Alternative Characterizations of Hereditary History-Preserving Bisimilarity via Backward Ready Multisets

Marco Bernardo, Andrea Esposito, and Claudio A. Mezzina

Dipartimento di Scienze Pure e Applicate, Università di Urbino, Urbino, Italy

Abstract. We provide two alternative characterizations of hereditary history-preserving bisimilarity: a denotational one, on stable configuration structures, and an operational one, on a reversible process calculus. The characterizing equivalence is forward-reverse bisimilarity extended with a check for backward ready multiset equality. Unlike previous approaches, the focus is thus on counting identically labeled events rather than uniquely identifying them. We also investigate the relationships between event identifier logic, characterizing the former bisimilarity, and backward ready multiset logic, characterizing the latter bisimilarity.

1 Introduction

In the spectrum of truly concurrent bisimilarities [23,19,32], there are two equivalences that are particularly important: history-preserving bisimilarity [34] and hereditary history-preserving bisimilarity [6]. They are the coarsest equivalence and the finest equivalence, respectively, that are preserved under action refinement and are capable of respecting causality, branching, and their interplay while abstracting from choices between identical alternatives [23]. Moreover, hereditary history-preserving bisimilarity can be obtained as a special case of a categorical definition of bisimilarity over concurrency models [25].

History-preserving and hereditary history-preserving bisimilarities are defined over truly concurrent models such as event structures [35] or their variants, in particular configuration structures [24]. A configuration is a finite set of non-conflicting events that is downward-closed with respect to a causality relation over events. The bisimulation game compares configuration transitions. While history-preserving bisimilarity considers only outgoing transitions, hereditary history-preserving bisimilarity takes into account also incoming transitions. In other words, the former stepwise matches only forward computations, whereas the latter examines backward computations too. Both equivalences rely on ternary bisimulation relations, where the third component is a labeling- and causality-preserving bijection from the set of events executed so far in the first structure to the set of events executed so far in the second structure.

Logical characterizations of both equivalences have been provided in [33,4]. Furthermore, an axiomatization for hereditary history-preserving bisimilarity has been developed over forward-only processes in [21]. Finally, history-preserving

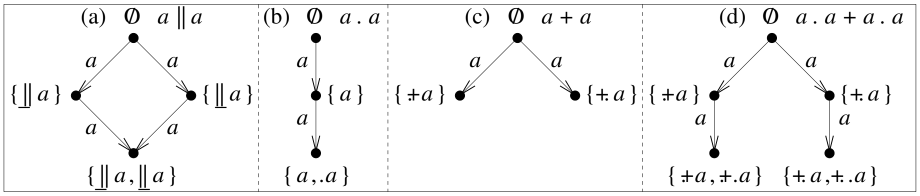

Fig. 1. Configuration graphs: autoconcurrency (a), autocausation (b), and autoconflict

bisimilarity is known to coincide with causal bisimilarity [15,16], hence the latter offers a characterization and an axiomatization [18] for the former. In this paper, we concentrate on characterizations of hereditary history-preserving bisimilarity.

The first alternative characterization of hereditary history-preserving bisimilarity has appeared in [6] for configuration graphs of prime event structures. The characterizing equivalence is called back-and-forth bisimilarity – not to be confused with the homonymous one in [17], which retrieves an interleaving semantics by constraining backward computations to take place along the corresponding forward computations even in the presence of concurrency. The main difference between hereditary history-preserving bisimilarity and back-and-forth bisimilarity is that the latter relies on binary bisimulation relations, hence no labeling- and causality-preserving bijection is stepwise built during the bisimulation game. The characterization result holds under the assumption of no autoconcurrency, i.e., the absence of configurations from which it is possible to execute two identically labeled, distinct events that are not in conflict with each other.

In Figures 1(a) and (b) we show the configuration graphs respectively associated with the following two processes for a given action a:

- Autoconcurrency on a, which is expressed as $a \parallel a$ where $\parallel$ stands for parallel composition. There are two equally labeled, non-conflicting events, denoted by $\underline{\|}a$ and $\|\!\underline{\,}a$, that can be executed in any order.
- Autocausation on a, which is expressed as $a \, . \, a$ where dot represents action prefix. There are two equally labeled, non-conflicting events, denoted by a and $.a$, such that the former has to be executed before the latter.

These two configuration graphs are back-and-forth bisimilar as witnessed by the symmetric binary relation that contains the pairs of configurations $(\emptyset, \emptyset)$, $(\{\underline{\|}a\}, \{a\})$, $(\{\|\!\underline{\,}a\}, \{a\})$, and $(\{\underline{\|}a, \|\!\underline{\,}a\}, \{a, .a\})$. However, they are not hereditary history-preserving bisimilar because, with respect to the last pair, there is no (labeling- and) causality-preserving bijection that maps the two independent events $\underline{\|}a$ and $\|\!\underline{\,}a$ to the two causally-related events a and $.a$.

The second alternative characterization of hereditary history-preserving bisimilarity has been given in [31]. The characterizing equivalence is the forward-reverse bisimilarity – very close in spirit to the back-and-forth bisimilarity of [6] – originally defined in [30] for a reversible variant of CCS [29] called CCSK. The operational semantics of CCSK produces labeled transition systems based on a forward transition relation and a backward one ensuring the loop property [13]. Each transition label comprises an action and a communication key; the latter is necessary when building backward transitions so as to know who synchronized with whom in the forward direction. In [31] forward-reverse bisimilarity has been generalized to configuration graphs of prime event structures and shown to coincide with hereditary history-preserving bisimilarity in the absence of repeated, identically labeled events along forward computations, which implies the absence of autoconcurrency (and autocausation), i.e., the assumption made in [6].

In [32] it has been shown, by working on stable configuration structures, how to relax the conditions under which the two characterization results of [6] and [31] hold. Specifically, it is sufficient to require the absence of equidepth autoconcurrency, i.e., the absence of identically labeled events occurring at the same depth within a configuration; the depth of an event is defined as the length of the longest causal chain of events up to and including the considered event.

The third alternative characterization of hereditary history-preserving bisimilarity has been provided in [3] and, unlike the previous two, does not need any restrictive assumption. Based on earlier work [2] – in which hereditary history-preserving bisimilarity was shown to coincide with back-and-forth barbed bisimulation congruence over singly-labeled processes, i.e., processes with no autoconcurrency and autoconflict (see Figure 1(c)) – it has been developed in the setting of a different reversible variant of CCS called RCCS [13,14,27]. While in CCSK all executed actions and discarded alternative subprocesses are kept within the syntax of processes so as to enable reversibility, in RCCS the same information is stored into stack-based memories attached to processes; the two approaches have been proven to be equivalent in [28]. The idea in [3] is to import hereditary history-preserving bisimilarity in the RCCS setting by encoding memories, i.e., the past behavior, as identified configuration structures. These are stable configuration structures enriched with unique event identifiers, used in transition labels and exploited when undoing synchronizations. The characterizing equivalence, called back-and-forth bisimilarity and defined over RCCS processes, relies on ternary bisimulation relations in which the third component is a bijection from the set of identifiers of the actions executed so far in the first process to the set of identifiers of the actions executed so far in the second process.

Having to reintroduce a third component in the bisimulation relations in order to exactly characterize hereditary history-preserving bisimilarity amounts to certifying that "reversibility is not just back and forth" [3], i.e., the forward and backward bisimulation games alone are not enough. The question then becomes whether and to what extent a systematic event identification is really necessary.

This question also arises from the fact that, in the aforementioned bisimulation games, CCSK transition labels such as $a[i]$ and $a[j]$ are deemed to be different if the two keys i and j are different [30] – which results in the absence of repeated, identically labeled events along forward computations [31] – while identified RCCS transition labels like $i{:}a$ and $j{:}a$ are viewed as compatible even if i and j are different [3]. On the one hand, in CCSK the two processes $a \parallel a$ and $a \,.\, a$ are told apart by forward-reverse bisimilarity because the former evolves to

$a[i] \,\|\, a[j]$, which can undo $a[i]$ and $a[j]$ in any order, while the latter evolves to $a[i] \,.\, a[j]$, from which only $a[j]$ can be undone, hence undoing $a[i]$ cannot be matched by undoing $a[j]$. On the other hand, in identified RCCS the same two processes are distinguished by back-and-forth bisimilarity because, although undoing $i{:}a$ can be matched by undoing $j{:}a$, it is not possible to establish a suitable bijection from a distributed memory containing $i{:}a$ in a location and $j{:}a$ in another location to a centralized memory containing $j{:}a$ on top of $i{:}a$.

In this paper we propose a totally different approach to exactly characterize hereditary history-preserving bisimilarity. Rather than the unique identification of identically labeled events, the focus is on *counting* them. Let us consider again Figures 1(a) and (b). If we look at the two top (resp. bottom) configurations, we note that the one on the left has two outgoing (resp. incoming) transitions, while the one on the right has only one. As for the top configuration on the left, in principle we may not know whether the branch is due to the fact that the two events are concurrent or conflicting. However, for the bottom configuration on the left we can certainly say that the two events are concurrent, as the models we are considering are truly concurrent and hence the configuration graph of process $a \,.\, a + a \,.\, a$ where + stands for nondeterministic choice (see Figure 1(d)) cannot be isomorphic to the one of $a \,\|\, a$ because it must have two different bottom configurations ($\{\mathord{.}\!\!+a, \mathord{.}\!\!+.a\}$ and $\{+\!\!\mathord{.}\,a, +\!\!\mathord{.}\,.a\}$) instead of a single one.

As an extension of the notion of backward ready set exploited in [8] to axiomatize forward-reverse bisimilarity over reversible concurrent processes, we define the *backward ready multiset* of a configuration or process to be the multiset of labels of its incoming transitions. After recalling in Section 2 the definitions of stable configuration structure [24], hereditary history-preserving bisimilarity [6], and event identifier logic [33], we provide the following contributions:

- In Section 3 we exhibit a denotational characterization on stable configuration structures: hereditary history-preserving bisimilarity turns out to coincide with forward-reverse bisimilarity extended with a clause for checking the equality of the backward ready multisets of matching configurations.
- In Section 4 we exhibit an operational characterization based on a variant of the reversible process calculus of [9,8] where executed action identification is limited to synchronizations. After revising its proved operational semantics inspired by [18] so as to faithfully account for causality and concurrency, we set up a backward-ready-multiset variant of forward-reverse bisimilarity and devise a backward ready multiset logic characterizing it. Then we define a denotational semantics based on stable configuration structures in which events are formalized as proof terms [10,11], so as to import the notion of hereditary history-preserving bisimilarity. We show that the stable configuration structures associated with two processes are hereditary history-preserving bisimilar iff the two processes are equated by the backward-ready-multiset variant of forward-reverse bisimilarity.
- In Section 5 we start the investigation of the relationships between the event identifier logic of [33] and our backward ready multiset logic.

Section 6 concludes the paper with directions for future work.

2 Hereditary History-Preserving Bisimilarity

In this section we recall hereditary history-preserving bisimilarity [6] over stable configuration structures [24] along with its logical characterization based on event identifier logic [33].

In the following two definitions taken from [23], $\mathcal{P}_{\text{fin}}(\mathcal{E})$ denotes the set of finite subsets of set $\mathcal{E}$ while $f \restriction X$ denotes the restriction of function f to set X.

Definition 1. *A* configuration structure *is a quadruple* $\mathsf{C} = (\mathcal{E}, \mathcal{C}, \mathcal{A}, l)$ *where:*

- $\mathcal{E}$ *is a set of* events.
- $\mathcal{C} \subseteq \mathcal{P}_{\text{fin}}(\mathcal{E})$ *is a set of* configurations.
- $\mathcal{A}$ *is a countable set of* labels.
- $l : \bigcup_{X \in \mathcal{C}} X \to \mathcal{A}$ *is a* labeling function.

C *is said to be* stable *iff it is:*

- *Rooted:* $\emptyset \in \mathcal{C}$.
- *Connected:* $\forall X \in \mathcal{C} \setminus \{\emptyset\}. \exists e \in X. X \setminus \{e\} \in \mathcal{C}$.
- *Closed under bounded unions and intersections:* $\forall X, Y, Z \in \mathcal{C}. X \cup Y \subseteq Z \implies X \cup Y, X \cap Y \in \mathcal{C}$.

The causality relation *over* $X \in \mathcal{C}$ *is defined by letting* $e_1 \leq_X e_2$ *for* $e_1, e_2 \in X$ *iff* $e_2 \in Y$ *implies* $e_1 \in Y$ *for all* $Y \in \mathcal{C}$ *such that* $Y \subseteq X$*; we write* $e_1 <_X e_2$ *when* $e_1 \leq_X e_2$ *and* $e_1 \neq e_2$*. Two events* $e_1, e_2 \in X$ *are* concurrent *in* X *iff* $e_1 \not<_X e_2$ *and* $e_2 \not<_X e_1$*. We write* $X \xrightarrow{a}_{\mathsf{C}} X'$ *for* $X, X' \in \mathcal{C}$ *and* $a \in \mathcal{A}$ *iff* $X \subseteq X'$, $X' \setminus X = \{e\}$*, and* $l(e) = a$. ■

Definition 2. *We say that two stable configuration structures* $\mathsf{C}_i = (\mathcal{E}_i, \mathcal{C}_i, \mathcal{A}, l_i)$, $i \in \{1, 2\}$*, are* hereditary history-preserving bisimilar*, written* $\mathsf{C}_1 \sim_{\text{HHPB}} \mathsf{C}_2$*, iff there exists a hereditary history-preserving bisimulation between* C_1 *and* C_2*, i.e., a relation* $\mathcal{B} \subseteq \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{P}(\mathcal{E}_1 \times \mathcal{E}_2)$ *such that:*

- $(\emptyset, \emptyset, \emptyset) \in \mathcal{B}$.
- *Whenever* $(X_1, X_2, f) \in \mathcal{B}$ *then:*
  - $f \subseteq \mathcal{E}_1 \times \mathcal{E}_2$ *is a bijection from* $X_1 \in \mathcal{C}_1$ *to* $X_2 \in \mathcal{C}_2$ *that preserves:*
    - *Labeling:* $l_1(e) = l_2(f(e))$ *for all* $e \in X_1$.
    - *Causality:* $e \leq_{X_1} e' \iff f(e) \leq_{X_2} f(e')$ *for all* $e, e' \in X_1$.
  - *For each* $X_1 \xrightarrow{a}_{\mathsf{C}_1} X'_1$ *there exist* $X_2 \xrightarrow{a}_{\mathsf{C}_2} X'_2$ *and* $f' \subseteq \mathcal{E}_1 \times \mathcal{E}_2$ *such that* $(X'_1, X'_2, f') \in \mathcal{B}$ *and* $f' \restriction X_1 = f$*, and vice versa.*
  - *For each* $X'_1 \xrightarrow{a}_{\mathsf{C}_1} X_1$ *there exist* $X'_2 \xrightarrow{a}_{\mathsf{C}_2} X_2$ *and* $f' \subseteq \mathcal{E}_1 \times \mathcal{E}_2$ *such that* $(X'_1, X'_2, f') \in \mathcal{B}$ *and* $f \restriction X'_1 = f'$*, and vice versa.* ■

Since there is a single transition relation, similar to [17,6] in the bisimulation game above a distinction is made between the outgoing transitions of X_1 and X_2 ($X_1 \xrightarrow{a}_{\mathsf{C}_1} X'_1$ and $X_2 \xrightarrow{a}_{\mathsf{C}_2} X'_2$ in the forward direction) and their incoming transitions ($X'_1 \xrightarrow{a}_{\mathsf{C}_1} X_1$ and $X'_2 \xrightarrow{a}_{\mathsf{C}_2} X_2$ in the backward direction).

Hereditary history-preserving bisimilarity is characterized by *event identifier logic* [33]. The set $\mathcal{L}_{\rm EI}$ of its formulas is generated by the following syntax:

$$\phi \ ::= \ \mathsf{true} \mid \neg \phi \mid \phi \wedge \phi \mid \langle x : a \rangle\!\rangle \phi \mid (x : a) \phi \mid \langle\!\langle x \rangle \phi$$

where $a \in \mathcal{A}$ and $x \in \mathcal{I}$, with $\mathcal{I}$ being a countable set of identifiers. The unary operators $\langle x : a \rangle\!\rangle \_$ and $(x : a) \_$ act as binders for the identifiers inside them. Therefore, the set of identifiers that occur *free* in $\phi \in \mathcal{L}_{\rm EI}$ is defined by induction on the syntactical structure of ϕ as follows:

$$\begin{aligned} \mathit{fi}(\mathsf{true}) &= \emptyset \\ \mathit{fi}(\neg \phi) &= \mathit{fi}(\phi) \\ \mathit{fi}(\phi_1 \wedge \phi_2) &= \mathit{fi}(\phi_1) \cup \mathit{fi}(\phi_2) \\ \mathit{fi}(\langle x : a \rangle\!\rangle \phi) &= \mathit{fi}(\phi) \setminus \{x\} \\ \mathit{fi}((x : a) \phi) &= \mathit{fi}(\phi) \setminus \{x\} \\ \mathit{fi}(\langle\!\langle x \rangle \phi) &= \mathit{fi}(\phi) \cup \{x\} \end{aligned}$$

where we say that ϕ is *closed* if $\mathit{fi}(\phi) = \emptyset$, *open* otherwise.

In order to assign meaning to open formulas, environments are employed to indicate what events the free identifiers are bound to. Given a configuration structure $\mathsf{C} = (\mathcal{E}, \mathcal{C}, \mathcal{A}, l)$, an *environment* is a partial function $\rho : \mathcal{I} \rightharpoonup \mathcal{E}$. Given $X \in \mathcal{C}$ and $\phi \in \mathcal{L}_{\rm EI}$, we say that ρ is a *permissible* environment for X and ϕ iff ρ maps every free identifier in ϕ to an event in X. Denoting with $\mathit{dom}(\rho)$ the domain of ρ, $\mathit{rge}(\rho)$ the codomain of ρ, and ρ_ϕ the restriction $\rho \upharpoonright \mathit{fi}(\phi)$, permissibility is formalized as $\mathit{fi}(\phi) \subseteq \mathit{dom}(\rho)$ and $\mathit{rge}(\rho_\phi) \subseteq X$. The set of permissible environments for X and ϕ is indicated by $\mathit{pe}(X, \phi)$.

The satisfaction relation $\models \subseteq (\mathcal{C} \times \mathcal{E}^{\mathcal{I}}) \times \mathcal{L}_{\rm EI}$, with $\mathcal{E}^{\mathcal{I}}$ being the set of functions from $\mathcal{I}$ to $\mathcal{E}$, i.e., the set of environments, is defined by induction on the syntactical structure of $\phi \in \mathcal{L}_{\rm EI}$ as follows:

$$\begin{array}{lll} X \models_\rho \mathsf{true} & & \\ X \models_\rho \neg \phi' & \text{iff} & X \not\models_\rho \phi' \\ X \models_\rho \phi_1 \wedge \phi_2 & \text{iff} & X \models_\rho \phi_1 \text{ and } X \models_\rho \phi_2 \\ X \models_\rho \langle x : a \rangle\!\rangle \phi' & \text{iff} & \text{there is } X \xrightarrow{l(e)}_{\mathsf{C}} X' \text{ such that } l(e) = a \text{ and } X' \models_{\rho[x \mapsto e]} \phi' \\ X \models_\rho (x : a) \phi' & \text{iff} & \text{there is } e \in X \text{ such that } l(e) = a \text{ and } X \models_{\rho[x \mapsto e]} \phi' \\ X \models_\rho \langle\!\langle x \rangle \phi' & \text{iff} & \text{there is } X' \xrightarrow{l(e)}_{\mathsf{C}} X \text{ such that } \rho(x) = e \text{ and } X' \models_\rho \phi' \end{array}$$

where it is understood that the environment in the subscript of every occurrence of $\models$ is permissible for the configuration on the left and the formula on the right. Moreover, $\rho[x \mapsto e]$ is $\rho \setminus \{(x, \rho(x))\} \cup \{(x, e)\}$ if $x \in \mathit{dom}(\rho)$, $\rho \cup \{(x, e)\}$ otherwise.

Let $\mathcal{L}_{\rm EI}^{\rm c}$ be the set of closed formulas of $\mathcal{L}_{\rm EI}$. Given $\phi \in \mathcal{L}_{\rm EI}^{\rm c}$, we write $X \models \phi$ as a shorthand for $X \models_\emptyset \phi$ and $\mathsf{C} \models \phi$ as a shorthand for $\emptyset \models \phi$. Image finiteness means no configuration has infinitely many transitions with the same label.

Theorem 1 ([33]). *Let $\mathsf{C}_i = (\mathcal{E}_i, \mathcal{C}_i, \mathcal{A}, l_i)$, $i \in \{1, 2\}$, be two image-finite stable configuration structures. Then $\mathsf{C}_1 \sim_{\rm HHPB} \mathsf{C}_2$ iff $\forall \phi \in \mathcal{L}_{\rm EI}^{\rm c}.\, \mathsf{C}_1 \models \phi \iff \mathsf{C}_2 \models \phi$.* ■

3 Characterization on Stable Configuration Structures

The first characterization that we provide for $\sim_{\rm HHPB}$ is on stable configuration structures. From a ternary bisimulation relation we move to a binary one where, instead of stepwise building a labeling- and causality-preserving bijection between the events of matching configurations – which are the events executed so far in both stable configuration structures – we just count the identically labeled incoming transitions of matching configurations. Given a configuration X, its *backward ready multiset* is defined as $brm(X) = \{\!| a \in \mathcal{A} \mid X' \xrightarrow{a}_{\mathsf{C}} X |\!\}$ where $\{\!|$ and $|\!\}$ are multiset delimiters. We thus decorate the resulting forward-reverse bisimilarity with the acronym brm, standing for backward ready multiset.

