{sso} that cannot be matched on the side of $Auth \,/\, \mathcal{A}_\mathcal{H}$;
- or perform two τ-actions thereby reaching state s_3, but the last traversed state, i.e., s_2, is not branching bisimilar to the initial state of $Auth \,/\, \mathcal{A}_\mathcal{H}$.

In a standard model of execution, where the computation can proceed only forward, the distinguishing power of branching bisimilarity may be considered too severe, as no practical covert channel actually occurs and the system can be deemed noninterfering as shown in Section 3. Indeed, a low-level user has no possibility of distinguishing the internal move performed by $Auth \,/\, \mathcal{A}_\mathcal{H}$ that leads to $l_{sso} \,.\, Auth$ from the sequence of internal moves

performed by $Auth \setminus \mathcal{A}_{\mathcal{H}}$ that lead to $l_{sso} \,.\, Auth$ as well. This motivates the fact that, historically, weak bisimilarity has been preferred in the setting of noninterference.

Now we know that, if we replace the branching bisimulation semantics with the weak back-and-forth bisimulation semantics, nothing changes about the outcome of noninterference verification. Assuming that the DBMS allows transactions to be reversed, it is instructive to discuss why BrSNNI is not satisfied by following the formalization of the weak back-and-forth bisimulation semantics provided in Section 5.

After $Auth \,/\, \mathcal{A}_{\mathcal{H}}$ performs the run $(r_1, (r_1 \xrightarrow{\tau} r_3' \xrightarrow{l_{sso}} r_1))$, process $Auth \setminus \mathcal{A}_{\mathcal{H}}$ can respond by performing the run $(s_1, (s_1 \xrightarrow{\tau} s_2 \xrightarrow{\tau} s_3 \xrightarrow{l_{sso}} s_1))$. If either process goes back by undoing l_{sso}, then the other one can undo l_{sso} as well and the states r_3' and s_3 are reached. However, if $Auth \setminus \mathcal{A}_{\mathcal{H}}$ goes further back by undoing $s_2 \xrightarrow{\tau} s_3$ too, then $Auth \,/\, \mathcal{A}_{\mathcal{H}}$ can:

- either undo $r_1 \xrightarrow{\tau} r_3'$, but in this case r_1 enables action l_{pwd} while s_2 does not;
- or stay idle, but in this case r_3' enables only l_{sso}, while s_2 can go along the path $s_2 \xrightarrow{\tau} s_4 \xrightarrow{l_{2fa}} s_1$ as well.

This line of reasoning immediately allows us to reveal a potential covert channel under reversible computing. In fact, let us assume that the transaction modeled by $Auth$ is not only executed forward, but also enables backward computations triggered, e.g., whenever debugging mode is activated. This may happen in response to some user-level malfunctioning, which may be due, for instance, to the authentication operation or to the transaction execution. As formally shown above, if the action l_{sso} performed at r_3' after the high-level interaction is undone along with the latter, then the system enables again the execution of the action l_{pwd}. This is motivated in our example by the fact that, by virtue of the transaction rollback, any kind of authentication becomes admissible again. On the other hand, this is not possible after undoing the action l_{sso} performed at state s_3, because in such a case the internal decision of the DBMS of adopting a highly secure mechanism is not reversed. In other words, by reversing the computation the low-level user can become aware of the fact that the transaction data are feeding the training set or not.

In the literature, there are several reverse debuggers working in this way like, e.g., UndoDB [Eng12], a Linux-based interactive time-travel debugger that can handle multiple threads and their backward execution. For instance, it is integrated within the DBMS SAP HANA [UDB] in order to reduce time-to-resolution of software failures. In our example, by virtue of the observations conducted above, if the system is executed backward just after performing l_{sso}, a low-level user can decide whether a high-level action had occurred before or not, thus revealing a covert channel. Such a covert channel is completely concealed during the forward execution of the system and is detected only when the system is executed backward. In general, this may happen when the reverse debugger is activated by virtue of some unexpected event (e.g., segmentation fault, stack overflow, memory corruption) caused intentionally or not, and by virtue of which some undesired information flow emerges toward low-level users.