Definition 3. *We say that two stable configuration structures $\mathsf{C}_i = (\mathcal{E}_i, \mathcal{C}_i, \mathcal{A}, l_i)$, $i \in \{1, 2\}$, are* brm-forward-reverse bisimilar*, written $\mathsf{C}_1 \sim_{\mathrm{FRB:brm}} \mathsf{C}_2$, iff there exists a brm-forward-reverse bisimulation between C_1 and C_2, i.e., a relation $\mathcal{B} \subseteq \mathcal{C}_1 \times \mathcal{C}_2$ such that $(\emptyset, \emptyset) \in \mathcal{B}$ and, whenever $(X_1, X_2) \in \mathcal{B}$, then:*

- *For each $X_1 \xrightarrow{a}_{\mathsf{C}_1} X_1'$ there exists $X_2 \xrightarrow{a}_{\mathsf{C}_2} X_2'$ such that $(X_1', X_2') \in \mathcal{B}$, and vice versa.*
- *For each $X_1' \xrightarrow{a}_{\mathsf{C}_1} X_1$ there exists $X_2' \xrightarrow{a}_{\mathsf{C}_2} X_2$ such that $(X_1', X_2') \in \mathcal{B}$, and vice versa.*
- *$brm(X_1) = brm(X_2)$.* ■

Theorem 2. *Let $\mathsf{C}_i = (\mathcal{E}_i, \mathcal{C}_i, \mathcal{A}, l_i)$, $i \in \{1, 2\}$, be two stable configuration structures. Then $\mathsf{C}_1 \sim_{\mathrm{HHPB}} \mathsf{C}_2$ iff $\mathsf{C}_1 \sim_{\mathrm{FRB:brm}} \mathsf{C}_2$.* ■

4 Operational Characterization

The second characterization that we provide for $\sim_{\mathrm{HHPB}}$ is operational. More precisely, we present a variant of the syntax (Section 4.1) and the proved operational semantics (Section 4.2) of the reversible process calculus of [9,8], followed by a redefinition of brm-forward-reverse bisimilarity on that variant along with a modal logic characterization (Section 4.3). Then we develop a denotational semantics for the modified calculus based on stable configuration structures (Section 4.4), so as to import the notion of hereditary history-preserving bisimilarity. Finally, we prove that the stable configuration structures associated with two processes are hereditary history-preserving bisimilar iff the two processes are brm-forward-reverse bisimilar (Section 4.5).

4.1 Syntax of Reversible Concurrent Processes

In the representation of a process, we are used to describe only its future behavior. However, in order to support reversibility in the style of [30], we need to equip the syntax with information about the past, in particular the actions that have already been executed. Taking inspiration from CCS [29] and CSP [12], given a countable set $\mathcal{A}$ of actions including an unobservable action denoted by τ, we extend as follows the syntax for reversible concurrent processes of [9,8]:

$$P ::= \underline{0} \mid a \,.\, P \mid a^{\dagger\xi} .\, P \mid P + P \mid P \parallel_L P$$
$$\xi ::= \varepsilon \mid \langle \theta, \theta \rangle_L$$

where $a \in \mathcal{A}$, $L \subseteq \mathcal{A} \setminus \{\tau\}$, ε is the empty string, θ is a proof term (its syntax will be provided in Section 4.2), and:

- $\underline{0}$ is the terminated process.
- $a \,.\, P$ is a process that can execute action a and whose forward continuation is P (unexecuted action prefix).
- $a^{\dagger\xi} .\, P$ is a process that executed action a and whose forward continuation is inside P, which can undo action a after all executed actions within P have been undone (executed action prefix).
- $P_1 + P_2$ expresses a nondeterministic choice between P_1 and P_2 as far as neither has executed any action yet, otherwise only the one that was selected in the past can move (past-sensitive alternative composition).
- $P_1 \,\|_L\, P_2$ expresses that P_1 and P_2 proceed independently of each other on actions in $\overline{L} = \mathcal{A} \setminus L$, while they have to synchronize on every action in L (parallel composition).

We can characterize two important classes of processes via as many predicates. Firstly, we define *initial* processes, in which all actions are unexecuted and hence no †-decoration appears:

$$\begin{array}{rl} \mathit{init}(\underline{0}) & \\ \mathit{init}(a \,.\, P) & \text{if } \mathit{init}(P) \\ \mathit{init}(P_1 + P_2) & \text{if } \mathit{init}(P_1) \wedge \mathit{init}(P_2) \\ \mathit{init}(P_1 \,\|_L\, P_2) & \text{if } \mathit{init}(P_1) \wedge \mathit{init}(P_2) \end{array}$$

Secondly, we define *well-formed* processes, whose set we denote by $\mathcal{P}$, in which both unexecuted and executed actions can occur in certain circumstances:

$$\begin{array}{rl} \mathit{wf}(\underline{0}) & \\ \mathit{wf}(a \,.\, P) & \text{if } \mathit{init}(P) \\ \mathit{wf}(a^{\dagger\xi} .\, P) & \text{if } \mathit{wf}(P) \\ \mathit{wf}(P_1 + P_2) & \text{if } (\mathit{wf}(P_1) \wedge \mathit{init}(P_2)) \vee (\mathit{init}(P_1) \wedge \mathit{wf}(P_2)) \\ \mathit{wf}(P_1 \,\|_L\, P_2) & \text{if } \mathit{wf}(P_1) \wedge \mathit{wf}(P_2) \end{array}$$

Well formedness not only imposes that every unexecuted action is followed by an initial process, but also that in every alternative composition at least one subprocess is initial. Multiple paths may arise in the presence of both alternative and parallel compositions. However, at each occurrence of the former, only the subprocess chosen for execution can move. Although not selected, the other subprocess is kept as an initial subprocess within the overall process, in the same way as executed actions are kept inside the syntax [11,30], so as to support reversibility. As an example, in $a^{\dagger} .\, b \,.\, \underline{0} + c \,.\, d \,.\, \underline{0}$ the subprocess $c \,.\, d \,.\, \underline{0}$ cannot move because a was selected in the choice between a and c.

It is worth noting that:

- $\underline{0}$ is both initial and well-formed.
- Any initial process is well-formed too.
- $\mathcal{P}$ also contains processes that are not initial like, e.g., $a^{\dagger} .\, b \,.\, \underline{0}$, which can either do b or undo a.
- In $\mathcal{P}$ the relative positions of already executed actions and actions to be executed matter. Precisely, an action of the former kind can never occur after one of the latter kind. For instance, $a^{\dagger} .\, b \,.\, \underline{0} \in \mathcal{P}$ whereas $b \,.\, a^{\dagger} .\, \underline{0} \notin \mathcal{P}$.
- In $\mathcal{P}$ the subprocesses of an alternative composition can be both initial, but cannot be both non-initial. For example, $a \,.\, \underline{0} + b \,.\, \underline{0} \in \mathcal{P}$ while $a^{\dagger} .\, \underline{0} + b^{\dagger} .\, \underline{0} \notin \mathcal{P}$.

Sometimes we will need to bring a process back to its initial version. This is accomplished by removing all †-decorations through function $\mathit{to\_init} : \mathcal{P} \rightarrow \mathcal{P}_{\mathrm{init}}$ with $\mathcal{P}_{\mathrm{init}}$ being the set of initial processes of $\mathcal{P}$, which is defined as follows:

$$\begin{array}{rcll} \mathit{to\_init}(P) & = & P & \text{if } \mathit{init}(P) \\ \mathit{to\_init}(a^{\dagger\xi} . P') & = & a \,.\, \mathit{to\_init}(P') & \\ \mathit{to\_init}(P_1 + P_2) & = & \mathit{to\_init}(P_1) + \mathit{to\_init}(P_2) & \text{if } \neg\mathit{init}(P_1) \vee \neg\mathit{init}(P_2) \\ \mathit{to\_init}(P_1 \parallel_L P_2) & = & \mathit{to\_init}(P_1) \parallel_L \mathit{to\_init}(P_2) & \text{if } \neg\mathit{init}(P_1) \vee \neg\mathit{init}(P_2) \end{array}$$

4.2 Proved Operational Semantics

According to [30] dynamic operators such as action prefix and alternative composition have to be made static in the operational semantic rules, so as to retain within the syntax all the information needed to enable reversibility. Unlike [30] we do not generate a forward transition relation and a backward one, but a single transition relation that we deem to be symmetric in order to enforce the *loop property* [13]: every executed action can be undone and every undone action can be redone. A backward transition from P' to P is subsumed by the corresponding forward transition t from P to P'. As already done in Sections 2 and 3 as well as in [17,6], we will view t as an *outgoing* transition of P when going forward, while we will view t as an *incoming* transition of P' when going backward.

Following [8] we provide an operational semantics based on [18], which is very concrete as every transition is labeled with a *proof term* [10,11]. This is an action preceded by the sequence of operator symbols in the scope of which the action occurs inside the source process of the transition. In the case of a binary operator, the corresponding symbol also specifies whether the action occurs to the left or to the right. The syntax that we adopt for the set Θ of proof terms is the following where $a \in \mathcal{A}$ and $L \subseteq \mathcal{A} \setminus \{\tau\}$:

$$\theta \ ::= \ a \mid ._a\theta \mid \dot{+}\theta \mid +\!\!.\,\theta \mid \lfloor\!\lfloor_L \theta \mid \rfloor\!\rfloor_L \theta \mid \langle \theta, \theta \rangle_L$$

The proved operational semantic rules are in Table 1 and generate the proved labeled transition system $(\mathcal{P}, \Theta, \longrightarrow)$ where $\longrightarrow \ \subseteq \mathcal{P} \times \Theta \times \mathcal{P}$ is the proved transition relation. We denote by $\mathbb{P} \subsetneq \mathcal{P}$ the set of processes that are *reachable* from an initial one via $\longrightarrow$. Not all well-formed processes are reachable; for example, $a^{\dagger} . \underline{0} \parallel_{\{a\}} \underline{0}$ is not reachable from $a \,.\, \underline{0} \parallel_{\{a\}} \underline{0}$ as action a on the left cannot synchronize with any action on the right. From now on we consider only $\mathbb{P}$ and denote by $\mathbb{P}_{\mathrm{init}}$ the subset of its initial processes. Every process in $\mathbb{P}$ may have several outgoing transitions and, if it is not initial, has at least one incoming transition.

The first rule for action prefix (Act$_{\mathrm{f}}$ where f stands for forward) applies only if P is initial and retains the executed action in the target process of the generated forward transition by decorating the action itself with †. The second rule (Act$_{\mathrm{p}}$ where p stands for propagation) propagates actions of inner initial subprocesses by putting an a-dot before them in the label for each outer executed a-action prefix that is encountered.

In both rules for alternative composition (Cho$_{\mathrm{l}}$ and Cho$_{\mathrm{r}}$ where l stands for left and r stands for right), the subprocess that has not been selected for

$(\textsc{Act}_\mathrm{f})\ \dfrac{init(P)}{a\,.\,P \xrightarrow{a} a^{\dagger}\,.\,P}$	$(\textsc{Act}_\mathrm{p})\ \dfrac{P \xrightarrow{\theta} P'}{a^{\dagger\xi}\,.\,P \xrightarrow{\cdot_a \theta} a^{\dagger\xi}\,.\,P'}$
$(\textsc{Cho}_\mathrm{l})\ \dfrac{P_1 \xrightarrow{\theta} P_1' \quad init(P_2)}{P_1 + P_2 \xrightarrow{{}_{\cdot}{+}\theta} P_1' + P_2}$	$(\textsc{Cho}_\mathrm{r})\ \dfrac{P_2 \xrightarrow{\theta} P_2' \quad init(P_1)}{P_1 + P_2 \xrightarrow{{+}_{\cdot}\theta} P_1 + P_2'}$
$(\textsc{Par}_\mathrm{l})\ \dfrac{P_1 \xrightarrow{\theta} P_1' \quad act(\theta) \notin L}{P_1 \,\|_L\, P_2 \xrightarrow{\lfloor\!\|_L \theta} P_1' \,\|_L\, P_2}$	$(\textsc{Par}_\mathrm{r})\ \dfrac{P_2 \xrightarrow{\theta} P_2' \quad act(\theta) \notin L}{P_1 \,\|_L\, P_2 \xrightarrow{\|\!\rfloor_L \theta} P_1 \,\|_L\, P_2'}$
$(\textsc{Syn})\ \dfrac{P_1 \xrightarrow{\theta_1} P_1' \quad P_2 \xrightarrow{\theta_2} P_2' \quad act(\theta_1) = act(\theta_2) \in L}{P_1 \,\|_L\, P_2 \xrightarrow{\langle \theta_1, \theta_2 \rangle_L} enr(P_1' \,\|_L\, P_2', \langle \theta_1, \theta_2 \rangle_L)}$	

Table 1. Proved operational semantic rules for reversible concurrent processes

execution is retained as an initial subprocess in the target process of the generated transition. When both subprocesses are initial, both rules for alternative composition are applicable, otherwise only one of them can be applied and in that case it is the non-initial subprocess that can move, because the other one has been discarded at the moment of the selection. The symbol ${}_{\cdot}{+}$ or ${+}_{\cdot}$ is added at the beginning of the proof term.

Due to the †-decorations of executed actions inside the process syntax, over the set $\mathbb{P}_{\mathrm{seq}}$ of *sequential* processes – in which there are no occurrences of parallel composition – every non-initial process has exactly one incoming transition, proved labeled transition systems turn out to be trees, and well formedness coincides with reachability [9].

Example 1. The proved labeled transition system underlying the initial sequential process $a\,.\,\underline{0}$ has a single transition $a\,.\,\underline{0} \xrightarrow{a} a^{\dagger}\,.\,\underline{0}$. In contrast, the proved labeled transition system underlying the initial sequential process $a\,.\,\underline{0} + a\,.\,\underline{0}$ has the two transitions $a\,.\,\underline{0} + a\,.\,\underline{0} \xrightarrow{{}_{\cdot}{+}a} a^{\dagger}\,.\,\underline{0} + a\,.\,\underline{0}$ and $a\,.\,\underline{0} + a\,.\,\underline{0} \xrightarrow{{+}_{\cdot}a} a\,.\,\underline{0} + a^{\dagger}\,.\,\underline{0}$. Note that the two target processes are different from each other due to the presence of action decorations, whereas a single a-transition from $a\,.\,\underline{0} + a\,.\,\underline{0}$ to $\underline{0}$ would be generated in the setting of a forward-only process calculus. ■

The three rules for parallel composition use partial function $act : \varTheta \rightharpoonup \mathcal{A}$ to extract an action from a proof term θ. This function, which will be used throughout the paper, is defined by induction on the syntactical structure of θ as follows:

$$\begin{aligned} act(a) &= a \\ act(\cdot_a \theta') &= act(\theta') \\ act({}_{\cdot}{+}\theta') = act({+}_{\cdot}\theta') &= act(\theta') \\ act(\lfloor\!\|_L \theta') = act(\|\!\rfloor_L \theta') &= act(\theta') \\ act(\langle \theta_1, \theta_2 \rangle_L) &= \begin{cases} act(\theta_1) & \text{if } act(\theta_1) = act(\theta_2) \\ \text{undefined} & \text{otherwise} \end{cases} \end{aligned}$$

In the first two rules ($\textsc{Par}_\mathrm{l}$ and $\textsc{Par}_\mathrm{r}$), a single subprocess proceeds by performing an action not belonging to L, with $\rfloor\!\rfloor_L$ or $\lfloor\!\lfloor_L$ being placed at the beginning of the proof term. In the third rule (Syn), both subprocesses synchronize on an action in L and the resulting proof term contains both individual proof terms. If $L = \emptyset$ or $L = \mathcal{A} \setminus \{\tau\}$, then the two subprocesses are fully independent or fully synchronized, respectively, on observable actions.

The natural target process $P'_1 \parallel_L P'_2$ of a synchronization has to be suitably manipulated in rule Syn to correctly reflect causality and concurrency. More precisely, the †-decoration of every executed action participating in the synchronization has to be enriched with a proof term of the form $\langle \theta_1, \theta_2 \rangle_L$. This is accomplished by taking $\mathit{enr}(P'_1 \parallel_L P'_2, \langle \theta_1, \theta_2 \rangle_L) = \mathit{enr}'(P'_1 \parallel_L P'_2, \langle \theta_1, \theta_2 \rangle_L, \langle \theta_1, \theta_2 \rangle_L)$ as target process, where partial function $\mathit{enr}' : \mathbb{P} \times \Theta \times \Theta \rightharpoonup \mathbb{P}$ is defined by induction on the syntactical structure of its first argument $P \in \mathbb{P}$ as follows:

$$\begin{aligned}
\mathit{enr}'(\underline{0}, \theta, \bar{\theta}) &= \underline{0} \\
\mathit{enr}'(a \,.\, P', \theta, \bar{\theta}) &= \text{undefined} \\
\mathit{enr}'(a^{\dagger \xi} \,.\, P', \theta, \bar{\theta}) &= \begin{cases} a^{\dagger \bar{\theta}} \,.\, P' & \text{if } \theta = a \\ a^{\dagger \xi} \,.\, \mathit{enr}'(P', \theta', \bar{\theta}) & \text{if } \theta = \,._a \theta' \\ \text{undefined} & \text{otherwise} \end{cases} \\
\mathit{enr}'(P_1 + P_2, \theta, \bar{\theta}) &= \begin{cases} \mathit{enr}'(P_1, \theta', \bar{\theta}) + P_2 & \text{if } \theta = +\!\!\rfloor \theta' \\ P_1 + \mathit{enr}'(P_2, \theta', \bar{\theta}) & \text{if } \theta = \lfloor\!\!+ \theta' \\ \text{undefined} & \text{otherwise} \end{cases} \\
\mathit{enr}'(P_1 \parallel_L P_2, \theta, \bar{\theta}) &= \begin{cases} \mathit{enr}'(P_1, \theta', \bar{\theta}) \parallel_L P_2 & \text{if } \theta = \rfloor\!\rfloor_L \theta' \\ P_1 \parallel_L \mathit{enr}'(P_2, \theta', \bar{\theta}) & \text{if } \theta = \lfloor\!\lfloor_L \theta' \\ \mathit{enr}'(P_1, \theta_1, \bar{\theta}) \parallel_L \mathit{enr}'(P_2, \theta_2, \bar{\theta}) & \text{if } \theta = \langle \theta_1, \theta_2 \rangle_L \\ \text{undefined} & \text{otherwise} \end{cases}
\end{aligned}$$

Example 2. The proved labeled transition system underlying the initial process $(a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}) \parallel_{\{a\}} a \,.\, a \,.\, \underline{0}$, which is the synchronization of autoconcurrency with autocausation, has the following two maximal transition sequences:

- $(a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}) \parallel_{\{a\}} a \,.\, a \,.\, \underline{0}$
 $\xrightarrow{\langle \rfloor\!\rfloor_\emptyset a, a \rangle_{\{a\}}} (a^{\dagger \langle \rfloor\!\rfloor_\emptyset a, a \rangle_{\{a\}}} \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}) \parallel_{\{a\}} a^{\dagger \langle \rfloor\!\rfloor_\emptyset a, a \rangle_{\{a\}}} \,.\, a \,.\, \underline{0}$
 $\xrightarrow{\langle \lfloor\!\lfloor_\emptyset a, ._a a \rangle_{\{a\}}} (a^{\dagger \langle \rfloor\!\rfloor_\emptyset a, a \rangle_{\{a\}}} \,.\, \underline{0} \parallel_\emptyset a^{\dagger \langle \lfloor\!\lfloor_\emptyset a, ._a a \rangle_{\{a\}}} \,.\, \underline{0}) \parallel_{\{a\}} a^{\dagger \langle \rfloor\!\rfloor_\emptyset a, a \rangle_{\{a\}}} \,.\, a^{\dagger \langle \lfloor\!\lfloor_\emptyset a, ._a a \rangle_{\{a\}}} \,.\, \underline{0}$
- $(a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}) \parallel_{\{a\}} a \,.\, a \,.\, \underline{0}$
 $\xrightarrow{\langle \lfloor\!\lfloor_\emptyset a, a \rangle_{\{a\}}} (a \,.\, \underline{0} \parallel_\emptyset a^{\dagger \langle \lfloor\!\lfloor_\emptyset a, a \rangle_{\{a\}}} \,.\, \underline{0}) \parallel_{\{a\}} a^{\dagger \langle \lfloor\!\lfloor_\emptyset a, a \rangle_{\{a\}}} \,.\, a \,.\, \underline{0}$
 $\xrightarrow{\langle \rfloor\!\rfloor_\emptyset a, ._a a \rangle_{\{a\}}} (a^{\dagger \langle \rfloor\!\rfloor_\emptyset a, ._a a \rangle_{\{a\}}} \,.\, \underline{0} \parallel_\emptyset a^{\dagger \langle \lfloor\!\lfloor_\emptyset a, a \rangle_{\{a\}}} \,.\, \underline{0}) \parallel_{\{a\}} a^{\dagger \langle \lfloor\!\lfloor_\emptyset a, a \rangle_{\{a\}}} \,.\, a^{\dagger \langle \rfloor\!\rfloor_\emptyset a, ._a a \rangle_{\{a\}}} \,.\, \underline{0}$

Note that the target processes of the two sequences are different thanks to the different additional decorations of the pairs of synchronizing executed actions. Without those decorations, the two sequences would end up in the same process $(a^\dagger \,.\, \underline{0} \parallel_\emptyset a^\dagger \,.\, \underline{0}) \parallel_{\{a\}} a^\dagger \,.\, a^\dagger \,.\, \underline{0}$ – thus yielding a diamond-shaped transition system – which would not reflect the fact that the two executed a-actions in $a^\dagger \,.\, a^\dagger \,.\, \underline{0}$ cannot be undone in any order as the first one causes the second one. ■

Example 3. The proved labeled transition system underlying the initial process $(a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}) \parallel_{\{a\}} (a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0})$, which is the synchronization of autoconcurrency with itself, has the following four maximal transition sequences:

- $(a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0}) \parallel_{\{a\}} (a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0})$
$$\xrightarrow{\langle \rfloor\!\rfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}} (a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0}) \parallel_{\{a\}} (a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0})$$
$$\xrightarrow{\langle \lfloor\!\lfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}} (a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0}) \parallel_{\{a\}} (a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0})$$
- $(a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0}) \parallel_{\{a\}} (a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0})$
$$\xrightarrow{\langle \lfloor\!\lfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}} (a\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0}) \parallel_{\{a\}} (a\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0})$$
$$\xrightarrow{\langle \rfloor\!\rfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}} (a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0}) \parallel_{\{a\}} (a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0})$$
- $(a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0}) \parallel_{\{a\}} (a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0})$
$$\xrightarrow{\langle \rfloor\!\rfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}} (a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0}) \parallel_{\{a\}} (a\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0})$$
$$\xrightarrow{\langle \lfloor\!\lfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}} (a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0}) \parallel_{\{a\}} (a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0})$$
- $(a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0}) \parallel_{\{a\}} (a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0})$
$$\xrightarrow{\langle \lfloor\!\lfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}} (a\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0}) \parallel_{\{a\}} (a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0})$$
$$\xrightarrow{\langle \rfloor\!\rfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}} (a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0}) \parallel_{\{a\}} (a^{\dagger \langle \lfloor\!\lfloor_{\emptyset} a, \rfloor\!\rfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger \langle \rfloor\!\rfloor_{\emptyset} a, \lfloor\!\lfloor_{\emptyset} a \rangle_{\{a\}}}\,.\,\underline{0})$$

While the target processes of the first (resp. last) two sequences are equal, the target process of the first two sequences is different from the one of the last two sequences due to the different additional decorations of the pairs of synchronizing executed actions. This results in a double-diamond-shaped transition system like the one of $(a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0}) + (a\,.\,\underline{0} \parallel_{\emptyset} a\,.\,\underline{0})$. Without those decorations, the four sequences would end up in the same process $(a^{\dagger}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger}\,.\,\underline{0}) \parallel_{\{a\}} (a^{\dagger}\,.\,\underline{0} \parallel_{\emptyset} a^{\dagger}\,.\,\underline{0})$, thus yielding a single-diamond-shaped transition system. ■

4.3 Forward-Reverse Bisimilarity and Backward Ready Multisets

We now redefine brm-forward-reverse bisimilarity over $\mathbb{P}$. Unlike stable configuration structures, for processes we can syntactically construct their (finite) backward ready multisets, intended as the multisets of actions occurring in the labels of their incoming transitions. In the following we use $\oplus$ for multiset union, which adds multiplicities of identical elements, and $\otimes$ for multiset intersection, which multiplies the multiplicities of those elements. The *backward ready multiset* of $P \in \mathbb{P}$ is inductively defined as follows where $\overline{L} = \mathcal{A} \setminus L$:

$$\begin{aligned}
brm(\underline{0}) &= \emptyset \\
brm(a\,.\,P') &= \emptyset \\
brm(a^{\dagger \xi}.\,P') &= \begin{cases} \{\!|\, a \,|\!\} & \text{if } init(P') \\ brm(P') & \text{if } \neg init(P') \end{cases} \\
brm(P_1 + P_2) &= \begin{cases} \emptyset & \text{if } init(P_1) \wedge init(P_2) \\ brm(P_1) & \text{if } \neg init(P_1) \wedge init(P_2) \\ brm(P_2) & \text{if } init(P_1) \wedge \neg init(P_2) \end{cases} \\
brm(P_1 \parallel_L P_2) &= (brm(P_1) \otimes \overline{L}) \oplus (brm(P_2) \otimes \overline{L}) \oplus (brm(P_1) \otimes brm(P_2) \otimes L)
\end{aligned}$$

The first two clauses stated below are the same as the ones of forward-reverse bisimilarity $\sim_{\mathrm{FRB}}$ over a single transition relation defined in [9,8]. Note the use of function *act* to abstract from operator symbols inside transition labels.

Definition 4. *We say that $P_1, P_2 \in \mathbb{P}$ are* brm-forward-reverse bisimilar*, written $P_1 \sim_{\mathrm{FRB:brm}} P_2$, iff P_1 and P_2 are related by a brm-forward-reverse bisimulation, i.e., a symmetric relation $\mathcal{B}$ over $\mathbb{P}$ such that, whenever $(Q_1, Q_2) \in \mathcal{B}$, then:*

- *For each $Q_1 \xrightarrow{\theta_1} Q'_1$ there exists $Q_2 \xrightarrow{\theta_2} Q'_2$ such that $act(\theta_1) = act(\theta_2)$ and $(Q'_1, Q'_2) \in \mathcal{B}$.*
- *For each $Q'_1 \xrightarrow{\theta_1} Q_1$ there exists $Q'_2 \xrightarrow{\theta_2} Q_2$ such that $act(\theta_1) = act(\theta_2)$ and $(Q'_1, Q'_2) \in \mathcal{B}$.*
- *$brm(Q_1) = brm(Q_2)$.* ■

Example 4. $a\,.\,\underline{0} \parallel_\emptyset a\,.\,\underline{0} \not\sim_{\mathrm{FRB:brm}} a\,.\,a\,.\,\underline{0}$ as in the forward bisimulation game they respectively reach $a^\dagger.\,\underline{0} \parallel_\emptyset a^\dagger\,.\,\underline{0}$ and $a^\dagger.\,a^\dagger.\,\underline{0}$ after performing two a-transitions, where $brm(a^\dagger.\,\underline{0} \parallel_\emptyset a^\dagger\,.\,\underline{0}) = \{\!|\, a, a \,|\!\} \neq \{\!|\, a \,|\!\} = brm(a^\dagger.\,a^\dagger.\,\underline{0})$. Likewise, $(a\,.\,\underline{0} \parallel_\emptyset a\,.\,\underline{0}) \parallel_{\{a\}} a\,.\,a\,.\,\underline{0} \not\sim_{\mathrm{FRB:brm}} (a\,.\,\underline{0} \parallel_\emptyset a\,.\,\underline{0})$. In contrast, $(a\,.\,\underline{0} \parallel_\emptyset a\,.\,\underline{0}) \parallel_{\{a\}} (a\,.\,\underline{0} \parallel_\emptyset a\,.\,\underline{0}) \sim_{\mathrm{FRB:brm}} (a\,.\,\underline{0} \parallel_\emptyset a\,.\,\underline{0}) + (a\,.\,\underline{0} \parallel_\emptyset a\,.\,\underline{0}) \sim_{\mathrm{FRB:brm}} a\,.\,\underline{0} \parallel_\emptyset a\,.\,\underline{0}$. ■

An axiomatization of $\sim_{\mathrm{FRB:brm}}$ can be derived from the one of $\sim_{\mathrm{FRB}}$ in [8] by using backward ready multisets instead of backward ready sets when extending action prefixes at process encoding time. We conclude this section by developing a modal logic characterization for $\sim_{\mathrm{FRB:brm}}$ inspired by the one of $\sim_{\mathrm{FRB}}$ in [7].

The set $\mathcal{L}_{\mathrm{BRM}}$ of formulas of the *backward ready multiset logic* is generated by the following syntax:

$$\phi \ ::= \ \mathsf{true} \mid M \mid \neg\phi \mid \phi \wedge \phi \mid \langle a \rangle \phi \mid \langle a^\dagger \rangle \phi$$

where $M : \mathcal{A} \to \mathbb{N}$ and $a \in \mathcal{A}$. The satisfaction relation $\models \subseteq \mathbb{P} \times \mathcal{L}_{\mathrm{BRM}}$ is defined by induction on the syntactical structure of $\phi \in \mathcal{L}_{\mathrm{BRM}}$ as follows:

$$\begin{array}{ll}
P \models \mathsf{true} & \\
P \models M & \text{iff } brm(P) = M \\
P \models \neg\phi' & \text{iff } P \not\models \phi' \\
P \models \phi_1 \wedge \phi_2 & \text{iff } P \models \phi_1 \text{ and } P \models \phi_2 \\
P \models \langle a \rangle \phi' & \text{iff there exists } P \xrightarrow{\theta} P' \text{ such that } act(\theta) = a \text{ and } P' \models \phi' \\
P \models \langle a^\dagger \rangle \phi' & \text{iff there exists } P' \xrightarrow{\theta} P \text{ such that } act(\theta) = a \text{ and } P' \models \phi'
\end{array}$$

Note that every $P \in \mathbb{P}$ is image finite, i.e., it has finitely many outgoing (and incoming) transitions labeled with proof terms containing the same action.

Theorem 3. *Let $P_1, P_2 \in \mathbb{P}$. Then $P_1 \sim_{\mathrm{FRB:brm}} P_2$ iff $\forall \phi \in \mathcal{L}_{\mathrm{BRM}}.\, P_1 \models \phi \Longleftrightarrow P_2 \models \phi$.* ■

4.4 Denotational Semantics on Stable Configuration Structures

To enable a comparison between hereditary history-preserving bisimilarity over stable configuration structures and brm-forward-reverse bisimilarity over processes, we proceed with the introduction of a denotational semantics for $\mathbb{P}$ based on stable configuration structures. The first step consists of redefining the process operators of Section 4.1 over stable configuration structures. Taking inspiration from [11], we do this by using proof terms in Θ to formalize events:

- The terminated stable configuration structure N is defined as $(\emptyset, \{\emptyset\}, \mathcal{A}, \emptyset)$.
- Let $a \in \mathcal{A}$ and $\mathsf{C} = (\mathcal{E}, \mathcal{C}, \mathcal{A}, l)$ be a stable configuration structure such that $\mathcal{E} \subseteq \Theta$. The action prefix $a \,.\, \mathsf{C}$ is defined as $(\mathcal{E}', \mathcal{C}', \mathcal{A}, l')$ where:
  - $\mathcal{E}' \,=\, \{a\} \cup \{{\cdot}_a\theta \mid \theta \in \mathcal{E}\}$.
  - $\mathcal{C}' \,=\, \{\emptyset\} \cup \{X' \in \mathcal{P}_{\mathrm{fin}}(\mathcal{E}') \mid \exists X \in \mathcal{C}.\, X' = \{a\} \cup \{{\cdot}_a\theta \mid \theta \in X\}\}$.
  - $l' \,=\, \{(a, a)\} \cup \{({\cdot}_a\theta, \mathit{act}({\cdot}_a\theta)) \mid \exists X \in \mathcal{C}.\, \theta \in X\}$.
- Let $\mathsf{C}_i = (\mathcal{E}_i, \mathcal{C}_i, \mathcal{A}, l_i)$ be a stable configuration structure such that $\mathcal{E}_i \subseteq \Theta$ for $i \in \{1, 2\}$. The alternative composition $\mathsf{C}_1 + \mathsf{C}_2$ is defined as $(\mathcal{E}, \mathcal{C}, \mathcal{A}, l)$ where:
  - $\mathcal{E} \,=\, \{{}_{\cdot}{+}\theta \mid \theta \in \mathcal{E}_1\} \cup \{{+}_{\cdot}\theta \mid \theta \in \mathcal{E}_2\}$.
  - $\mathcal{C} \,=\, \{X \in \mathcal{P}_{\mathrm{fin}}(\mathcal{E}) \mid \exists X_1 \in \mathcal{C}_1.\, X = \{{}_{\cdot}{+}\theta \mid \theta \in X_1\}\} \cup$
    $\{X \in \mathcal{P}_{\mathrm{fin}}(\mathcal{E}) \mid \exists X_2 \in \mathcal{C}_2.\, X = \{{+}_{\cdot}\theta \mid \theta \in X_2\}\}$.
  - $l \,=\, \{({}_{\cdot}{+}\theta, \mathit{act}({}_{\cdot}{+}\theta)) \mid \exists X_1 \in \mathcal{C}_1.\, \theta \in X_1\} \cup$
    $\{({+}_{\cdot}\theta, \mathit{act}({+}_{\cdot}\theta)) \mid \exists X_2 \in \mathcal{C}_2.\, \theta \in X_2\}$.
- Let $\mathsf{C}_i = (\mathcal{E}_i, \mathcal{C}_i, \mathcal{A}, l_i)$ be a stable configuration structure such that $\mathcal{E}_i \subseteq \Theta$ for $i \in \{1, 2\}$ and $L \subseteq \mathcal{A} \setminus \{\tau\}$. The parallel composition $\mathsf{C}_1 \parallel_L \mathsf{C}_2$ is defined as $(\mathcal{E}, \mathcal{C}, \mathcal{A}, l)$ where:
  - $\mathcal{E} \,=\, \{{}_{\cdot}\!\parallel_L\theta \mid \theta \in \mathcal{E}_1 \wedge \mathit{act}(\theta) \notin L\} \cup \{\parallel\!{}_{\cdot L}\theta \mid \theta \in \mathcal{E}_2 \wedge \mathit{act}(\theta) \notin L\} \cup$
    $\{\langle\theta_1, \theta_2\rangle_L \mid \theta_1 \in \mathcal{E}_1 \wedge \theta_2 \in \mathcal{E}_2 \wedge \mathit{act}(\theta_1) = \mathit{act}(\theta_2) \in L\}$.
  - $\mathcal{C} \,=\, \{X \in \mathcal{P}_{\mathrm{fin}}(\mathcal{E}) \mid \mathit{proj}_1(X) \in \mathcal{C}_1 \wedge \mathit{proj}_2(X) \in \mathcal{C}_2 \wedge \forall e, e' \in X.$
    $((\mathit{proj}_1(\{e\}) = \mathit{proj}_1(\{e'\}) \neq \emptyset \vee \mathit{proj}_2(\{e\}) = \mathit{proj}_2(\{e'\}) \neq \emptyset) \Longrightarrow$
    $e = e') \wedge$ [local injectivity of projections]
    $(e \neq e' \Longrightarrow$ [coincidence freeness (a single event per transition)]
    $\exists Y \subseteq X.\, (\mathit{proj}_1(Y) \in \mathcal{C}_1 \wedge \mathit{proj}_2(Y) \in \mathcal{C}_2 \wedge (e \in Y \Longleftrightarrow e' \notin Y)))\}$

    with projections being defined as follows:
    - $\mathit{proj}_1(X) \,=\, \{\theta_1 \in \mathcal{E}_1 \mid {}_{\cdot}\!\parallel_L\theta_1 \in X \vee \exists \theta_2 \in \mathcal{E}_2.\, \langle\theta_1, \theta_2\rangle_L \in X\}$.
    - $\mathit{proj}_2(X) \,=\, \{\theta_2 \in \mathcal{E}_2 \mid \parallel\!{}_{\cdot L}\theta_2 \in X \vee \exists \theta_1 \in \mathcal{E}_1.\, \langle\theta_1, \theta_2\rangle_L \in X\}$.
  - $l \,=\, \{({}_{\cdot}\!\parallel_L\theta, \mathit{act}({}_{\cdot}\!\parallel_L\theta)) \mid \exists X_1 \in \mathcal{C}_1.\, \theta \in X_1 \wedge \mathit{act}(\theta) \notin L\} \cup$
    $\{(\parallel\!{}_{\cdot L}\theta, \mathit{act}(\parallel\!{}_{\cdot L}\theta)) \mid \exists X_2 \in \mathcal{C}_2.\, \theta \in X_2 \wedge \mathit{act}(\theta) \notin L\} \cup$
    $\{(\langle\theta_1, \theta_2\rangle_L, \mathit{act}(\langle\theta_1, \theta_2\rangle_L)) \mid \exists X_1 \in \mathcal{C}_1.\, \exists X_2 \in \mathcal{C}_2.$
    $\theta_1 \in X_1 \wedge \theta_2 \in X_2 \wedge \mathit{act}(\theta_1) = \mathit{act}(\theta_2) \in L\}$.

Then with each process $P \in \mathbb{P}$ we denotationally associate a stable configuration structure semantics in a way similar to [35,3], with the notable difference that we represent events via proof terms. More precisely, each process is given a pair formed by a stable configuration structure, built by using the operators above, and a configuration of that structure. The idea is that all processes reachable from the same initial process share the same configuration structure. In contrast, the designated configuration uniquely identifies the specific process through the proof terms labeling a sequence of proved transitions by means of which the considered process is reached from the initial one.

Note that such a sequence is empty if P is initial – which corresponds to the empty configuration – and unique if P is sequential. In the case that P is neither initial nor sequential, if there are several transition sequences reaching it – meaning that non-synchronizing actions of different parallel subprocesses have been executed – then they result in the same configuration [11], because independent actions can be executed in any order and the order of the elements within a configuration – which is a set – does not matter.

Definition 5. *The* stable configuration structure semantics *of* $P \in \mathbb{P}$ *is the pair* $[\![P]\!] = (\mathsf{C}_P, X_P)$ *where:*

- $\mathsf{C}_P = scs(to\_init(P))$*, with the stable configuration structure* $scs(Q)$ *associated with an initial process* $Q \in \mathbb{P}$ *being defined by induction on the syntactical structure of* Q *as follows:*
  - $scs(\underline{0}) = \mathsf{N}$.
  - $scs(a \,.\, Q') = a \,.\, scs(Q')$.
  - $scs(Q_1 + Q_2) = scs(Q_1) + scs(Q_2)$.
  - $scs(Q_1 \parallel_L Q_2) = scs(Q_1) \parallel_L scs(Q_2)$.
- $X_P = \emptyset$ *if* P *is initial, otherwise* $X_P = \{\theta_i \mid 1 \leq i \leq n\}$ *for some* $n \in \mathbb{N}_{\geq 1}$ *such that there exists* $P_{i-1} \xrightarrow{\theta_i} P_i$ *for all* $1 \leq i \leq n$ *with* $P_0 = to\_init(P)$ *and* $P_n = P$. ■

Example 5. $[\![a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}]\!]$ comprises (see Figure 1(a)):

- The two events $\rfloor\!\parallel_\emptyset a$ and $\lfloor\!\parallel_\emptyset a$.
- The four configurations $\emptyset$, $\{\rfloor\!\parallel_\emptyset a\}$, $\{\lfloor\!\parallel_\emptyset a\}$, and $\{\rfloor\!\parallel_\emptyset a, \lfloor\!\parallel_\emptyset a\}$.
- The two maximal computations $\emptyset \xrightarrow{a}_{\mathsf{C}_{a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}}} \{\rfloor\!\parallel_\emptyset a\} \xrightarrow{a}_{\mathsf{C}_{a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}}} \{\rfloor\!\parallel_\emptyset a, \lfloor\!\parallel_\emptyset a\}$ and $\emptyset \xrightarrow{a}_{\mathsf{C}_{a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}}} \{\lfloor\!\parallel_\emptyset a\} \xrightarrow{a}_{\mathsf{C}_{a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}}} \{\rfloor\!\parallel_\emptyset a, \lfloor\!\parallel_\emptyset a\}$.

In contrast, $[\![a \,.\, a \,.\, \underline{0}]\!]$ comprises (see Figure 1(b)):

- The two events a and $._a a$.
- The three configurations $\emptyset$, $\{a\}$, and $\{a, ._a a\}$.
- The only maximal computation $\emptyset \xrightarrow{a}_{\mathsf{C}_{a \,.\, a \,.\, \underline{0}}} \{a\} \xrightarrow{a}_{\mathsf{C}_{a \,.\, a \,.\, \underline{0}}} \{a, ._a a\}$.

Therefore, $[\![a \,.\, \underline{0} \parallel_\emptyset a \,.\, \underline{0}]\!] \not\sim_{\mathrm{HHPB}} [\![a \,.\, a \,.\, \underline{0}]\!]$ because a causally precedes $._a a$ while $\rfloor\!\parallel_\emptyset a$ and $\lfloor\!\parallel_\emptyset a$ are independent of each other and hence no (labeling- and) causality-preserving bijection would relate the former two events to the latter two. ■

4.5 Operational Characterization Result

We start by establishing a connection between proved transitions of processes and transitions of stable configuration structures associated with processes.

Lemma 1. *Let* $P, P' \in \mathbb{P}$ *and* $\theta \in \Theta$*. Then* $P \xrightarrow{\theta} P'$ *iff* $X_P \xrightarrow{act(\theta)}_{\mathsf{C}_P} X_{P'}$. ■

We are now in a position of proving our operational characterization result.

Theorem 4. *Let* $P_1, P_2 \in \mathbb{P}$*. Then* $[\![P_1]\!] \sim_{\mathrm{HHPB}} [\![P_2]\!]$ *iff* $P_1 \sim_{\mathrm{FRB:brm}} P_2$. ■

5 Relationships between $\mathcal{L}_{\mathrm{EI}}$ and $\mathcal{L}_{\mathrm{BRM}}$

From Theorems 4, 3, and 1 it follows that two processes satisfy the same formulas of $\mathcal{L}_{\mathrm{BRM}}$ iff their associated stable configuration structures satisfy the same formulas of $\mathcal{L}_{\mathrm{EI}}$. It is therefore interesting to investigate the relationships between the two logics. On the one hand, we show how to reinterpret $\mathcal{L}_{\mathrm{EI}}$ over processes (Section 5.1) and $\mathcal{L}_{\mathrm{BRM}}$ over stable configuration structures (Section 5.2). On the other hand, we discuss how to translate $\mathcal{L}_{\mathrm{BRM}}$ into $\mathcal{L}_{\mathrm{EI}}$ (Section 5.3) and vice versa (Section 5.4).