7. Conclusions

Our study of branching-bisimilarity-based noninterference properties has established a connection with reversible computing, in the sense that those properties are directly applicable to reversible systems expressed in a standard process language with no need of decorating

executed actions [PU07, BR23] or storing them into stack-based memories [DK04]. To the best of our knowledge, this is the first attempt of defining noninterference properties relying on branching bisimilarity so as to reason about information leakage in reversible systems.

In particular, we have rephrased in the setting of branching bisimilarity the taxonomy of nondeterministic noninterference properties based on weak bisimilarity [FG01, FR06]. This generates an extended taxonomy (Figure 4) that is conservative with respect to the classical one and emphasizes the strictness of certain property inclusions as well as the incomparability of other properties. In addition, we have studied preservation (Theorem 4.3) and compositionality (Theorem 4.5) features of the new noninterference properties relying on branching bisimilarity. Moreover, some ancillary results (Lemmas 4.4 and 4.7) have been established about parallel composition, restriction, and hiding under branching bisimilarity, SBrSNNI, and SBrNDC, which elicit general patterns applicable also under weak bisimilarity, SBSNNI, and SBNDC.

With respect to the earlier version of our study [EAB23], the considered process language now supports recursion – which has required us to introduce the aforementioned ancillary results and resort to the notion of branching bisimulation up to $\approx_{\mathrm{b}}$ – and the taxonomy has been extended in such a way to include a persistent variant of the property of non-deducibility on composition.

We have then shown through a database management system example that potential covert channels arising in reversible systems cannot be revealed by employing weak bisimulation semantics. Indeed, the higher discriminating power of branching bisimilarity is necessary to capture information flows emerging when backward computations are admitted. The correspondence discovered in [DMV90] between branching bisimilarity and weak back-and-forth bisimilarity confirms the adequacy of our approach based on the former.

As for future work, we are planning to further extend the noninterference taxonomy so as to include more expressive properties that take into account also quantitative aspects of process behavior like in [ABG04, HMPR21]. To accomplish this for reversible systems, it is necessary to understand whether the approach of [DMV90] generalizes to quantitative back-and-forth bisimilarities. Some results for the probabilistic case can be found in [EAB24].

Acknowledgment

This research has been supported by the PRIN 2020 project *NiRvAna – Noninterference and Reversibility Analysis in Private Blockchains*. We are grateful to Rob van Glabbeek for the valuable discussions on up-to techniques for branching bisimilarity. We finally thank the anonymous reviewers for their comments and suggestions.

References

[AB11] A. Aldini and M. Bernardo. Component-oriented verification of noninterference. *Journal of Systems Architecture*, 57:282–293, 2011.

[ABG04] A. Aldini, M. Bravetti, and R. Gorrieri. A process-algebraic approach for the analysis of probabilistic noninterference. *Journal of Computer Security*, 12:191–245, 2004.

[AC19] M. Al-Rubaie and J.M. Chang. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17:49–58, 2019.

[Ald06] A. Aldini. Classification of security properties in a Linda-like process algebra. *Science of Computer Programming*, 63:16–38, 2006.

[Ben73] C.H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.

[BFLX22] Y. Bai, M. Fan, Y. Li, and C. Xie. Privacy risk assessment of training data in machine learning. In *Proc. of the 34th IEEE Int. Conf. on Communications (ICC 2022)*, pages 1015–1015. IEEE-CS Press, 2022.

[BHR84] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31:560–599, 1984.

[Boo20] S. Boonkrong. *Authentication and Access Control*. Apress, 2020.

[BR23] M. Bernardo and S. Rossi. Reverse bisimilarity vs. forward bisimilarity. In *Proc. of the 26th Int. Conf. on Foundations of Software Science and Computation Structures (FOSSACS 2023)*, volume 13992 of *LNCS*, pages 265–284. Springer, 2023.

[DK04] V. Danos and J. Krivine. Reversible communicating systems. In *Proc. of the 15th Int. Conf. on Concurrency Theory (CONCUR 2004)*, volume 3170 of *LNCS*, pages 292–307. Springer, 2004.

[DK05] V. Danos and J. Krivine. Transactions in RCCS. In *Proc. of the 16th Int. Conf. on Concurrency Theory (CONCUR 2005)*, volume 3653 of *LNCS*, pages 398–412. Springer, 2005.