5.1 Reinterpreting $\mathcal{L}_{\mathrm{EI}}$ over $\mathbb{P}$

The only non-trivial case is the one of the binder $(x : a)$. The process analogous of an event in a configuration that is labeled with a certain action is a subprocess starting with an executed occurrence of that action. Indicating with $sp(P)$ the set of all subprocesses of P, let $apt(a^{\dagger}\,.\,P', P)$ be the proof term associated with the execution of action a of the subterm $a^{\dagger}\,.\,P'$ of P. Formally, $apt(a^{\dagger}\,.\,P', P) = \theta$ iff $a^{\dagger}\,.\,P' \in sp(P)$ and there exist $P'', P''' \in \mathbb{P}$ such that $a\,.\,to\_init(P') \in sp(P'')$, $P'' \xrightarrow{\theta} P'''$, $act(\theta) = a$, and $a^{\dagger}\,.\,to\_init(P') \in sp(P''')$.

The satisfaction relation $\models \subseteq (\mathbb{P} \times \Theta^{\mathcal{I}}) \times \mathcal{L}_{\mathrm{EI}}$ is defined by induction on the syntactical structure of $\phi \in \mathcal{L}_{\mathrm{EI}}$ as follows:

$P \models_{\rho} \mathsf{true}$

$P \models_{\rho} \neg\phi'$ iff $P \not\models_{\rho} \phi'$

$P \models_{\rho} \phi_1 \wedge \phi_2$ iff $P \models_{\rho} \phi_1$ and $P \models_{\rho} \phi_2$

$P \models_{\rho} \langle x : a\rangle\rangle\phi'$ iff there is $P \xrightarrow{\theta} P'$ such that $act(\theta) = a$ and $P' \models_{\rho[x \mapsto \theta]} \phi'$

$P \models_{\rho} (x : a)\phi'$ iff there is $a^{\dagger}\,.\,P' \in sp(P)$ such that $P \models_{\rho[x \mapsto apt(a^{\dagger}\,.\,P', P)]} \phi$

$P \models_{\rho} \langle\langle x\rangle\phi'$ iff there is $P' \xrightarrow{\theta} P$ such that $\rho(x) = \theta$ and $P' \models_{\rho} \phi'$

where it is understood that the environment in the subscript of every occurrence of $\models$ is permissible for the configuration identifying (in the associated denotational semantics) the process on the left – e.g., X_P in the case of process P – and the formula on the right.

Theorem 5. *Let $P \in \mathbb{P}$. Then $\forall \phi \in \mathcal{L}_{\mathrm{EI}}. \forall \rho \in pe(X_P, \phi). P \models_{\rho} \phi \iff [\![P]\!] \models_{\rho} \phi$.* ∎

To prove that, consequently, $\mathcal{L}_{\mathrm{EI}}$ reinterpreted over $\mathbb{P}$ characterizes $\sim_{\mathrm{FRB:brm}}$, we follow [33] and hence first show that any substitution of the variables freely occurring in a formula requires a modification of the permissible environment.

Lemma 2. *Let $P \in \mathbb{P}$, $\phi \in \mathcal{L}_{\mathrm{EI}}$, and $\rho \in pe(X_P, \phi)$. Given a substitution σ that – not necessarily injectively – maps $fi(\phi)$ to a set of fresh identifiers that do not occur either free or bound in ϕ, let $\sigma(\phi)$ be the formula obtained from ϕ by replacing each occurrence of $x \in fi(\phi)$ with $\sigma(x)$ and let $\rho^{\sigma} \in pe(X_P, \sigma(\phi))$ be the environment obtained from ρ by replacing each $x \in fi(\phi)$ with $\sigma(x)$. Then $P \models_{\rho} \phi$ iff $P \models_{\rho^{\sigma}} \sigma(\phi)$.* ∎

Corollary 1. *Let $P_1, P_2 \in \mathbb{P}$. Then $P_1 \sim_{\mathrm{FRB:brm}} P_2$ iff $\exists f_{1,2}. \forall \phi \in \mathcal{L}_{\mathrm{EI}}. \forall \rho \in pe(X_{P_1}, \phi). P_1 \models_{\rho} \phi \iff P_2 \models_{f_{1,2} \circ \rho} \phi$ where $f_{1,2}$ is a label-preserving bijection from X_{P_1} to X_{P_2}.* ∎

5.2 Reinterpreting $\mathcal{L}_{\mathrm{BRM}}$ over Stable Configuration Structures

Let us denote by $[\![\mathbb{P}]\!]$ the set of all stable configuration structures – whose events are proof terms – that turn out to be the denotational semantics of some $P \in \mathbb{P}$. Recalling that $[\![P]\!] = (\mathcal{C}_P, X_P)$, when writing $[\![P]\!] \models \phi$ we mean $X_P \models \phi$.

The satisfaction relation $\models \subseteq [\![\mathbb{P}]\!] \times \mathcal{L}_{\mathrm{BRM}}$ is defined by induction on the syntactical structure of $\phi \in \mathcal{L}_{\mathrm{BRM}}$ as follows:

$$\begin{array}{lll}
[\![P]\!] \models \mathsf{true} & & \\
[\![P]\!] \models M & \text{iff} & \{\!| a \in \mathcal{A} \mid X_{P'} \xrightarrow{a}_{\mathcal{C}_{P'}} X_P |\!\} = M \\
[\![P]\!] \models \neg\phi' & \text{iff} & [\![P]\!] \not\models \phi' \\
[\![P]\!] \models \phi_1 \wedge \phi_2 & \text{iff} & [\![P]\!] \models \phi_1 \text{ and } [\![P]\!] \models \phi_2 \\
[\![P]\!] \models \langle a \rangle \phi' & \text{iff} & \text{there exists } X_P \xrightarrow{a}_{\mathcal{C}_P} X_{P'} \text{ such that } [\![P']\!] \models \phi' \\
[\![P]\!] \models \langle a^\dagger \rangle \phi' & \text{iff} & \text{there exists } X_{P'} \xrightarrow{a}_{\mathcal{C}_{P'}} X_P \text{ such that } [\![P']\!] \models \phi'
\end{array}$$

Every process and its associated stable configuration structure satisfy the same formulas of $\mathcal{L}_{\mathrm{BRM}}$. As a consequence, $\mathcal{L}_{\mathrm{BRM}}$ reinterpreted over stable configuration structures characterizes $\sim_{\mathrm{HHPB}}$.

Theorem 6. *Let $P \in \mathbb{P}$. Then $\forall \phi \in \mathcal{L}_{\mathrm{BRM}}.\, P \models \phi \iff [\![P]\!] \models \phi$.* ∎

Corollary 2. *Let $P_1, P_2 \in \mathbb{P}$. Then $[\![P_1]\!] \sim_{\mathrm{HHPB}} [\![P_2]\!]$ iff $\forall \phi \in \mathcal{L}_{\mathrm{BRM}}.\, [\![P_1]\!] \models \phi \iff [\![P_2]\!] \models \phi$.* ∎

5.3 Translating $\mathcal{L}_{\mathrm{BRM}}$ into $\mathcal{L}_{\mathrm{EI}}$

The main difficulty is the encoding of multisets, as they are not present in $\mathcal{L}_{\mathrm{EI}}$. In the translation function we thus introduce two additional parameters:

- The first one is a finite set A of actions, e.g., those occurring in a process P. Since $P \models M$ iff $brm(P) = M$, the $\mathcal{L}_{\mathrm{EI}}$ formula corresponding to M has to express the fact that every action in the support of M, i.e., $supp(M) = \{a \in \mathcal{A} \mid M(a) > 0\}$, can be undone a number of times equal to its multiplicity, while any action in $A \setminus supp(M)$ cannot be undone at all. We assume that $supp(M)$ is finite to avoid infinite conjunctions in the translation.
- The second one is a finite sequence $\varrho_n : \{1, \dots, n\} \to \mathcal{I} \times \mathcal{A}$ of pairs each formed by an identifier and an action. It acts like a stack-based memory that keeps track of executed actions, bound to variables via $\langle x : a \rangle\!\rangle$ and $(x : a)$.

The translation function $\mathcal{T}_{\mathrm{BE}} : \mathcal{L}_{\mathrm{BRM}} \times \mathcal{P}_{\mathrm{fin}}(\mathcal{A}) \times \{\varrho_n \mid n \in \mathbb{N}_{\geq 1}\} \to \mathcal{L}_{\mathrm{EI}}$ is defined by induction on the syntactical structure of $\phi \in \mathcal{L}_{\mathrm{BRM}}$ as follows:

$$\begin{array}{rcll}
\mathcal{T}_{\mathrm{BE}}(\mathsf{true}, A, \varrho_n) & = & \mathsf{true} & \\
\mathcal{T}_{\mathrm{BE}}(M, A, \varrho_n) & = & \displaystyle\bigwedge_{a_i \in supp(M)} \left(\bigwedge_{k=1}^{M(a_i)} \langle\!\langle x_{i,k} \rangle \mathsf{true} \wedge \bigwedge_{h=1}^{\sharp(a_i, \varrho_n) - M(a_i)} \neg \langle\!\langle z_{i,h} \rangle \mathsf{true} \right) & \\
& & \wedge \displaystyle\bigwedge_{b \in A \setminus supp(M)} \neg (y : b) \langle\!\langle y \rangle \mathsf{true} & \text{with } y \text{ fresh} \\
\mathcal{T}_{\mathrm{BE}}(\neg\phi', A, \varrho_n) & = & \neg \mathcal{T}_{\mathrm{BE}}(\phi', A, \varrho_n) & \\
\mathcal{T}_{\mathrm{BE}}(\phi_1 \wedge \phi_2, A, \varrho_n) & = & \mathcal{T}_{\mathrm{BE}}(\phi_1, A, \varrho_n) \wedge \mathcal{T}_{\mathrm{BE}}(\phi_2, A, \varrho_n) & \\
\mathcal{T}_{\mathrm{BE}}(\langle a \rangle \phi', A, \varrho_n) & = & \langle x : a \rangle\!\rangle \mathcal{T}_{\mathrm{BE}}(\phi', A, \varrho_n \cup \{(n+1, (x, a))\}) & \text{with } x \text{ fresh} \\
\mathcal{T}_{\mathrm{BE}}(\langle a^\dagger \rangle \phi', A, \varrho_n) & = & (x : a) \langle\!\langle x \rangle \mathcal{T}_{\mathrm{BE}}(\phi', A, \varrho_n) & \text{with } x \text{ fresh}
\end{array}$$

where in the translation of M the finite sequence ϱ_n is such that:

- $(x_{i,k}, a_i) \in rge(\varrho_n)$, with $x_{i,k} \neq x_{i,k'}$ for $k \neq k'$ and all the identifiers $x_{i,k}$ being taken starting from the end of ϱ_n, i.e., the top of the stack.
- $\sharp(a_i, \varrho_n)$ is the number of occurrences of a_i in ϱ_n.
- $(z_{i,h}, a_i) \in rge(\varrho_n)$, with $z_{i,h} \notin \{x_{i,k} \mid 1 \leq k \leq M(a_i)\}$ and $z_{i,h} \neq z_{i,h'}$ for $h \neq h'$.

Theorem 7. *Let $P \in \mathbb{P}$, $\phi \in \mathcal{L}_{\mathrm{BRM}}$, and $act(P)$ be the set of actions in P. Then $P \models \phi$ iff $\exists \varrho_n . \exists \rho \in pe(P, \mathcal{T}_{\mathrm{BE}}(\phi, act(P), \varrho_n)). P \models_\rho \mathcal{T}_{\mathrm{BE}}(\phi, act(P), \varrho_n)$.* ■

5.4 Translating $\mathcal{L}_{\mathrm{EI}}$ into $\mathcal{L}_{\mathrm{BRM}}$

The challenge is the encoding of formulas of the form $(x : a)\phi$, because identifiers are not present in $\mathcal{L}_{\mathrm{BRM}}$ and the satisfaction of these formulas is not necessarily related to actions to be done or undone in this moment. Rather, it is generically related to executed actions. On the other hand, it is not clear how multisets would come into play. The study of this translation is left for future work.

6 Conclusions

In this paper we have proposed an entirely new approach to characterize hereditary history-preserving bisimilarity, both denotationally and operationally, even in the presence of autoconcurrency, autocausation, and autoconflict. Unlike [3], the focus is on counting identically labeled events rather than uniquely identifying them, thus avoiding bijections between events altogether. Moreover, on the operational side, it has turned out that proof terms naturally lend themselves to identification purposes; in a reversible setting like ours, they have been used for the first time in [1]. Finally, we have logically characterized backward-ready-multiset forward-reverse bisimilarity with backward ready multiset logic and investigated the relationships of the latter with event identifier logic [33].

The operational characterization is particularly important for several reasons. Firstly, in addition to the equational characterization over forward-only processes developed in [21], hereditary history-preserving bisimilarity can be axiomatized over reversible processes by resorting to the approach of [18] as applied in [8], provided that backward ready multisets are considered in place of backward ready sets. Secondly, in addition to the logics of [33,4], hereditary history-preserving bisimilarity can be characterized also in terms of backward ready multiset logic. The latter is simpler as it does not make use of variables and binders, but the former contain fragments that have been proven to characterize various behavioral equivalences in the true concurrency spectrum [23,19,32].

As for future work, we would like to complete the investigation of the relationships between backward ready multiset logic and event identifier logic, as well as to extend it to the logic of [4]. Another direction to pursue is whether our results apply to configuration structures that are not stable, i.e., in which it is not necessarily the case that causality among events can be always represented in terms of partial orders, possibly defined locally to each configuration. Hereditary history-preserving bisimilarity has been defined over non-stable models in [22,5] and a logical characterization for it has been provided in [5], which is a conservative extension of the one in [4].

Finally, we plan to study backward-ready-multiset forward-reverse bisimilarity checking by taking into account, as far as hereditary history-preserving bisimilarity is concerned, the undecidability result over finite labeled transition systems extended with an independence relation between transitions of [26] and the polynomial-time algorithm over basic parallel processes of [20].

Acknowledgments. We are grateful to Rob van Glabbeek for bringing to our attention the process examined in Example 2 as well as the anonymous reviewers for their comments and suggestions. This research has been supported by the PRIN 2020 project *NiRvAna – Noninterference and Reversibility Analysis in Private Blockchains*, the PRIN 2022 project *DeKLA – Developing Kleene Logics and Their Applications*, and the INdAM-GNCS 2024 project *MARVEL – Modelli Composizionali per l'Analisi di Sistemi Reversibili Distribuiti*.

References

1. Aubert, C.: Concurrencies in reversible concurrent calculi. In: Proc. of the 14th Int. Conf. on Reversible Computation (RC 2022). LNCS, vol. 13354, pp. 146–163. Springer (2022)
2. Aubert, C., Cristescu, I.: Contextual equivalences in configuration structures and reversibility. Journal of Logical and Algebraic Methods in Programming **86**, 77–106 (2017)
3. Aubert, C., Cristescu, I.: How reversibility can solve traditional questions: The example of hereditary history-preserving bisimulation. In: Proc. of the 31st Int. Conf. on Concurrency Theory (CONCUR 2020). LIPIcs, vol. 171, pp. 7:1–7:23 (2020)
4. Baldan, P., Crafa, S.: A logic for true concurrency. Journal of the ACM **61**, 24:1–24:36 (2014)
5. Baldan, P., Gorla, D., Padoan, T., Salvo, I.: Behavioural logics for configuration structures. Theoretical Computer Science **913**, 94–112 (2022)
6. Bednarczyk, M.A.: Hereditary history preserving bisimulations or what is the power of the future perfect in program logics. Technical Report, Polish Academy of Sciences, Gdansk (1991)
7. Bernardo, M., Esposito, A.: Modal logic characterizations of forward, reverse, and forward-reverse bisimilarities. In: Proc. of the 14th Int. Symp. on Games, Automata, Logics, and Formal Verification (GANDALF 2023). EPTCS, vol. 390, pp. 67–81 (2023)
8. Bernardo, M., Esposito, A., Mezzina, C.A.: Expansion laws for forward-reverse, forward, and reverse bisimilarities via proved encodings. In: Proc. of the Combined 31st Int. Workshop on Expressiveness in Concurrency and 21st Workshop on Structural Operational Semantics (EXPRESS/SOS 2024). EPTCS, vol. 412, pp. 51–70 (2024)

9. Bernardo, M., Rossi, S.: Reverse bisimilarity vs. forward bisimilarity. In: Proc. of the 26th Int. Conf. on Foundations of Software Science and Computation Structures (FOSSACS 2023). LNCS, vol. 13992, pp. 265–284. Springer (2023)
10. Boudol, G., Castellani, I.: A non-interleaving semantics for CCS based on proved transitions. Fundamenta Informaticae **11**, 433–452 (1988)
11. Boudol, G., Castellani, I.: Flow models of distributed computations: Three equivalent semantics for CCS. Information and Computation **114**, 247–314 (1994)
12. Brookes, S.D., Hoare, C.A.R., Roscoe, A.W.: A theory of communicating sequential processes. Journal of the ACM **31**, 560–599 (1984)
13. Danos, V., Krivine, J.: Reversible communicating systems. In: Proc. of the 15th Int. Conf. on Concurrency Theory (CONCUR 2004). LNCS, vol. 3170, pp. 292–307. Springer (2004)
14. Danos, V., Krivine, J.: Transactions in RCCS. In: Proc. of the 16th Int. Conf. on Concurrency Theory (CONCUR 2005). LNCS, vol. 3653, pp. 398–412. Springer (2005)
15. Darondeau, P., Degano, P.: Causal trees. In: Proc. of the 16th Int. Coll. on Automata, Languages and Programming (ICALP 1989). LNCS, vol. 372, pp. 234–248. Springer (1989)
16. Darondeau, P., Degano, P.: Causal trees: Interleaving + causality. In: Proc. of the LITP Spring School on Theoretical Computer Science: Semantics of Systems of Concurrent Processes. LNCS, vol. 469, pp. 239–255. Springer (1990)
17. De Nicola, R., Montanari, U., Vaandrager, F.: Back and forth bisimulations. In: Proc. of the 1st Int. Conf. on Concurrency Theory (CONCUR 1990). LNCS, vol. 458, pp. 152–165. Springer (1990)
18. Degano, P., Priami, C.: Proved trees. In: Proc. of the 19th Int. Coll. on Automata, Languages and Programming (ICALP 1992). LNCS, vol. 623, pp. 629–640. Springer (1992)
19. Fecher, H.: A completed hierarchy of true concurrent equivalences. Information Processing Letters **89**, 261–265 (2004)
20. Fröschle, S., Jančar, P., Lasota, S., Sawa, Z.: Non-interleaving bisimulation equivalences on basic parallel processes. Information and Computation **208**, 42–62 (2010)
21. Fröschle, S., Lasota, S.: Decomposition and complexity of hereditary history preserving bisimulation on BPP. In: Proc. of the 16th Int. Conf. on Concurrency Theory (CONCUR 2005). LNCS, vol. 3653, pp. 263–277. Springer (2005)
22. van Glabbeek, R.J.: On the expressiveness of higher dimensional automata. Theoretical Computer Science **368**, 168–194 (2006)
23. van Glabbeek, R.J., Goltz, U.: Refinement of actions and equivalence notions for concurrent systems. Acta Informatica **37**, 229–327 (2001)
24. van Glabbeek, R.J., Plotkin, G.D.: Configuration structures, event structures and Petri nets. Theoretical Computer Science **410**, 4111–4159 (2009)
25. Joyal, A., Nielsen, M., Winskel, G.: Bisimulation from open maps. Information and Computation **127**, 164–185 (1996)
26. Jurdzinski, M., Nielsen, M., Srba, J.: Undecidability of domino games and hhp-bisimilarity. Information and Computation **184**, 343–368 (2003)
27. Krivine, J.: A verification technique for reversible process algebra. In: Proc. of the 4th Int. Workshop on Reversible Computation (RC 2012). LNCS, vol. 7581, pp. 204–217. Springer (2012)
28. Lanese, I., Medić, D., Mezzina, C.A.: Static versus dynamic reversibility in CCS. Acta Informatica **58**, 1–34 (2021)
29. Milner, R.: Communication and Concurrency. Prentice Hall (1989)

30. Phillips, I., Ulidowski, I.: Reversing algebraic process calculi. Journal of Logic and Algebraic Programming **73**, 70–96 (2007)
31. Phillips, I., Ulidowski, I.: Reversibility and models for concurrency. In: Proc. of the 4th Int. Workshop on Structural Operational Semantics (SOS 2007). ENTCS, vol. 192(1), pp. 93–108. Elsevier (2007)
32. Phillips, I., Ulidowski, I.: A hierarchy of reverse bisimulations on stable configuration structures. Mathematical Structures in Computer Science **22**, 333–372 (2012)
33. Phillips, I., Ulidowski, I.: Event identifier logic. Mathematical Structures in Computer Science **24(2)**, 1–51 (2014)
34. Rabinovich, A.M., Trakhtenbrot, B.A.: Behavior structures and nets. Fundamenta Informaticae **11**, 357–404 (1988)
35. Winskel, G.: Event structures. In: Advances in Petri Nets. LNCS, vol. 255, pp. 325–392. Springer (1986)

A Proofs of Results

Since the proofs of Theorems 3, 5, 6, and 7, of Lemma 2, and of Corollary 1 proceed by induction on the depth of a formula of $\mathcal{L}_{\rm EI}$ or $\mathcal{L}_{\rm BRM}$, where the depth is intended as an upper bound to the depth of the syntax tree of the considered formula, we collect here the related definitions:

- The depth of $\phi \in \mathcal{L}_{\rm EI}$ is defined by induction on the syntactical structure of ϕ as follows:

$$\begin{aligned} depth(\mathsf{true}) &= 0 \\ depth(\neg \phi') &= 1 + depth(\phi') \\ depth(\phi_1 \wedge \phi_2) &= 1 + \max(depth(\phi_1), depth(\phi_2)) \\ depth(\langle x : a \rangle\!\rangle \phi') &= 1 + depth(\phi') \\ depth((x : a)\phi') &= 1 + depth(\phi') \\ depth(\langle\!\langle x \rangle \phi') &= 1 + depth(\phi') \end{aligned}$$

- The depth of $\phi \in \mathcal{L}_{\rm BRM}$ is defined by induction on the syntactical structure of ϕ as follows:

$$\begin{aligned} depth(\mathsf{true}) &= 0 \\ depth(M) &= 0 \\ depth(\neg \phi') &= 1 + depth(\phi') \\ depth(\phi_1 \wedge \phi_2) &= 1 + \max(depth(\phi_1), depth(\phi_2)) \\ depth(\langle a \rangle \phi') &= 1 + depth(\phi') \\ depth(\langle a^{\dagger} \rangle \phi') &= 1 + depth(\phi') \end{aligned}$$

Theorem 2 holds under the assumption that, in the presence of autoconcurrency, for each maximal set of conflicting events (i.e., events jointly occurring in no configuration) all the events in the set are caused by the same event. The two configuration structures in Figure 12 of [32], which come from the two event structures in Figure 11 of the same paper and do not satisfy the assumption, are distinguished by $\sim_{\rm HHPB}$ but identified by $\sim_{\rm FRB:brm}$ as pointed out by I. Ulidowski, I. Phillips, and C. Aubert in a personal communication to the authors. Below is the revised proof.

Proof of Theorem 2. The proof is divided into two parts:

- Suppose that $\mathcal{C}_1 \sim_{\rm HHPB} \mathcal{C}_2$ due to some hereditary history-preserving bisimulation $\mathcal{B}$. Then $\mathcal{C}_1 \sim_{\rm FRB:brm} \mathcal{C}_2$ follows by proving that $\mathcal{B}' = \{(X_1, X_2) \mid (X_1, X_2, f) \in \mathcal{B}\}$ is a brm-forward-reverse bisimulation. Observing that the starting clause and the clauses for outgoing and incoming transitions matching of $\sim_{\rm FRB:brm}$ (see Definition 3) are a simplification of those of $\sim_{\rm HHPB}$ (see Definition 2), given $(X_1, X_2) \in \mathcal{B}'$, i.e., $(X_1, X_2, f) \in \mathcal{B}$ for some labeling- and causality-preserving bijection f from X_1 to X_2, we just have to show that $brm(X_1) = brm(X_2)$.
  Suppose that this is not the case, say X_1 has fewer incoming a-transitions than X_2. Without loss of generality, we can assume that X_1 has one incoming a-transition while X_2 has two. Then in $\mathcal{C}_2$ there is a diamond closing into X_2,

i.e., there exist three configurations Y_2, X'_2, and X''_2 and two a-labeled events e'_2 and e''_2 such that $Y_2 \xrightarrow{l_2(e'_2)}_{C_2} X'_2$, $Y_2 \xrightarrow{l_2(e''_2)}_{C_2} X''_2$, $X'_2 \xrightarrow{l_2(e''_2)}_{C_2} X_2$, and $X''_2 \xrightarrow{l_2(e'_2)}_{C_2} X_2$, with e'_2 and e''_2 concurrent in X_2.
Due to $(X_1, X_2, f) \in \mathcal{B}$, in C_1 there must be two configurations Y_1 and X'_1 and two a-labeled events e'_1 and e''_1 such that $Y_1 \xrightarrow{l_1(e'_1)}_{C_1} X'_1 \xrightarrow{l_1(e''_1)}_{C_1} X_1$, with $e'_1 \leq_{X_1} e''_1$ because X_1 has a single incoming a-transition. Since $\mathcal{B}$ is a hereditary history-preserving bisimulation, f should relate $e'_1, e''_1 \in X_1$ with $e'_2, e''_2 \in X_2$ in a causality-preserving way, but this is not possible because $f(e'_1) \not\leq_{X_2} f(e''_1)$ where $f(e'_1) \in \{e'_2, e''_2\}$ and $f(e''_1) \in \{e'_2, e''_2\} \setminus \{f(e'_1)\}$.

- Suppose that $C_1 \sim_{\text{FRB:brm}} C_2$ due to some maximal brm-forward-reverse bisimulation $\mathcal{B}$. Then, given $(X_1, X_2) \in \mathcal{B}$, the existence in C_1 of a sequence of transitions $X_{1,n} \xrightarrow{l_1(e_{1,n})}_{C_1} X_{1,n-1} \dots X_{1,1} \xrightarrow{l_1(e_{1,1})}_{C_1} X_1$ implies the existence in C_2 of a sequence of transitions $X_{2,n} \xrightarrow{l_2(e_{2,n})}_{C_2} X_{2,n-1} \dots X_{2,1} \xrightarrow{l_2(e_{2,1})}_{C_2} X_2$ such that $l_1(e_{1,h}) = l_2(e_{2,h})$ and $(X_{1,h}, X_{2,h}) \in \mathcal{B}$ for all $h = 1, \dots, n$, and vice versa. Note that $e_{1,h} \neq e_{1,k}$ and $e_{2,h} \neq e_{2,k}$ for all $h \neq k$ because in each transition the source configuration and the target configuration differ by one event, which is the executed event (see Definition 1).
  Thus $C_1 \sim_{\text{HHPB}} C_2$ follows by proving that $\mathcal{B}' = \{(X_1, X_2, \{(e_{1,h}, e_{2,h}) \mid h \in H\}) \mid (X_1, X_2) \in \mathcal{B} \wedge X_{i,|H|} \xrightarrow{l_i(e_{i,|H|})}_{C_i} X_{i,|H|-1} \dots X_{i,1} \xrightarrow{l_i(e_{i,1})}_{C_i} X_i$ for $i \in \{1, 2\} \wedge l_1(e_{1,h}) = l_2(e_{2,h})$ for all $h \in H \wedge (X_{1,h}, X_{2,h}) \in \mathcal{B}$ for all $h \in H$ $\wedge\ X_{1,|H|} = \emptyset = X_{2,|H|}\}$ is a hereditary history-preserving bisimulation. Observing that $(\emptyset, \emptyset) \in \mathcal{B}$ implies $(\emptyset, \emptyset, \emptyset) \in \mathcal{B}'$, take $(X_1, X_2, \{(e_{1,h}, e_{2,h}) \mid h \in H\}) \in \mathcal{B}'$, so that $(X_1, X_2) \in \mathcal{B}$:
  - If $X_1 \xrightarrow{a}_{C_1} X'_1$ due to some a-labeled event e_1, then $X_2 \xrightarrow{a}_{C_2} X'_2$ due to some a-labeled event e_2 such that $(X'_1, X'_2) \in \mathcal{B}$. Since $e_1 \notin X_1$ and $e_2 \notin X_2$, it holds that $(X'_1, X'_2, \{(e_{1,h}, e_{2,h}) \mid h \in H\} \cup \{(e_1, e_2)\}) \in \mathcal{B}'$.
    If we start from $X_2 \xrightarrow{a}_{C_2} X'_2$, then we reason in the same way.
  - If $X'_1 \xrightarrow{a}_{C_1} X_1$ due to some a-labeled event e_1, then $X'_2 \xrightarrow{a}_{C_2} X_2$ due to some a-labeled event e_2 such that $(X'_1, X'_2) \in \mathcal{B}$. Since $e_1 \notin X'_1$, $e_2 \notin X'_2$, and $brm(X_1) = brm(X_2)$, the latter transition can be selected in such a way to satisfy $\{(e_{1,h}, e_{2,h}) \mid h \in H\} \upharpoonright X'_1 = \{(e_{1,h}, e_{2,h}) \mid h \in H\} \setminus \{(e_1, e_2)\}$, hence $(X'_1, X'_2, \{(e_{1,h}, e_{2,h}) \mid h \in H\} \setminus \{(e_1, e_2)\}) \in \mathcal{B}'$.
    If we start from $X'_2 \xrightarrow{a}_{C_2} X_2$, then we reason in the same way.
  - $f = \{(e_{1,h}, e_{2,h}) \mid h \in H\}$ certainly is a bijection from X_1 to X_2 – as the events along either computation are different from each other, so the two reached configurations X_1 and X_2 contain the same number of events, and paired in a stepwise manner – that preserves labeling – by definition of $\mathcal{B}'$. If $|H| \leq 1$ then causality is trivially preserved.
    Suppose that X_1 and X_2 break causality and, among all the pairs of $\mathcal{B}$-related configurations that break causality, they are the closest ones to $\emptyset$ and $\emptyset$ (in terms of number of transitions to be executed from either empty configuration). Breaking causality means that $X_1 \xrightarrow{a}_{C_1} X'_1$ due to an a-labeled event $e'_1 \notin X_1$ such that $e_1 \leq_{X'_1} e'_1$ for some $e_1 \in X_1$,

but all the responses $X_2 \xrightarrow{a}_{C_2} X'_2$ complying with $\mathcal{B}$, which is maximal, are such that $g(e_1) \not\leq_{X'_2} g(e'_1)$, where g is f extended with the new pair of events.
(In Figure 12 of [32], X_1 and X_2 are states $6_\mathcal{E}$ and $6_\mathcal{F}$, X'_1 and X'_2 are states $13_\mathcal{E}$ and $12_\mathcal{F}$, $e_1 = a_2$, $e'_1 = b_2$, $g(e_1) = a_2$, $g(e'_1) = b_3$.)
From X_1 and X_2 we go backward by following the respective computations undertaken in the forward direction with respect to $\mathcal{B}'$ – without undoing e_1 – until we reach Y_1 and Y_2 such that $(Y_1, Y_2) \in \mathcal{B}$ having several incoming transitions. Note that Y_1 and Y_2 have the same number of incoming transitions because $\mathcal{B}$ is a brm-forward-reverse bisimulation.
(In Figure 12 of [32], Y_1 and Y_2 are again states $6_\mathcal{E}$ and $6_\mathcal{F}$.)
In Y_1 and Y_2 we undo two identically labeled transitions matched by $\mathcal{B}$, respectively corresponding to two events e''_1 and e''_2, different from the two transitions with which we arrived at Y_1 and Y_2 via the computations undertaken in the forward direction with respect to $\mathcal{B}'$. Let Y'_1 and Y'_2 be the two newly reached configurations such that $(Y'_1, Y'_2) \in \mathcal{B}$.
(In Figure 12 of [32], Y'_1 and Y'_2 are states $3_\mathcal{E}$ and $3_\mathcal{F}$, $e''_1 = a_3$, $e''_2 = a_3$.)
There are two cases:

* If in Y'_1 it is possible to perform a transition corresponding to e'_1, then we execute it so as to recreate $e_1 \leq e'_1$ along a different computation of C_1. Therefore Y'_2 responds by executing an identically labeled transition corresponding to some event e'_2 such that $h(e_1) \leq h(e'_1) = e'_2$, where h is the resulting bijection, as Y'_1 and Y'_2 are closer to $\emptyset$ and $\emptyset$ than X_1 and X_2 – remember the assumption that X_1 and X_2 are the first $\mathcal{B}$-related configurations that break causality. Let Y''_1 and Y''_2 be the two newly reached configurations such that $(Y''_1, Y''_2) \in \mathcal{B}$.
  (In Figure 12 of [32], Y''_1 and Y''_2 are states $8_\mathcal{E}$ and $8_\mathcal{F}$, $e'_2 = b_2$.)
  In Y''_1 it is possible to perform a transition corresponding to e''_1 otherwise we would not have been able to go from $\emptyset$ to X'_1 via Y'_1 and X_1 by encountering e''_1 and e'_1. In other words, e''_1 and e'_1 are concurrent. If in Y''_2 it were possible to perform a transition corresponding to e''_2, then in X_2 we had to be able to execute a transition corresponding to e'_2 thanks to concurrency, but this contradicts the assumption that X_2 cannot respond to X_1 in a causality-preserving way. Therefore e''_2 and e'_2 have to be in conflict with each other (note that $e'_2 \notin X_2$ and $e''_2 \notin Y''_2$), but this contradicts the assumption of the absence of conflicting events caused by different events. Indeed, since e''_2 is taken from the bottom side of a diamond opposite to the one of $g(e_1)$, e''_2 must be concurrent to $g(e_1)$, hence if e''_2 and e'_2 were conflicting, then they would not be caused by the same event.
* If in Y'_1 it is not possible to perform a transition corresponding to e'_1, then we keep going forward until we find it by executing transitions corresponding to all the events that have been undone by going from X_1 and X_2 back to Y_1 and Y_2.

If the aforementioned Y_1 and Y_2 did not exist, then while going backward from X_1 and X_2 by following the respective computations undertaken in the forward direction with respect to $\mathcal{B}'$ – without undoing e_1 – we should reach Z_1 and Z_2 such that $(Z_1, Z_2) \in \mathcal{B}$ with at least one of them having several outgoing transitions. The reason is that if neither Y_1 and Y_2 nor Z_1 and Z_2 existed, then there would be a single computation from $\emptyset$ to X_1 and a single computation from $\emptyset$ to X_2 – in which all events are causally related and hence there is no concurrency – thus contradicting $e_1 \leq_{X'_1} e'_1$ and $g(e_1) \not\leq_{X'_2} g(e'_1)$ as it would hold that $g(e_1) \leq_{X'_2} g(e'_1)$. However, since after leaving Z_1 and Z_2 towards X_1 and X_2 there would be neither $\mathcal{B}$-related configurations with several incoming transitions nor $\mathcal{B}$-related configurations at least one of which has several outgoing transitions, there would be a single computation from Z_1 to X_1 and a single computation from Z_2 to X_2 – in which all events are causally related and hence there is no concurrency – thus contradicting again $e_1 \leq_{X'_1} e'_1$ and $g(e_1) \not\leq_{X'_2} g(e'_1)$ as it would hold that $g(e_1) \leq_{X'_2} g(e'_1)$. ■

Proof of Theorem 3. The proof is divided into two parts:

- Assuming that $P_1 \sim_{\mathrm{FRB:brm}} P_2$ and $P_1 \models \phi$ for an arbitrary formula $\phi \in \mathcal{L}_{\mathrm{FRB:brm}}$, we prove that $P_2 \models \phi$ too by proceeding by induction on $k = \mathit{depth}(\phi)$:
  - If $k = 0$ then ϕ is either $\mathtt{true}$ or M. In the former case, it is trivially satisfied by P_2 too. In the latter case, it is satisfied by P_2 too because $P_1 \sim_{\mathrm{FRB:brm}} P_2$ and hence $\mathit{brm}(P_1) = \mathit{brm}(P_2)$.
  - If $k \geq 1$ then there are four cases:
    - ∗ If ϕ is $\neg\phi'$ then from $P_1 \models \neg\phi'$ we derive that $P_1 \not\models \phi'$. If it were $P_2 \models \phi'$ then by the induction hypothesis it would hold that $P_1 \models \phi'$, which is not the case. Therefore $P_2 \not\models \phi'$ and hence $P_2 \models \neg\phi'$ too.
    - ∗ If ϕ is $\phi_1 \wedge \phi_2$ then from $P_1 \models \phi_1 \wedge \phi_2$ we derive that $P_1 \models \phi_1$ and $P_1 \models \phi_2$. From the induction hypothesis it follows that $P_2 \models \phi_1$ and $P_2 \models \phi_2$ and hence $P_2 \models \phi_1 \wedge \phi_2$ too.
    - ∗ If ϕ is $\langle a \rangle \phi'$ then from $P_1 \models \langle a \rangle \phi'$ we derive that there exists $P_1 \xrightarrow{\theta_1} P'_1$ such that $\mathit{act}(\theta_1) = a$ and $P'_1 \models \phi'$. From $P_1 \sim_{\mathrm{FRB:brm}} P_2$ it then follows that there exists $P_2 \xrightarrow{\theta_2} P'_2$ such that $\mathit{act}(\theta_2) = a$ and $P'_1 \sim_{\mathrm{FRB:brm}} P'_2$. By applying the induction hypothesis we derive that $P'_2 \models \phi'$ and hence $P_2 \models \langle a \rangle \phi'$ too.
    - ∗ If ϕ is $\langle a^\dagger \rangle \phi'$ then from $P_1 \models \langle a^\dagger \rangle \phi'$ we derive that there exists $P'_1 \xrightarrow{\theta_1} P_1$ such that $\mathit{act}(\theta_1) = a$ and $P'_1 \models \phi'$. From $P_1 \sim_{\mathrm{FRB:brm}} P_2$ it then follows that there exists $P'_2 \xrightarrow{\theta_2} P_2$ such that $\mathit{act}(\theta_2) = a$ and $P'_1 \sim_{\mathrm{FRB:brm}} P'_2$. By applying the induction hypothesis we derive that $P'_2 \models \phi'$ and hence $P_2 \models \langle a^\dagger \rangle \phi'$ too.
- Assuming that P_1 and P_2 satisfy the same formulas in $\mathcal{L}_{\mathrm{BRM}}$, we prove that the symmetric relation $\mathcal{B} = \{(Q_1, Q_2) \in \mathbb{P} \times \mathbb{P} \mid Q_1 \text{ and } Q_2 \text{ satisfy the same formulas in } \mathcal{L}_{\mathrm{BRM}}\}$ is a brm-forward-reverse bisimulation.
  Given $(Q_1, Q_2) \in \mathcal{B}$:

- If $Q_1 \xrightarrow{\theta_1} Q'_1$ suppose by contradiction that there is no Q'_2 satisfying the same formulas as Q'_1 such that $Q_2 \xrightarrow{\theta_2} Q'_2$ and $act(\theta_1) = act(\theta_2)$, i.e., $(Q'_1, Q'_2) \in \mathcal{B}$ for no Q'_2 $act(\theta_1)$-reachable from Q_2. Since Q_2 has finitely many outgoing transitions, the set of processes that Q_2 can reach by performing an $act(\theta_1)$-transition is finite, say $\{Q'_{2,1}, \dots, Q'_{2,n}\}$ with $n \geq 0$. Since none of the processes in the set satisfies the same formulas as Q'_1, for each $1 \leq i \leq n$ there exists $\phi_i \in \mathcal{L}_{\mathrm{BRM}}$ such that $Q'_1 \models \phi_i$ but $Q'_{2,i} \not\models \phi_i$.

  We can then construct the formula $\langle act(\theta_1) \rangle \bigwedge\limits_{i=1}^{n} \phi_i$ that is satisfied by Q_1 but not by Q_2; if $n = 0$ then it is sufficient to take $\langle act(\theta_1) \rangle \mathsf{true}$. This formula violates $(Q_1, Q_2) \in \mathcal{B}$, hence there must exist at least one Q'_2 satisfying the same formulas as Q'_1 such that $Q_2 \xrightarrow{\theta_2} Q'_2$ and $act(\theta_1) = act(\theta_2)$, so that $(Q'_1, Q'_2) \in \mathcal{B}$.
- If $Q'_1 \xrightarrow{\theta_1} Q_1$ suppose by contradiction that there is no Q'_2 satisfying the same formulas as Q'_1 such that $Q'_2 \xrightarrow{\theta_2} Q_2$ and $act(\theta_1) = act(\theta_2)$, i.e., $(Q'_1, Q'_2) \in \mathcal{B}$ for no Q'_2 $act(\theta_1)$-reaching Q_2. Since Q_2 has finitely many incoming transitions, the set of processes that can reach Q_2 by performing an $act(\theta_1)$-transition is finite, say $\{Q'_{2,1}, \dots, Q'_{2,n}\}$ with $n \geq 0$. Since none of the processes in the set satisfies the same formulas as Q'_1, for each $1 \leq i \leq n$ there exists $\phi_i \in \mathcal{L}_{\mathrm{BRM}}$ such that $Q'_1 \models \phi_i$ but $Q'_{2,i} \not\models \phi_i$.