[DMV90] R. De Nicola, U. Montanari, and F. Vaandrager. Back and forth bisimulations. In *Proc. of the 1st Int. Conf. on Concurrency Theory (CONCUR 1990)*, volume 458 of *LNCS*, pages 152–165. Springer, 1990.

[EAB23] A. Esposito, A. Aldini, and M. Bernardo. Branching bisimulation semantics enables noninterference analysis of reversible systems. In *Proc. of the 43rd Int. Conf. on Formal Techniques for Distributed Objects, Components, and Systems (FORTE 2023)*, volume 13910 of *LNCS*, pages 57–74. Springer, 2023.

[EAB24] A. Esposito, A. Aldini, and M. Bernardo. Noninterference analysis of reversible probabilistic systems. In *Proc. of the 44th Int. Conf. on Formal Techniques for Distributed Objects, Components, and Systems (FORTE 2024)*, volume 14678 of *LNCS*, pages 39–59. Springer, 2024.

[Eng12] J. Engblom. A review of reverse debugging. In *Proc. of the 4th System, Software, SoC and Silicon Debug Conf. (S4D 2012)*, pages 1–6. IEEE-CS Press, 2012.

[FG01] R. Focardi and R. Gorrieri. Classification of security properties. In *Proc. of the 1st Int. School on Foundations of Security Analysis and Design (FOSAD 2000)*, volume 2171 of *LNCS*, pages 331–396. Springer, 2001.

[FPR02] R. Focardi, C. Piazza, and S. Rossi. Proofs methods for bisimulation based information flow security. In *Proc. of the 3rd Int. Workshop on Verification, Model Checking, and Abstract Interpretation (VMCAI 2002)*, volume 2294 of *LNCS*, pages 16–31. Springer, 2002.

[FR06] R. Focardi and S. Rossi. Information flow security in dynamic contexts. *Journal of Computer Security*, 14:65–110, 2006.

[Gla93] R.J. van Glabbeek. A complete axiomatization for branching bisimulation congruence of finite-state behaviours. In *Proc. of the 18th Int. Symp. on Mathematical Foundations of Computer Science (MFCS 1993)*, volume 711 of *LNCS*, pages 473–484. Springer, 1993.

[Gla01] R.J. van Glabbeek. The linear time – branching time spectrum I. In *Handbook of Process Algebra*, pages 3–99. Elsevier, 2001.

[GLM14] E. Giachino, I. Lanese, and C.A. Mezzina. Causal-consistent reversible debugging. In *Proc. of the 17th Int. Conf. on Fundamental Approaches to Software Engineering (FASE 2014)*, volume 8411 of *LNCS*, pages 370–384. Springer, 2014.

[GM82] J.A. Goguen and J. Meseguer. Security policies and security models. In *Proc. of the 2nd IEEE Symp. on Security and Privacy (SSP 1982)*, pages 11–20. IEEE-CS Press, 1982.

[GM18] R. Giacobazzi and I. Mastroeni. Abstract non-interference: A unifying framework for weakening information-flow. *ACM Trans. on Privacy and Security*, 21(2):9:1–9:31, 2018.

[GV90] J.F. Groote and F. Vaandrager. An efficient algorithm for branching bisimulation and stuttering equivalence. In *Proc. of the 17th Int. Coll. on Automata, Languages and Programming (ICALP 1990)*, volume 443 of *LNCS*, pages 626–638. Springer, 1990.

[GW96] R.J. van Glabbeek and W.P. Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the ACM*, 43:555–600, 1996.

[HMPR21] J. Hillston, A. Marin, C. Piazza, and S. Rossi. Persistent stochastic non-interference. *Fundamenta Informaticae*, 181:1–35, 2021.

[HS12] D. Hedin and A. Sabelfeld. A perspective on information-flow control. In *Software Safety and Security – Tools for Analysis and Verification*, pages 319–347. IOS Press, 2012.

[JGKW20] D.N. Jansen, J.F. Groote, J.J.A. Keiren, and A. Wijs. An $O(m \log n)$ algorithm for branching bisimilarity on labelled transition systems. In *Proc. of the 26th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS 2020)*, volume 12079 of *LNCS*, pages 3–20. Springer, 2020.