  We can then construct the formula $\langle act(\theta_1)^\dagger \rangle \bigwedge\limits_{i=1}^{n} \phi_i$ that is satisfied by Q_1 but not by Q_2; if $n = 0$ then it is sufficient to take $\langle act(\theta_1)^\dagger \rangle \mathsf{true}$. This formula violates $(Q_1, Q_2) \in \mathcal{B}$, hence there must exist at least one Q'_2 satisfying the same formulas as Q'_1 such that $Q'_2 \xrightarrow{\theta_2} Q_2$ and $act(\theta_1) = act(\theta_2)$, so that $(Q'_1, Q'_2) \in \mathcal{B}$.
- The fact that $brm(Q_1) = brm(Q_2)$ follows from the fact that Q_1 and Q_2 satisfy, in particular, the same formulas of the form M. ■

Proof of Lemma 1. The proof is divided into two parts:

- Suppose that $P \xrightarrow{\theta} P'$. We show that $X_P \xrightarrow{act(\theta)}_{\mathsf{C}_P} X_{P'}$ by proceeding by induction on the number $n \in \mathbb{N}_{\geq 1}$ of applications of operational semantic rules in Table 1 that are necessary to derive the transition $P \xrightarrow{\theta} P'$:
  - If $n = 1$ then P is $a \,.\, Q$ with $init(Q)$, $a \,.\, Q \xrightarrow{a} a^\dagger \,.\, Q$ by rule $\textsc{Act}_{\mathrm{f}}$, and P' is $a^\dagger \,.\, Q$. From $init(P)$ it follows that $[\![P]\!] = (\mathsf{C}_P, \emptyset)$ and $[\![P']\!] = (\mathsf{C}_P, \{a\})$, hence $X_P \xrightarrow{a}_{\mathsf{C}_P} X_{P'}$.
  - If $n > 1$ then there are three cases:
    - ∗ Let P be $a^\dagger \,.\, Q$ so that $P \xrightarrow{\cdot a \theta'} P'$ with P' being $a^\dagger \,.\, Q'$. Then $Q \xrightarrow{\theta'} Q'$ by rule $\textsc{Act}_{\mathrm{p}}$, hence $X_Q \xrightarrow{act(\theta')}_{\mathsf{C}_Q} X_{Q'}$ by the induction hypothesis with $[\![Q]\!] = (\mathsf{C}_Q, X_Q)$ and $[\![Q']\!] = (\mathsf{C}_Q, X_{Q'})$.

      Let $to\_init(Q) \xrightarrow{\theta_1} \dots \xrightarrow{\theta_n} Q \xrightarrow{\theta'} Q'$ with $n \in \mathbb{N}$, so $X_Q = \{\theta_i \mid$

$1 \leq i \leq n\}$ and $X_{Q'} = X_Q \cup \{\theta'\}$. Then $\mathit{to\_init}(P) = a\,.\,\mathit{to\_init}(Q) \xrightarrow{a} \xrightarrow{._a\theta_1} \dots \xrightarrow{._a\theta_n} P \xrightarrow{._a\theta'} P'$, $[\![P]\!] = (a\,.\,\mathsf{C}_Q, X_P)$ with $X_P = \{a\} \cup \{._a\theta_i \mid \theta_i \in X_Q\}$, and $[\![P']\!] = (a\,.\,\mathsf{C}_Q, X_{P'})$ with $X_{P'} = X_P \cup \{._a\theta'\}$ and $._a\theta' \notin X_P$. Thus $X_P \xrightarrow{act(._a\theta')}_{\mathsf{C}_P} X_{P'}$ with $\mathsf{C}_P = a\,.\,\mathsf{C}_Q$.

- ∗ Let P be $P_1 + P_2$. There are two subcases:
  - · If P_1 moves, i.e., $P \xrightarrow{+\theta'} P'$ with P' being $P'_1 + P_2$ and $\mathit{init}(P_2)$, then $P_1 \xrightarrow{\theta'} P'_1$ by rule Cho$_1$, hence $X_{P_1} \xrightarrow{act(\theta')}_{\mathsf{C}_{P_1}} X_{P'_1}$ by the induction hypothesis with $[\![P_1]\!] = (\mathsf{C}_{P_1}, X_{P_1})$ and $[\![P'_1]\!] = (\mathsf{C}_{P_1}, X_{P'_1})$. Let $\mathit{to\_init}(P_1) \xrightarrow{\theta_1} \dots \xrightarrow{\theta_n} P_1 \xrightarrow{\theta'} P'_1$ with $n \in \mathbb{N}$, so $X_{P_1} = \{\theta_i \mid 1 \leq i \leq n\}$ and $X_{P'_1} = X_{P_1} \cup \{\theta'\}$. Then $\mathit{to\_init}(P) = \mathit{to\_init}(P_1) + P_2 \xrightarrow{+\theta_1} \dots \xrightarrow{+\theta_n} P_1 + P_2 \xrightarrow{+\theta'} P'$, $[\![P]\!] = (\mathsf{C}_{P_1} + \mathsf{C}_{P_2}), X_P)$ with $X_P = \{+\theta_i \mid \theta_i \in X_{P_1}\}$, and $[\![P']\!] = (\mathsf{C}_{P_1} + \mathsf{C}_{P_2}, X_{P'})$ with $X_{P'} = X_P \cup \{+\theta'\}$ and $+\theta' \notin X_P$. Thus $X_P \xrightarrow{act(+\theta')}_{\mathsf{C}_P} X_{P'}$ with $\mathsf{C}_P = \mathsf{C}_{P_1} + \mathsf{C}_{P_2}$.
  - · The subcase in which P_2 moves and P_1 is initial is like the previous one.
- ∗ Let P be $P_1 \,\|_L\, P_2$. Given two sequences $\overrightarrow{\theta_1}$ and $\overrightarrow{\theta_2}$ of proof terms labeling two sequences of proved transitions respectively departing from P_1 and P_2, we characterize their interleaving and synchronization through the function $\mathit{zip} : \Theta^* \times \Theta^* \times 2^{\mathcal{A}\setminus\{\tau\}} \to \Theta^*$ defined by induction on the sum of the lengths of its first two arguments $\overrightarrow{\theta_1}, \overrightarrow{\theta_2} \in \Theta^*$ as follows:

$$\mathit{zip}(\overrightarrow{\theta_1}, \overrightarrow{\theta_2}, L) = \begin{cases} \|\!\rfloor_L \theta'\, \mathit{zip}(\overrightarrow{\theta'_1}, \overrightarrow{\theta_2}) & \text{if } \overrightarrow{\theta_1} = \theta'\overrightarrow{\theta'_1} \wedge act(\theta') \notin L \wedge (\overrightarrow{\theta_2} = \varepsilon \vee \\ & \quad (\overrightarrow{\theta_2} = \theta''\overrightarrow{\theta''_2} \wedge (act(\theta'') \in L \vee |\overrightarrow{\theta_1}| \geq |\overrightarrow{\theta_2}|))) \\ \lfloor\!\|_L \theta''\, \mathit{zip}(\overrightarrow{\theta_1}, \overrightarrow{\theta''_2}) & \text{if } \overrightarrow{\theta_2} = \theta''\overrightarrow{\theta''_2} \wedge act(\theta'') \notin L \wedge (\overrightarrow{\theta_1} = \varepsilon \vee \\ & \quad (\overrightarrow{\theta_1} = \theta'\overrightarrow{\theta'_1} \wedge (act(\theta') \in L \vee |\overrightarrow{\theta_1}| < |\overrightarrow{\theta_2}|))) \\ \langle \theta', \theta'' \rangle_L\, \mathit{zip}(\overrightarrow{\theta'_1}, \overrightarrow{\theta''_2}) & \text{if } \overrightarrow{\theta_1} = \theta'\overrightarrow{\theta'_1} \wedge \overrightarrow{\theta_2} = \theta''\overrightarrow{\theta''_2} \wedge act(\theta') = act(\theta'') \in L \\ \varepsilon & \text{otherwise} \end{cases}$$

  There are three subcases:
  - · If $act(\theta) \notin L$ and P_1 moves, i.e., $P \xrightarrow{\|\!\rfloor_L \theta'} P'$ with P' being $P'_1 \,\|_L\, P_2$, then $P_1 \xrightarrow{\theta'} P'_1$ by rule Par$_1$, hence $X_{P_1} \xrightarrow{act(\theta')}_{\mathsf{C}_{P_1}} X_{P'_1}$ by the induction hypothesis with $[\![P_1]\!] = (\mathsf{C}_{P_1}, X_{P_1})$ and $[\![P'_1]\!] = (\mathsf{C}_{P_1}, X_{P'_1})$. Let $\mathit{to\_init}(P_1) \xrightarrow{\theta_{1,1}} \dots \xrightarrow{\theta_{1,n_1}} P_1 \xrightarrow{\theta'} P'_1$ with $n_1 \in \mathbb{N}$, so $X_{P_1} = \{\theta_{1,i} \mid 1 \leq i \leq n_1\}$ and $X_{P'_1} = X_{P_1} \cup \{\theta'\}$. Also let $\mathit{to\_init}(P_2) \xrightarrow{\theta_{2,1}} \dots \xrightarrow{\theta_{2,n_2}} P_2$ with $n_2 \in \mathbb{N}$, so $X_{P_2} = \{\theta_{2,i} \mid 1 \leq i \leq n_2\}$. We denote by $\overrightarrow{\theta_1}$ and $\overrightarrow{\theta_2}$ the two sequences of proof terms. Then $\mathit{to\_init}(P) = \mathit{to\_init}(P_1) \,\|_L\, \mathit{to\_init}(P_2)$ reaches P via a sequence of proved transitions labeled with $\mathit{zip}(\overrightarrow{\theta_1}, \overrightarrow{\theta_2}, L)$ and afterwards P' via $P \xrightarrow{\|\!\rfloor_L \theta'} P'$, $[\![P]\!] = (\mathsf{C}_{P_1} \,\|_L\, \mathsf{C}_{P_2}, X_P)$ with $X_P = \{\bar{\theta} \mid$

$\bar{\theta}$ in $zip(\overrightarrow{\theta_1}, \overrightarrow{\theta_2}, L)\}$, and $[\![P']\!] = (C_{P_1} \parallel_L C_{P_2}, X_{P'})$ with $X_{P'} = X_P \cup \{\rVert_L \theta'\}$ and $\rVert_L \theta' \notin X_P$. Thus $X_P \xrightarrow{act(\rVert_L \theta')}_{C_P} X_{P'}$ with $C_P = C_{P_1} \parallel_L C_{P_2}$.

· The subcase in which $act(\theta) \notin L$ and P_2 moves is like the previous one.

· If $act(\theta) \in L$, i.e., $P \xrightarrow{\langle \theta_1, \theta_2 \rangle_L} P'$ with P' being $enr(P'_1 \parallel_L P'_2, \langle \theta_1, \theta_2 \rangle_L)$, then $P_k \xrightarrow{\theta_k} P'_k$ for $k \in \{1, 2\}$ by rule Syn, hence $X_{P_k} \xrightarrow{act(\theta_k)}_{C_{P_k}} X_{P'_k}$ by the induction hypothesis with $[\![P_k]\!] = (C_{P_k}, X_{P_k})$ and $[\![P'_k]\!] = (C_{P_k}, X_{P'_k})$.

For $k \in \{1, 2\}$ let $to\_init(P_k) \xrightarrow{\theta_{k,1}} \dots \xrightarrow{\theta_{k,n_k}} P_k \xrightarrow{\theta_k} P'_k$ with $n_k \in \mathbb{N}$, so $X_{P_k} = \{\theta_{k,i} \mid 1 \leq i \leq n_k\}$ and $X_{P'_k} = X_{P_k} \cup \{\theta_k\}$; we denote by $\overrightarrow{\theta_k}$ the sequence of proof terms. Then $to\_init(P) = to\_init(P_1) \parallel_L to\_init(P_2)$ reaches P via a sequence of proved transitions labeled with $zip(\overrightarrow{\theta_1}, \overrightarrow{\theta_2}, L)$ and afterwards P' via $P \xrightarrow{\langle \theta_1, \theta_2 \rangle_L} P'$, $[\![P]\!] = (C_{P_1} \parallel_L C_{P_2}, X_P)$ with $X_P = \{\bar{\theta} \mid \bar{\theta}$ in $zip(\overrightarrow{\theta_1}, \overrightarrow{\theta_2}, L)\}$, and $[\![P']\!] = (C_{P_1} \parallel_L C_{P_2}, X_{P'})$ with $X_{P'} = X_P \cup \{\langle \theta_1, \theta_2 \rangle_L\}$ and $\langle \theta_1, \theta_2 \rangle_L \notin X_P$. Thus $X_P \xrightarrow{act(\langle \theta_1, \theta_2 \rangle_L)}_{C_P} X_{P'}$ with $C_P = C_{P_1} \parallel_L C_{P_2}$.

– Suppose that $X_P \xrightarrow{act(\theta)}_{C_P} X_{P'}$. Consider a variant Θ' of Θ in which $\dot{+}$, $\underset{.}{+}$, $\rVert_L$, and $\lVert_L$ are given a subscript equal to the subprocess that does not move and this is applied to the corresponding rules in Table 1 as well as the corresponding operations on stable configuration structures in Section 4.4.

Although the configuration structure transition $X_P \xrightarrow{act(\theta)}_{C_P} X_{P'}$ is not generated inductively, from the only event θ – enriched as described above – in $X_{P'} \setminus X_P$ we can pinpoint $act(\theta)$ within P and P'.

Assuming that θ does not have subterms of the form $\langle \theta_1, \theta_2 \rangle_L$, meaning that $X_P \xrightarrow{act(\theta)}_{C_P} X_{P'}$ originated from a single action, we define the notion of process context $\mathrm{C}[\bullet] = ctx(\theta)$ by induction on the syntactical structure of $\theta \in \Theta'$ as follows:

$$ctx(\theta) \;=\; \begin{cases} \bullet & \text{if } \theta \in \mathcal{A} \\ a^\dagger .\, ctx(\theta') & \text{if } \theta = \cdot_a \theta' \\ ctx(\theta') + Q & \text{if } \theta = \dot{+}_Q \theta' \text{ where } init(Q) \\ Q + ctx(\theta') & \text{if } \theta = \underset{.}{+}_Q \theta' \text{ where } init(Q) \\ ctx(\theta') \parallel_L Q & \text{if } \theta = \rVert_{L,Q} \theta' \\ Q \parallel_L ctx(\theta') & \text{if } \theta = \lVert_{L,Q} \theta' \end{cases}$$

Since X_P is the set of proof terms labeling a sequence of proved transitions from $to\_init(P)$ to P, from $X_P \xrightarrow{act(\theta)}_{C_P} X_{P'}$ it follows that P must contain an occurrence of $act(\theta)$ in an initial subprocess of the form $act(\theta) \,.\, \bar{P}$. Then $P = \mathrm{C}[act(\theta) \,.\, \bar{P}]$ with $act(\theta) \,.\, \bar{P} \xrightarrow{act(\theta)} act(\theta)^\dagger .\, \bar{P}$ and hence $P = \mathrm{C}[act(\theta) \,.\, \bar{P}] \xrightarrow{\theta} \mathrm{C}[act(\theta)^\dagger .\, \bar{P}] = P'$.

If θ has subterms of the form $\langle \theta_1, \theta_2 \rangle_L$, then we proceed in a similar way by constructing a context with as many •-holes as there are synchronizing subprocesses. The following clause has to be added to the inductive definition of process context: $ctx(\theta) = ctx(\theta_1) \parallel_L ctx(\theta_2)$ if $\theta = \langle \theta_1, \theta_2 \rangle_L$. ■

Similar to Theorem 2, Theorem 4 holds under the assumption that, in the presence of autoconcurrency, for each maximal set of conflicting actions (i.e., actions inside subprocesses composed by nondeterministic choice) no two actions in the set occur in other two subprocesses that are composed in parallel – remember that events are unique while every action may have several occurrences. The two configuration structures in Figure 12 of [32], which come from the two event structures in Figure 11 of the same paper, are respectively expressed by the two processes (with a final relabeling $a_i \mapsto a$ and $b_i \mapsto b$):

$$((a_1 \,.\, b_1 \,.\, \underline{0}) \parallel_\emptyset (a_2 \,.\, b_2 \,.\, \underline{0}) \parallel_\emptyset (a_3 \,.\, b_3 \,.\, \underline{0}))$$
$$\parallel_{\{a_1, a_3, b_1, b_2, b_3\}}$$
$$((a_1 \,.\, \underline{0} + a_3 \,.\, \underline{0}) \parallel_\emptyset (b_1 \,.\, \underline{0} + b_2 \,.\, \underline{0}) \parallel_{\{b_2\}} (b_2 \,.\, \underline{0} + b_3 \,.\, \underline{0}))$$

and:

$$((a_1 \,.\, b_1 \,.\, \underline{0}) \parallel_\emptyset (a_2 \,.\, b_2 \,.\, \underline{0}) \parallel_\emptyset (a_3 \,.\, b_3 \,.\, \underline{0}))$$
$$\parallel_{\{a_1, a_3, b_1, b_2\}}$$
$$((a_1 \,.\, \underline{0} + a_3 \,.\, \underline{0}) \parallel_{\{a_3\}} (a_3 \,.\, \underline{0} + b_2 \,.\, \underline{0}) \parallel_{\{b_2\}} (b_1 \,.\, \underline{0} + b_2 \,.\, \underline{0}))$$

which do not satisfy the assumption. Below is the revised proof.

Proof of Theorem 4. The proof is divided into two parts:

- Suppose that $[\![P_1]\!] \sim_{\mathrm{HHPB}} [\![P_2]\!]$. Then $P_1 \sim_{\mathrm{FRB:brm}} P_2$ follows by proving that the symmetric relation $\mathcal{B} = \{(Q_1, Q_2) \mid [\![Q_1]\!] \sim_{\mathrm{HHPB}} [\![Q_2]\!]\}$ is a brm-forward-reverse bisimulation. Let $(Q_1, Q_2) \in \mathcal{B}$, so that $[\![Q_1]\!] \sim_{\mathrm{HHPB}} [\![Q_2]\!]$:
  - If $Q_1 \xrightarrow{\theta_1} Q'_1$ then $X_{Q_1} \xrightarrow{act(\theta_1)}_{C_{Q_1}} X_{Q'_1}$ due to Lemma 1, hence $X_{Q_2} \xrightarrow{act(\theta_2)}_{C_{Q_2}} X_{Q'_2}$ with $act(\theta_1) = act(\theta_2)$ and $[\![Q'_1]\!] \sim_{\mathrm{HHPB}} [\![Q'_2]\!]$ because $[\![Q_1]\!] \sim_{\mathrm{HHPB}} [\![Q_2]\!]$, from which it follows that $Q_2 \xrightarrow{\theta_2} Q'_2$ due to Lemma 1 with $(Q'_1, Q'_2) \in \mathcal{B}$.
  - If $Q'_1 \xrightarrow{\theta_1} Q_1$ then $X_{Q'_1} \xrightarrow{act(\theta_1)}_{C_{Q'_1}} X_{Q_1}$ due to Lemma 1, hence $X_{Q'_2} \xrightarrow{act(\theta_2)}_{C_{Q'_2}} X_{Q_2}$ with $act(\theta_1) = act(\theta_2)$ and $[\![Q'_1]\!] \sim_{\mathrm{HHPB}} [\![Q'_2]\!]$ because $[\![Q_1]\!] \sim_{\mathrm{HHPB}} [\![Q_2]\!]$, from which it follows that $Q'_2 \xrightarrow{\theta_2} Q_2$ due to Lemma 1 with $(Q'_1, Q'_2) \in \mathcal{B}$.
  - The stable configuration structure semantics preserves actions inside transition labels (Lemma 1), uniquely identifies different occurrences of the same action in a process via as many different proof terms in the set of events, and is able to distinguish between causality and concurrency like the proved operational semantics of Table 1. Therefore, from $[\![Q_1]\!] \sim_{\mathrm{HHPB}} [\![Q_2]\!]$ – in particular the incoming transition matching between X_{Q_1} and X_{Q_2} and the labeling- and causality-preserving bijection from X_{Q_1} to X_{Q_2} – it follows that there must be a one-to-one correspondence between the incoming proved transitions of Q_1 and the incoming proved transitions of Q_2, hence $brm(Q_1) = brm(Q_2)$.

– Suppose that $P_1 \sim_{\mathrm{FRB:brm}} P_2$. Then the existence of a sequence of proved transitions $\mathit{to\_init}(P_1) = P_{1,1} \xrightarrow{\theta_{P_{1,1}}} P_{1,2} \dots P_{1,n} \xrightarrow{\theta_{P_{1,n}}} P_1$ implies the existence of a sequence of proved transitions $\mathit{to\_init}(P_2) = P_{2,1} \xrightarrow{\theta_{P_{2,1}}} P_{2,2} \dots P_{2,n} \xrightarrow{\theta_{P_{2,n}}} P_2$ such that $\mathit{act}(\theta_{P_{1,h}}) = \mathit{act}(\theta_{P_{2,h}})$ and $P_{1,h} \sim_{\mathrm{FRB:brm}} P_{2,h}$ for all $h = 1, \dots, n$, and vice versa. Note that $\theta_{P_{1,h}} \neq \theta_{P_{1,k}}$ and $\theta_{P_{2,h}} \neq \theta_{P_{2,k}}$ for all $h \neq k$ because different occurrences of the same action in a process are identified by different proof terms.
Thus $[\![P_1]\!] \sim_{\mathrm{HHPB}} [\![P_2]\!]$ follows by proving that $\mathcal{B} = \{(X_{Q_1}, X_{Q_2}, \{(\theta_{Q_{1,h}}, \theta_{Q_{2,h}}) \mid h \in H\}) \mid Q_1 \sim_{\mathrm{FRB:brm}} Q_2 \wedge \mathit{to\_init}(Q_i) = Q_{i,1} \xrightarrow{\theta_{Q_{i,1}}} Q_{i,2} \dots Q_{i,|H|} \xrightarrow{\theta_{Q_{i,|H|}}} Q_i$ for $i \in \{1, 2\} \wedge \mathit{act}(\theta_{Q_{1,h}}) = \mathit{act}(\theta_{Q_{2,h}})$ for all $h \in H \wedge Q_{1,h} \sim_{\mathrm{FRB:brm}} Q_{2,h}$ for all $h \in H\}$ is a hereditary-history preserving bisimulation. Observing that $Q_1 \sim_{\mathrm{FRB:brm}} Q_2$ implies $(\emptyset, \emptyset, \emptyset) \in \mathcal{B}$ when Q_1 and Q_2 are both initial, take $(X_{Q_1}, X_{Q_2}, \{(\theta_{Q_{1,h}}, \theta_{Q_{2,h}}) \mid h \in H\}) \in \mathcal{B}$, so that $Q_1 \sim_{\mathrm{FRB:brm}} Q_2$:

- If $X_{Q_1} \xrightarrow{\mathit{act}(\theta_1)}_{\mathsf{c}_{Q_1}} X_{Q'_1}$ then $Q_1 \xrightarrow{\theta_1} Q'_1$ due to Lemma 1, hence $Q_2 \xrightarrow{\theta_2} Q'_2$ with $\mathit{act}(\theta_1) = \mathit{act}(\theta_2)$ and $Q'_1 \sim_{\mathrm{FRB:brm}} Q'_2$ because $Q_1 \sim_{\mathrm{FRB:brm}} Q_2$, from which it follows that $X_{Q_2} \xrightarrow{\mathit{act}(\theta_2)}_{\mathsf{c}_{Q_2}} X_{Q'_2}$ due to Lemma 1. Since $\theta_1 \notin X_{Q_1}$ and $\theta_2 \notin X_{Q_2}$, it holds that $(X_{Q'_1}, X_{Q'_2}, \{(\theta_{Q_{1,h}}, \theta_{Q_{2,h}}) \mid h \in H\} \cup \{(\theta_1, \theta_2)\}) \in \mathcal{B}$.
  If we start from $X_{Q_2} \xrightarrow{\mathit{act}(\theta_2)}_{\mathsf{c}_{Q_2}} X_{Q'_2}$, then we reason in the same way.
- If $X_{Q'_1} \xrightarrow{\mathit{act}(\theta_1)}_{\mathsf{c}_{Q'_1}} X_{Q_1}$ then $Q'_1 \xrightarrow{\theta_1} Q_1$ due to Lemma 1, hence $Q'_2 \xrightarrow{\theta_2} Q_2$ with $\mathit{act}(\theta_1) = \mathit{act}(\theta_2)$ and $Q'_1 \sim_{\mathrm{FRB:brm}} Q'_2$ because $Q_1 \sim_{\mathrm{FRB:brm}} Q_2$, from which it follows that $X_{Q'_2} \xrightarrow{\mathit{act}(\theta_2)}_{\mathsf{c}_{Q'_2}} X_{Q_2}$ due to Lemma 1. Since $\theta_1 \notin X_{Q'_1}$, $\theta_2 \notin X_{Q'_2}$, and $\mathit{brm}(Q_1) = \mathit{brm}(Q_2)$, the latter transition can be selected in such a way to satisfy $\{(\theta_{Q_{1,h}}, \theta_{Q_{2,h}}) \mid h \in H\} \upharpoonright X_{Q'_1} = \{(\theta_{Q_{1,h}}, \theta_{Q_{2,h}}) \mid h \in H\} \setminus \{(\theta_1, \theta_2)\}$, hence $(X_{Q'_1}, X_{Q'_2}, \{(\theta_{Q_{1,h}}, \theta_{Q_{2,h}}) \mid h \in H\} \setminus \{(\theta_1, \theta_2)\}) \in \mathcal{B}$.
  If we start from $X_{Q'_2} \xrightarrow{\mathit{act}(\theta_2)}_{\mathsf{c}_{Q'_2}} X_{Q_2}$, then we reason in the same way.
- $f = \{(\theta_{Q_{1,h}}, \theta_{Q_{2,h}}) \mid h \in H\}$ certainly is a bijection from X_{Q_1} to X_{Q_2} – as the events along either computation are different from each other, so the two reached configurations X_{Q_1} and X_{Q_2} contain the same number of events, and paired in a stepwise manner – that preserves labeling – by definition of $\mathcal{B}$. If $|H| \leq 1$ then causality is trivially preserved.
  Suppose that X_{Q_1} and X_{Q_2} break causality and, among all the pairs of configurations associated with $\sim_{\mathrm{FRB:brm}}$-equivalent processes that break causality, they are the closest ones to $\emptyset$ and $\emptyset$ (in terms of number of transitions to be executed from either empty configuration). The rest of the proof is like the one of the corresponding part of the proof of Theorem 2, with Lemma 1 being exploited as well. ∎

Proof of Theorem 5. We proceed by induction on $k = \mathit{depth}(\phi)$:

- If $k = 0$ then ϕ is true and both P and $[\![P]\!]$ satisfy it.
- If $k \geq 1$ then there are five cases:
  - If ϕ is $\neg\phi'$ then by the induction hypothesis:
$$\begin{aligned} P \models_\rho \neg\phi' \;&\text{iff}\; P \not\models_\rho \phi' \\ &\text{iff}\; [\![P]\!] \not\models_\rho \phi' \\ &\text{iff}\; [\![P]\!] \models_\rho \neg\phi' \end{aligned}$$
  - If ϕ is $\phi_1 \wedge \phi_2$ then by the induction hypothesis:
$$\begin{aligned} P \models_\rho \phi_1 \wedge \phi_2 \;&\text{iff}\; P \models_\rho \phi_1 \text{ and } P \models_\rho \phi_2 \\ &\text{iff}\; [\![P]\!] \models_\rho \phi_1 \text{ and } [\![P]\!] \models_\rho \phi_2 \\ &\text{iff}\; [\![P]\!] \models_\rho \phi_1 \wedge \phi_2 \end{aligned}$$
  - If ϕ is $\langle x : a\rangle\!\rangle\phi'$ then by Lemma 1 and the induction hypothesis:
$$\begin{aligned} P \models_\rho \langle x : a\rangle\!\rangle\phi' \;&\text{iff there is } P \xrightarrow{\theta} P' \text{ s.t. } \mathit{act}(\theta) = a \text{ and } P' \models_{\rho[x\mapsto\theta]} \phi' \\ &\text{iff there is } X_P \xrightarrow{\mathit{act}(\theta)}_{\mathsf{C}_P} X_{P'} \text{ s.t. } \mathit{act}(\theta) = a \text{ and } [\![P']\!] \models_{\rho[x\mapsto\theta]} \phi' \\ &\text{iff } [\![P]\!] \models_\rho \langle x : a\rangle\!\rangle\phi' \end{aligned}$$
  - If ϕ is $(x : a)\phi'$ then by the induction hypothesis:
$$\begin{aligned} P \models_\rho (x : a)\phi' \;&\text{iff there is } a^\dagger. P' \in \mathit{sp}(P) \text{ s.t. } P \models_{\rho[x\mapsto \mathit{apt}(a^\dagger . P', P)]} \phi' \\ &\text{iff there is } \theta \in X_P \text{ s.t. } \theta = \mathit{apt}(a^\dagger. P', P) \text{ and } [\![P]\!] \models_{\rho[x\mapsto \mathit{apt}(a^\dagger . P', P)]} \phi' \\ &\text{iff there is } \theta \in X_P \text{ s.t. } \mathit{act}(\theta) = a \text{ and } [\![P]\!] \models_{\rho[x\mapsto\theta]} \phi' \\ &\text{iff } [\![P]\!] \models_\rho (x : a)\phi' \end{aligned}$$
  - If ϕ is $\langle\!\langle x\rangle\phi'$ then by Lemma 1 and the induction hypothesis:
$$\begin{aligned} P \models_\rho \langle\!\langle x\rangle\phi' \;&\text{iff there is } P' \xrightarrow{\theta} P \text{ s.t. } \rho(x) = \theta \text{ and } P' \models_\rho \phi' \\ &\text{iff there is } X_{P'} \xrightarrow{\mathit{act}(\theta)}_{\mathsf{C}_{P'}} X_P \text{ s.t. } \rho(x) = \theta \text{ and } [\![P']\!] \models_\rho \phi' \\ &\text{iff } [\![P]\!] \models_\rho \langle\!\langle x\rangle\phi' \end{aligned}$$
■

Proof of Lemma 2. We proceed by induction on $k = \mathit{depth}(\phi)$:

- If $k = 0$ then ϕ must be true. Since true is closed, we have that $\sigma(\mathsf{true}) = \mathsf{true}$ and $\rho^\sigma = \rho$, hence the result trivially follows.
- If $k \geq 1$ then there are five cases:
  - If ϕ is $\neg\phi'$ then by the induction hypothesis:
$$\begin{aligned} P \models_\rho \neg\phi' \;&\text{iff}\; P \not\models_\rho \phi' \\ &\text{iff}\; P \not\models_{\rho^\sigma} \sigma(\phi') \\ &\text{iff}\; P \models_{\rho^\sigma} \sigma(\neg\phi') \end{aligned}$$
  - If ϕ is $\phi_1 \wedge \phi_2$ then by the induction hypothesis:
$$\begin{aligned} P \models_\rho \phi_1 \wedge \phi_2 \;&\text{iff}\; P \models_\rho \phi_1 \text{ and } P \models_\rho \phi_2 \\ &\text{iff}\; P \models_{\rho^{\sigma_1}} \sigma_1(\phi_1) \text{ and } P \models_{\rho^{\sigma_2}} \sigma_2(\phi_2) \\ &\text{iff}\; P \models_{\rho^\sigma} \sigma(\phi_1) \text{ and } P \models_{\rho^\sigma} \sigma(\phi_2) \\ &\text{iff}\; P \models_{\rho^\sigma} \sigma(\phi_1 \wedge \phi_2) \end{aligned}$$
    provided that $\sigma_1(x) = \sigma_2(x)$ for all $x \in \mathit{fi}(\phi_1) \cap \mathit{fi}(\phi_2)$ and $\sigma \upharpoonright \mathit{fi}(\phi_k) = \sigma_k$ for $k \in \{1, 2\}$.

- If ϕ is $\langle x : a \rangle\!\rangle \phi'$ then by the induction hypothesis:

$$\begin{aligned} P \models_\rho \langle x : a \rangle\!\rangle \phi' \;\; & \text{iff there is } P \xrightarrow{\theta} P' \text{ s.t. } act(\theta) = a \text{ and } P' \models_{\rho[x \mapsto \theta]} \phi' \\ & \text{iff there is } P \xrightarrow{\theta} P' \text{ s.t. } act(\theta) = a \text{ and } P' \models_{(\rho[x \mapsto \theta])^{\sigma'}} \sigma'(\phi') \\ & \text{iff there is } P \xrightarrow{\theta} P' \text{ s.t. } act(\theta) = a \text{ and } P' \models_{\rho^\sigma[x \mapsto \theta]} \sigma(\phi') \\ & \text{iff } P \models_{\rho^\sigma} \sigma(\langle x : a \rangle\!\rangle \phi') \end{aligned}$$

provided that $\sigma = \sigma'$ if $x \notin \mathit{fi}(\phi')$ while $\sigma[x \mapsto \sigma'(x)] = \sigma'$ if $x \in \mathit{fi}(\phi')$.

- If ϕ is $(x : a)\phi'$ then by the induction hypothesis:

$$\begin{aligned} P \models_\rho (x : a)\phi' \;\; & \text{iff there is } a^\dagger . P' \in sp(P) \text{ s.t. } P \models_{\rho[x \mapsto apt(a^\dagger . P', P)]} \phi' \\ & \text{iff there is } a^\dagger . P' \in sp(P) \text{ s.t. } P \models_{(\rho[x \mapsto apt(a^\dagger . P', P)])^{\sigma'}} \sigma'(\phi') \\ & \text{iff there is } a^\dagger . P' \in sp(P) \text{ s.t. } P \models_{\rho^\sigma[x \mapsto apt(a^\dagger . P', P)]} \sigma(\phi') \\ & \text{iff } P \models_{\rho^\sigma} \sigma((x : a)\phi') \end{aligned}$$

provided that $\sigma = \sigma'$ if $x \notin \mathit{fi}(\phi')$ while $\sigma[x \mapsto \sigma'(x)] = \sigma'$ if $x \in \mathit{fi}(\phi')$.

- If ϕ is $\langle\!\langle x \rangle \phi'$ then by the induction hypothesis:

$$\begin{aligned} P \models_\rho \langle\!\langle x \rangle \phi' \;\; & \text{iff there is } P' \xrightarrow{\theta} P \text{ s.t. } \rho(x) = \theta \text{ and } P' \models_\rho \phi' \\ & \text{iff there is } P' \xrightarrow{\theta} P \text{ s.t. } \rho^{\sigma'}(\sigma'(x)) = \theta \text{ and } P' \models_{\rho^{\sigma'}} \sigma'(\phi') \\ & \text{iff there is } P' \xrightarrow{\theta} P \text{ s.t. } \rho^{\sigma}(\sigma(x)) = \theta \text{ and } P' \models_{\rho^\sigma} \sigma(\phi') \\ & \text{iff } P \models_{\rho^\sigma} \sigma(\langle\!\langle x \rangle \phi') \end{aligned}$$

as $\rho^{\sigma'}(\sigma'(x)) = \rho(x)$, provided that $\sigma \setminus \{(x, \sigma(x))\} = \sigma'$ if $x \notin \mathit{fi}(\phi')$ while $\sigma = \sigma'$ if $x \in \mathit{fi}(\phi')$. ■

Proof of Corollary 1. The proof is divided into two parts:

– Assuming that $P_1 \sim_{\mathrm{FRB:brm}} P_2$, we observe that the existence of a sequence of proved transitions $\mathit{to\_init}(P_1) \xrightarrow{\theta_{P_1,1}} \dots \xrightarrow{\theta_{P_1,n}} P_1$ implies the existence of a sequence of proved transitions $\mathit{to\_init}(P_2) \xrightarrow{\theta_{P_2,1}} \dots \xrightarrow{\theta_{P_2,n}} P_2$ such that $act(\theta_{P_1,h}) = act(\theta_{P_2,h})$ for all $h = 1, \dots, n$, and vice versa, where $\{\theta_{P_1,h} \mid 1 \leq h \leq n\} = X_{P_1}$ and $\{\theta_{P_2,h} \mid 1 \leq h \leq n\} = X_{P_2}$. Note that $n = 0$ when P_1 and P_2 are both initial; moreover $\theta_{P_1,h} \neq \theta_{P_1,k}$ and $\theta_{P_2,h} \neq \theta_{P_2,k}$ for all $h \neq k$. Let $f_{1,2} = \{(\theta_{P_1,h}, \theta_{P_2,h}) \mid 1 \leq h \leq n\}$, which clearly is a label-preserving bijection from X_{P_1} to X_{P_2}.
  We proceed by induction on $k = \mathit{depth}(\phi)$:
  - If $k = 0$ then ϕ must be true, which is trivially satisfied by P_1 and P_2 regardless of their respective permissible environments.
  - If $k \geq 1$ then there are five cases:
    * If ϕ is $\neg\phi'$ then by the induction hypothesis:

$$\begin{aligned} P_1 \models_\rho \neg\phi' \;\; & \text{iff } P_1 \not\models_\rho \phi' \\ & \text{iff } P_2 \not\models_{f_{1,2} \circ \rho} \phi' \\ & \text{iff } P_2 \models_{f_{1,2} \circ \rho} \neg\phi' \end{aligned}$$

    * If ϕ is $\phi_1 \wedge \phi_2$ then by the induction hypothesis:

$$\begin{aligned} P_1 \models_\rho \phi_1 \wedge \phi_2 \;\; & \text{iff } P_1 \models_\rho \phi_1 \text{ and } P_1 \models_\rho \phi_2 \\ & \text{iff } P_2 \models_{f_{1,2} \circ \rho} \phi_1 \text{ and } P_2 \models_{f_{1,2} \circ \rho} \phi_2 \\ & \text{iff } P_2 \models_{f_{1,2} \circ \rho} \phi_1 \wedge \phi_2 \end{aligned}$$

  - If ϕ is $\langle x : a \rangle\!\rangle \phi'$ then by $P_1 \sim_{\mathrm{FRB:brm}} P_2$ and the induction hypothesis:
    $P_1 \models_\rho \langle x : a \rangle\!\rangle \phi'$ iff there is $P_1 \xrightarrow{\theta_1} P'_1$ s.t. $act(\theta_1) = a$ and $P'_1 \models_{\rho[x \mapsto \theta_1]} \phi'$
    iff there is $P_2 \xrightarrow{\theta_2} P'_2$ s.t. $act(\theta_2) = a$ and $P'_2 \models_{f'_{1,2} \circ \rho[x \mapsto \theta_1]} \phi'$
    iff $P_2 \models_{f_{1,2} \circ \rho} \langle x : a \rangle\!\rangle \phi'$
    provided that $f_{1,2} \cup \{(\theta_1, \theta_2)\} = f'_{1,2}$.
  - If ϕ is $(x : a)\phi'$ then by $P_1 \sim_{\mathrm{FRB:brm}} P_2$ and the induction hypothesis:
    $P_1 \models_\rho (x : a)\phi'$ iff there is $a^\dagger . P'_1 \in sp(P_1)$ s.t. $P_1 \models_{\rho[x \mapsto apt(a^\dagger . P'_1, P_1)]} \phi'$
    iff there is $a^\dagger . P'_2 \in sp(P_2)$ s.t. $P_2 \models_{f_{1,2} \circ \rho[x \mapsto apt(a^\dagger . P'_1, P_1)]} \phi'$
    iff $P_2 \models_{f_{1,2} \circ \rho} (x : a)\phi'$
  - If ϕ is $\langle\!\langle x \rangle \phi'$ then by $P_1 \sim_{\mathrm{FRB:brm}} P_2$ and the induction hypothesis:
    $P_1 \models_\rho \langle\!\langle x \rangle \phi'$ iff there is $P'_1 \xrightarrow{\theta_1} P_1$ s.t. $\rho(x) = \theta_1$ and $P'_1 \models_\rho \phi'$
    iff there is $P'_2 \xrightarrow{\theta_2} P_2$ s.t. $\rho(x) = \theta_2$ and $P'_2 \models_{f'_{1,2} \circ \rho} \phi'$
    iff $P_2 \models_{f_{1,2} \circ \rho} \langle\!\langle x \rangle \phi'$
    provided that $f_{1,2} = f'_{1,2} \cup \{(\theta_1, \theta_2)\}$.

– Assuming that there exists a label-preserving bijection from X_{P_1} to X_{P_2} such that P_1 and P_2 satisfy the same formulas of $\mathcal{L}_{\mathrm{EI}}$ under suitable permissible environments related by the aforementioned bijection, the result follows by proving that the symmetric relation $\mathcal{B} = \{(Q_1, Q_2) \mid \exists f_{1,2} . \forall \phi \in \mathcal{L}_{\mathrm{EI}} . \forall \rho \in pe(Q_1, \phi) . Q_1 \models_\rho \phi \iff Q_2 \models_{f_{1,2} \circ \rho} \phi$ where $f_{1,2}$ is a label-preserving bijection from X_{Q_1} to $X_{Q_2}\}$ is a brm-forward-reverse bisimulation.
Given $(Q_1, Q_2) \in \mathcal{B}$:

  - If $Q_1 \xrightarrow{\theta_1} Q'_1$ suppose by contradiction that there is no Q'_2 satisfying the same formulas as Q'_1 for some label-preserving bijection $f'_{1,2}$ from $X_{Q'_1}$ to $X_{Q'_2}$ such that $Q_2 \xrightarrow{\theta_2} Q'_2$ and $act(\theta_1) = act(\theta_2)$, i.e., $(Q'_1, Q'_2) \in \mathcal{B}$ for no Q'_2 $act(\theta_1)$-reachable from Q_2. Since Q_2 has finitely many outgoing transitions, the set of processes that Q_2 can reach by performing an $act(\theta_1)$-transition is finite, say $\{Q'_{2,1}, \ldots, Q'_{2,n}\}$ with $n \geq 0$. Since none of the processes in the set satisfies the same formulas as Q'_1, for each $1 \leq i \leq n$ there exists $\phi_i \in \mathcal{L}_{\mathrm{EI}}$ such that $Q'_1 \models_{\rho_i} \phi_i$ but $Q'_{2,i} \not\models_{f'_{1,2,i} \circ \rho_i} \phi_i$ for all label-preserving bijections $f'_{1,2,i}$ from $X_{Q'_1}$ to $X_{Q'_{2,i}}$.
    Since the formulas $\phi_1, \ldots, \phi_n$ may contain different identifiers, let $\{z_\theta \mid \theta \in X_{Q'_1}\}$ be a set of fresh identifiers different from each other and the related environment ρ' be defined by $\rho'(z_\theta) = \theta$ for all $\theta \in X_{Q'_1}$. Also let every substitution σ_i be defined by $\sigma_i(x) = z_{\rho'(x)}$ for all $x \in fi(\phi_i)$, so that $\rho_i(x) = \rho'(\sigma_i(x))$. It holds that $Q'_1 \models_{\rho'} \sigma_i(\phi_i)$ by Lemma 2 and $Q'_{2,i} \not\models_{f'_{1,2,i} \circ \rho'} \sigma_i(\phi_i)$ for all label-preserving bijections $f'_{1,2,i}$ from $X_{Q'_1}$ to $X_{Q'_{2,i}}$.
    We can then construct the formula $\langle z_{\theta_1} : act(\theta_1) \rangle\!\rangle \bigwedge_{i=1}^{n} \sigma_i(\phi_i)$ that is satisfied by Q_1 under ρ such that $\rho[z_{\theta_1} \mapsto \theta_1] = \rho'$ but not by Q_2 under $f_{1,2} \circ \rho$ for all label-preserving bijections $f_{1,2}$ from X_{Q_1} to X_{Q_2}; if $n = 0$

then it is sufficient to take $\langle z_{\theta_1} : act(\theta_1)\rangle\rangle\mathsf{true}$. This formula violates $(Q_1, Q_2) \in \mathcal{B}$, hence there must exist at least one Q'_2 satisfying the same formulas as Q'_1 for some label-preserving bijection $f'_{1,2}$ from $X_{Q'_1}$ to $X_{Q'_2}$ such that $Q_2 \xrightarrow{\theta_2} Q'_2$ and $act(\theta_1) = act(\theta_2)$, so that $(Q'_1, Q'_2) \in \mathcal{B}$.