[Kel76] R.M. Keller. Formal verification of parallel programs. *Communications of the ACM*, 19:371–384, 1976.

[Lan61] R. Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5:183–191, 1961.

[LES18] J.S. Laursen, L.-P. Ellekilde, and U.P. Schultz. Modelling reversible execution of robotic assembly. *Robotica*, 36:625–654, 2018.

[LLM+13] I. Lanese, M. Lienhardt, C.A. Mezzina, A. Schmitt, and J.-B. Stefani. Concurrent flexible reversibility. In *Proc. of the 22nd European Symp. on Programming (ESOP 2013)*, volume 7792 of *LNCS*, pages 370–390. Springer, 2013.

[LNPV18] I. Lanese, N. Nishida, A. Palacios, and G. Vidal. CauDEr: A causal-consistent reversible debugger for Erlang. In *Proc. of the 14th Int. Symp. on Functional and Logic Programming (FLOPS 2018)*, volume 10818 of *LNCS*, pages 247–263. Springer, 2018.

[Man11] H. Mantel. Information flow and noninterference. In *Encyclopedia of Cryptography and Security*, pages 605–607. Springer, 2011.

[Mar03] F. Martinelli. Analysis of security protocols as open systems. *Theoretical Computer Science*, 290:1057–1106, 2003.

[Mil89] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[Par81] D. Park. Concurrency and automata on infinite sequences. In *Proc. of the 5th GI Conf. on Theoretical Computer Science*, volume 104 of *LNCS*, pages 167–183. Springer, 1981.

[Pin17] G.M. Pinna. Reversing steps in membrane systems computations. In *Proc. of the 18th Int. Conf. on Membrane Computing (CMC 2017)*, volume 10725 of *LNCS*, pages 245–261. Springer, 2017.

[PP14] K.S. Perumalla and A.J. Park. Reverse computation for rollback-based fault tolerance in large parallel systems – Evaluating the potential gains and systems effects. *Cluster Computing*, 17:303–313, 2014.

[PU07] I. Phillips and I. Ulidowski. Reversing algebraic process calculi. *Journal of Logic and Algebraic Programming*, 73:70–96, 2007.

[PUY12] I. Phillips, I. Ulidowski, and S. Yuen. A reversible process calculus and the modelling of the ERK signalling pathway. In *Proc. of the 4th Int. Workshop on Reversible Computation (RC 2012)*, volume 7581 of *LNCS*, pages 218–232. Springer, 2012.

[SM92] D. Sangiorgi and R. Milner. The problem of "weak bisimulation up to". In *Proc. of the 3rd Int. Conf. on Concurrency Theory (CONCUR 1992)*, volume 630 of *LNCS*, pages 32–46. Springer, 1992.

[SOJB18] M. Schordan, T. Oppelstrup, D.R. Jefferson, and P.D. Barnes Jr. Generation of reversible C++ code for optimistic parallel discrete event simulation. *New Generation Computing*, 36:257–280, 2018.

[SPP19] H. Siljak, K. Psara, and A. Philippou. Distributed antenna selection for massive MIMO using reversing Petri nets. *IEEE Wireless Communication Letters*, 8:1427–1430, 2019.

[UDB] UndoDB case studies. Last visited November 2024. URL: `https://undo.io/resources/type/case-studies/`.

[VKH10] E. de Vries, V. Koutavas, and M. Hennessy. Communicating transactions. In *Proc. of the 21st Int. Conf. on Concurrency Theory (CONCUR 2010)*, volume 6269 of *LNCS*, pages 569–583. Springer, 2010.

[VS18] M. Vassor and J.-B. Stefani. Checkpoint/rollback vs causally-consistent reversibility. In *Proc. of the 10th Int. Conf. on Reversible Computation (RC 2018)*, volume 11106 of *LNCS*, pages 286–303. Springer, 2018.

[ZM04] L. Zheng and A. Myers. Dynamic security labels and noninterference. In *Proc. of the 2nd IFIP Workshop on Formal Aspects in Security and Trust (FAST 2004)*, volume 173 of *IFIP AICT*, pages 27–40. Springer, 2004.