- If $Q'_1 \xrightarrow{\theta_1} Q_1$ suppose by contradiction that there is no Q'_2 satisfying the same formulas as Q'_1 for some label-preserving bijection $f'_{1,2}$ from $X_{Q'_1}$ to $X_{Q'_2}$ such that $Q'_2 \xrightarrow{\theta_2} Q_2$ and $act(\theta_1) = act(\theta_2)$, i.e., $(Q'_1, Q'_2) \in \mathcal{B}$ for no Q'_2 $act(\theta_1)$-reaching Q_2. Since Q_2 has finitely many incoming transitions, the set of processes that can reach Q_2 by performing an $act(\theta_1)$-transition is finite, say $\{Q'_{2,1}, \dots, Q'_{2,n}\}$ with $n \geq 0$. Since none of the processes in the set satisfies the same formulas as Q'_1, for each $1 \leq i \leq n$ there exists $\phi_i \in \mathcal{L}_{\rm EI}$ such that $Q'_1 \models_{\rho_i} \phi_i$ but $Q'_{2,i} \not\models_{f'_{1,2,i} \circ \rho_i} \phi_i$.
Since the formulas $\phi_1, \dots, \phi_n$ may contain different identifiers, let $\{z_\theta \mid \theta \in X_{Q'_1}\}$ be a set of fresh identifiers different from each other and the related environment ρ' be defined by $\rho'(z_\theta) = \theta$ for all $\theta \in X_{Q'_1}$. Also let every substitution σ_i be defined by $\sigma_i(x) = z_{\rho'(x)}$ for all $x \in \mathit{fi}(\phi_i)$, so that $\rho_i(x) = \rho'(\sigma_i(x))$. It holds that $Q'_1 \models_{\rho'} \sigma_i(\phi_i)$ by Lemma 2 and $Q'_{2,i} \not\models_{f'_{1,2,i} \circ \rho'} \sigma_i(\phi_i)$ for all label-preserving bijections $f'_{1,2,i}$ from $X_{Q'_1}$ to $X_{Q'_{2,i}}$.
We can then construct the formula $\langle\langle z_{\theta_1}\rangle \bigwedge_{i=1}^{n} \sigma_i(\phi_i)$ that is satisfied by Q_1 under $\rho = \rho'$ but not by Q_2 under $f_{1,2} \circ \rho$ for all label-preserving bijections $f_{1,2}$ from X_{Q_1} to X_{Q_2}; if $n = 0$ then it is sufficient to take $\langle z_{\theta_1}\rangle\rangle\mathsf{true}$. This formula violates $(Q_1, Q_2) \in \mathcal{B}$, hence there must exist at least one Q'_2 satisfying the same formulas as Q'_1 for some label-preserving bijection $f'_{1,2}$ from $X_{Q'_1}$ to $X_{Q'_2}$ such that $Q'_2 \xrightarrow{\theta_2} Q_2$ and $act(\theta_1) = act(\theta_2)$, so that $(Q'_1, Q'_2) \in \mathcal{B}$.
- Suppose by contradiction that $brm(Q_1) \neq brm(Q_2)$. In order not to fall back into one of the previous cases, we assume that there is an action a with different nonzero multiplicities in $brm(Q_1)$ and $brm(Q_2)$. Without loss of generality, we further assume that a occurs with multiplicity 2 in $brm(Q_1)$ and 1 in $brm(Q_2)$.
We can then construct the formula $\langle\langle x\rangle\mathsf{true} \wedge \langle\langle y\rangle\mathsf{true}$ that is satisfied by Q_1 under ρ such that $act(\rho(x)) = act(\rho(y)) = a$ and $\rho(x) \neq \rho(y)$ but not by Q_2 under $f_{1,2} \circ \rho$ for all label-preserving bijections $f_{1,2}$ from X_{Q_1} to X_{Q_2}. This formula violates $(Q_1, Q_2) \in \mathcal{B}$, hence it must be the case that $brm(Q_1) = brm(Q_2)$. ■

Proof of Theorem 6. We proceed by induction on $k = depth(\phi)$:

– If $k = 0$ then there are two cases:

- If ϕ is true then both P and $[\![P]\!]$ satisfy it.

- If ϕ is M then by Lemma 1:

$$\begin{aligned} P \models M \ \text{ iff } & \ brm(P) = M \\ \text{iff } & \ \{\!|\ act(\theta) \mid P' \xrightarrow{\theta} P \,|\!\} = M \\ \text{iff } & \ \{\!|\ act(\theta) \mid X_{P'} \xrightarrow{act(\theta)}_{\mathsf{C}_{P'}} X_P \,|\!\} = M \\ \text{iff } & \ [\![P]\!] \models M \end{aligned}$$

– If $k \geq 1$ then there are four cases:

- If ϕ is $\neg\phi'$ then by the induction hypothesis:

$$\begin{aligned} P \models \neg\phi' \ \text{ iff } & \ P \not\models \phi' \\ \text{iff } & \ [\![P]\!] \not\models \phi' \\ \text{iff } & \ [\![P]\!] \models \neg\phi' \end{aligned}$$

- If ϕ is $\phi_1 \wedge \phi_2$ then by the induction hypothesis:

$$\begin{aligned} P \models \phi_1 \wedge \phi_2 \ \text{ iff } & \ P \models \phi_1 \text{ and } P \models \phi_2 \\ \text{iff } & \ [\![P]\!] \models \phi_1 \text{ and } [\![P]\!] \models \phi_2 \\ \text{iff } & \ [\![P]\!] \models \phi_1 \wedge \phi_2 \end{aligned}$$

- If ϕ is $\langle a \rangle \phi'$ then by Lemma 1 and the induction hypothesis:

$$\begin{aligned} P \models \langle a \rangle \phi' \ \text{ iff } & \ \text{there is } P \xrightarrow{\theta} P' \text{ s.t. } act(\theta) = a \text{ and } P' \models \phi' \\ \text{iff } & \ \text{there is } X_P \xrightarrow{act(\theta)}_{\mathsf{C}_P} X_{P'} \text{ s.t. } act(\theta) = a \text{ and } [\![P']\!] \models \phi' \\ \text{iff } & \ [\![P]\!] \models \langle a \rangle \phi' \end{aligned}$$

- If ϕ is $\langle a^\dagger \rangle \phi'$ then by Lemma 1 and the induction hypothesis:

$$\begin{aligned} P \models \langle a^\dagger \rangle \phi' \ \text{ iff } & \ \text{there is } P' \xrightarrow{\theta} P \text{ s.t. } act(\theta) = a \text{ and } P' \models \phi' \\ \text{iff } & \ \text{there is } X_{P'} \xrightarrow{act(\theta)}_{\mathsf{C}_{P'}} X_P \text{ s.t. } act(\theta) = a \text{ and } [\![P']\!] \models \phi' \\ \text{iff } & \ [\![P]\!] \models \langle a^\dagger \rangle \phi' \end{aligned}$$

■

Proof of Corollary 2. From Theorems 4, 3 and 6 it follows that:

$$\begin{aligned} [\![P_1]\!] \sim_{\text{HHPB}} [\![P_2]\!] \ \text{ iff } & \ P_1 \sim_{\text{FRB:brm}} P_2 \\ \text{iff } & \ \forall \phi \in \mathcal{L}_{\text{BRM}}.\, P_1 \models \phi \Longleftrightarrow P_2 \models \phi \\ \text{iff } & \ \forall \phi \in \mathcal{L}_{\text{BRM}}.\, [\![P_1]\!] \models \phi \Longleftrightarrow [\![P_2]\!] \models \phi \end{aligned}$$

■

Proof of Theorem 7. We proceed by induction on $k = depth(\phi)$:

– If $k = 0$ then there are two cases:

- If ϕ is true then $\mathcal{T}_{\text{BE}}(\mathsf{true}, act(P), \varrho_n) = \mathsf{true}$ and P satisfies both formulas (the second one for all ϱ_n and ρ).
- If ϕ is M then we divide the proof into two parts. Starting from $P \models M$, we derive that $brm(P) = M$, hence for all $a_i \in supp(M)$ there exists $P'_{a_{i,k}} \xrightarrow{\theta_{a_{i,k}}} P$ for $1 \leq k \leq M(a_i)$ such that $act(\theta_{a_{i,k}}) = a_i$, with P having no other incoming transitions. If we consider any sequence of proved transitions $P_1 \xrightarrow{\theta_1} P_2 \xrightarrow{\theta_2} \ldots \xrightarrow{\theta_m} P_{m+1}$ such that P_1 is $to\_init(P)$ and P_{m+1} is P, all the proof terms $\theta_{a_{i,k}}$ appear in the sequence $\theta_1, \ldots, \theta_m$ because the transitions $P'_{a_{i,k}} \xrightarrow{\theta_{a_{i,k}}} P$ are all independent from each other. In the construction of ϱ_n, for all $a_i \in supp(M)$ we have to consider all the a_i-transitions along any path from $to\_init(P)$ to P. Therefore, we take

as n the number of proof terms θ_j in the sequence $\theta_1, \dots, \theta_m$ such that $act(\theta_j) = a_i$ for all $a_i \in supp(M)$. Since $brm(P) = M$, the total number of incoming transitions of P is $\sum_{a_i \in supp(M)} M(a_i)$, while the number of proof terms θ_j in the sequence $\theta_1, \dots, \theta_m$ such that $act(\theta_j) = a_i$ and $\theta_j \notin \{\theta_{a_{i,k}} \mid 1 \leq k \leq M(a_i)\}$ is $n - \sum_{a_i \in supp(M)} M(a_i)$. We can map all numbers between 1 and $n - \sum_{a_i \in supp(M)} M(a_i)$ to the pairs $(z_{i,h}, a_i)$ and all numbers between $n - \sum_{a_i \in supp(M)} M(a_i) + 1$ and n to the pairs $(x_{i,k}, a_i)$. At this point, we can construct ρ by mapping every $x_{i,k}$ to $\theta_{a_{i,k}}$ and every $z_{i,h}$ to $\theta_{a_{i,h}}$ such that there is no transition $P'_{a_{i,h}} \xrightarrow{\theta_{a_{i,h}}} P$.

It turns out that $P \models_\rho \mathcal{T}_{\mathrm{BE}}(M, act(P), \varrho_n) = \bigwedge_{a_i \in supp(M)} (\bigwedge_{k=1}^{M(a_i)} \langle\langle x_{i,k} \rangle\mathsf{true}$ $\wedge \bigwedge_{h=1}^{\sharp(a_i, \varrho_n) - M(a_i)} \neg \langle\langle z_{i,h} \rangle \mathsf{true}) \wedge \bigwedge_{b \in act(P) \setminus supp(M)} \neg (y : b) \langle\langle y \rangle \mathsf{true}$ as we now show by considering each of the three main conjunctions separately:

- $P \models_\rho \bigwedge_{k=1}^{M(a_i)} \langle\langle x_{i,k} \rangle \mathsf{true}$ because, from the way we have constructed ρ, we know that it maps every $x_{i,k}$ exactly to the proof term $\theta_{i,k}$ such that there exists a transition $P'_{i,k} \xrightarrow{\theta_{a_{i,k}}} P$ with $act(\theta_{a_{i,k}}) = a_i$. Moreover, every $P'_{i,k}$ trivially satisfies true.
- $P \models_\rho \bigwedge_{h=1}^{\sharp(a_i, \varrho_m) - M(a_i)} \neg \langle\langle z_{i,h} \rangle \mathsf{true}$ because, as in the previous case, we have constructed ρ in such a way that every $z_{i,h}$ is mapped to the proof term $\theta_{a_{i,h}}$ such that $act(\theta_{a_{i,h}}) = a_i$ and there is no transition $P'_{a_{i,h}} \xrightarrow{\theta_{a_{i,h}}} P$. Hence, $P \models_\rho \neg \langle\langle z_{i,h} \rangle \mathsf{true}$ for any $z_{i,h}$.
- $P \models_\rho \bigwedge_{b \in act(P) \setminus supp(M)} \neg (y : b) \langle\langle y \rangle \mathsf{true}$ because if $b \notin supp(M)$ then:
  - either there is no $b^\dagger . P' \in sp(P)$ and hence $P \not\models_\rho (y : b) \langle\langle y \rangle \mathsf{true}$, i.e., $P \models_\rho \neg (y : b) \langle\langle y \rangle \mathsf{true}$;
  - or, if such a process exists, P does not have any incoming transition $P' \xrightarrow{\theta} P$ such that $act(\theta) = b$ and hence $P \not\models_{\rho[y \mapsto \theta]} \langle\langle y \rangle \mathsf{true}$, from which it follows that $P \not\models_\rho (y : b) \langle\langle y \rangle \mathsf{true}$, i.e., $P \models_\rho \neg (y : b) \langle\langle y \rangle \mathsf{true}$.

The converse is straightforward because, if there exist ϱ_n and $\rho \in pe(\mathcal{T}_{\mathrm{BE}}(M, act(P), \varrho_n))$ such that $P \models_\rho \mathcal{T}_{\mathrm{BE}}(M, act(P), \varrho_n)$, then we can note that the conjunctions in $\mathcal{T}_{\mathrm{BE}}(M, act(P), \varrho_n)$ express the fact that P has exactly $\sum_{a_i \in supp(M)} M(a_i)$ incoming transitions such that every $a_i \in supp(M)$ appears $M(a_i)$ times, while there are no incoming transitions of P labeled with actions not in $supp(M)$. Thus, we can derive that $brm(P) = M$, i.e., $P \models M$.

– If $k \geq 1$ then there are four cases:

- If ϕ is $\neg\phi'$ then by the induction hypothesis:

$$\begin{aligned} P \models \neg\phi' \text{ iff } & P \not\models \phi' \\ \text{iff } & \exists \varrho_n . \exists \rho . P \not\models_\rho \mathcal{T}_{\mathrm{BE}}(\phi', act(P), \varrho_n) \\ \text{iff } & \exists \varrho_n . \exists \rho . P \models_\rho \neg\mathcal{T}_{\mathrm{BE}}(\phi', act(P), \varrho_n) \\ \text{iff } & \exists \varrho_n . \exists \rho . P \models_\rho \mathcal{T}_{\mathrm{BE}}(\neg\phi', act(P), \varrho_n) \end{aligned}$$

- If ϕ is $\phi_1 \wedge \phi_2$ then we divide the proof into two parts. Starting from $P \models \phi_1 \wedge \phi_2$, by the induction hypothesis and Lemma 2:

$$\begin{aligned} P \models \phi_1 \wedge \phi_2 \text{ implies } & P \models \phi_1 \text{ and } P \models \phi_2 \\ \text{implies } & \exists \varrho_{n_1} . \exists \rho_1 . P \models_{\rho_1} \mathcal{T}_{\mathrm{BE}}(\phi_1, act(P), \varrho_{n_1}) \text{ and } \exists \varrho_{n_2} . \exists \rho_2 . P \models_{\rho_2} \mathcal{T}_{\mathrm{BE}}(\phi_2, act(P), \varrho_{n_2}) \\ \text{implies } & \exists \varrho_{n_1} . \exists \varrho_{n_2} . \exists \rho^\sigma . P \models_{\rho^\sigma} \sigma(\mathcal{T}_{\mathrm{BE}}(\phi_1, act(P), \varrho_{n_1})) \text{ and } P \models_{\rho^\sigma} \sigma(\mathcal{T}_{\mathrm{BE}}(\phi_2, act(P), \varrho_{n_2})) \\ \text{implies } & \exists \varrho_{n_1} . \exists \varrho_{n_2} . \exists \rho^\sigma . P \models_{\rho^\sigma} \sigma(\mathcal{T}_{\mathrm{BE}}(\phi_1, act(P), \varrho_{n_1})) \wedge \sigma(\mathcal{T}_{\mathrm{BE}}(\phi_2, act(P), \varrho_{n_2})) \\ \text{implies } & \exists \varrho_n . \exists \rho^\sigma . P \models_{\rho^\sigma} \sigma(\mathcal{T}_{\mathrm{BE}}(\phi_1, act(P), \varrho_n)) \wedge \sigma(\mathcal{T}_{\mathrm{BE}}(\phi_2, act(P), \varrho_n)) \\ \text{implies } & \exists \varrho_n^\sigma . \exists \rho^\sigma . P \models_{\rho^\sigma} \mathcal{T}_{\mathrm{BE}}(\phi_1, act(P), \varrho_n^\sigma) \wedge \mathcal{T}_{\mathrm{BE}}(\phi_2, act(P), \varrho_n^\sigma) \\ \text{implies } & \exists \varrho_n^\sigma . \exists \rho^\sigma . P \models_{\rho^\sigma} \mathcal{T}_{\mathrm{BE}}(\phi_1 \wedge \phi_2, act(P), \varrho_n^\sigma) \end{aligned}$$

where:

  - ρ^σ maps a set fresh of identifiers z_θ, one for each θ appearing in any path from $to\_init(P)$ to P, to the corresponding proof term, i.e., $\rho^\sigma(z_\theta) = \theta$.
  - σ is defined as $\sigma(x) = z_{\rho_1(x)}$ for all $x \in fi(\mathcal{T}_{\mathrm{BE}}(\phi_1, act(P), \varrho_{n_1}))$ and $\sigma(x) = z_{\rho_2(x)}$ for all $x \in fi(\mathcal{T}_{\mathrm{BE}}(\phi_2, act(P), \varrho_{n_2}))$.
  - Since $P \models \phi_1$ and $P \models \phi_2$, both ϱ_{n_1} and ϱ_{n_2} have to be built consistently with any path from $to\_init(P)$ to P, hence they can be replaced by a single ϱ_n that is built in the same consistent way.
  - ϱ_n^σ is defined as $\varrho_n^\sigma(m) = (\sigma(x), a)$ for all $1 \leq m \leq n$ such that $\varrho_n(m) = (x, a)$ and $x \in fi(\mathcal{T}_{\mathrm{BE}}(\phi, act(P), \varrho_n))$. Since all free identifiers are captured by ϱ_n, $\sigma(\mathcal{T}_{\mathrm{BE}}(\phi, act(P), \varrho_n)) = \mathcal{T}_{\mathrm{BE}}(\phi, act(P), \varrho_n^\sigma)$.

As for the converse:

$$\begin{aligned} \exists \varrho_n . \exists \rho . P \models_\rho \mathcal{T}_{\mathrm{BE}}(\phi_1 \wedge \phi_2, act(P), \varrho_n) \text{ implies } & \exists \varrho_n . \exists \rho . P \models_\rho \mathcal{T}_{\mathrm{BE}}(\phi_1, act(P), \varrho_n) \wedge \mathcal{T}_{\mathrm{BE}}(\phi_2, act(P), \varrho_n) \\ \text{implies } & \exists \varrho_n . \exists \rho . P \models_\rho \mathcal{T}_{\mathrm{BE}}(\phi_1, act(P), \varrho_n) \text{ and } P \models_\rho \mathcal{T}_{\mathrm{BE}}(\phi_2, act(P), \varrho_n) \\ \text{implies } & P \models \phi_1 \text{ and } P \models \phi_2 \\ \text{implies } & P \models \phi_1 \wedge \phi_2 \end{aligned}$$

- If ϕ is $\langle a \rangle \phi'$ then by the induction hypothesis:

$$\begin{aligned} P \models \langle a \rangle \phi' \text{ iff } & \text{there is } P \xrightarrow{\theta} P' \text{ s.t. } act(\theta) = a \text{ and } P' \models \phi' \\ \text{iff } & \text{there is } P \xrightarrow{\theta} P' \text{ s.t. } act(\theta) = a \\ & \quad \text{and } \exists \varrho_n . \exists \rho . P' \models_{\rho[x \mapsto \theta]} \mathcal{T}_{\mathrm{BE}}(\phi', act(P'), \varrho_n \cup \{(n+1, (x, a))\}) \\ \text{iff } & \exists \varrho_n . \exists \rho . P \models_\rho \langle x : a \rangle\rangle \mathcal{T}_{\mathrm{BE}}(\phi', act(P), \varrho_n \cup \{(n+1, (x, a))\}) \\ \text{iff } & \exists \varrho_n . \exists \rho . P \models_\rho \mathcal{T}_{\mathrm{BE}}(\langle a \rangle \phi', act(P), \varrho_n) \end{aligned}$$

- If ϕ is $\langle a^\dagger \rangle \phi'$ then by the induction hypothesis:

$$\begin{aligned} P \models \langle a^\dagger \rangle \phi' \text{ iff } & \text{there is } P' \xrightarrow{\theta} P \text{ s.t. } act(\theta) = a \text{ and } P' \models \phi' \\ \text{iff } & \text{there is } P' \xrightarrow{\theta} P \text{ s.t. } act(\theta) = a \text{ and } \exists \varrho_n . \exists \rho . P' \models_{\rho[x \mapsto \theta]} \mathcal{T}_{\mathrm{BE}}(\phi', act(P'), \varrho_n) \\ \text{iff } & \text{there is } P' \xrightarrow{\theta} P \text{ s.t. } act(\theta) = a \text{ and } \exists \varrho_n . \exists \rho . P \models_{\rho[x \mapsto \theta]} \langle\langle x \rangle \mathcal{T}_{\mathrm{BE}}(\phi', act(P), \varrho_n) \\ \text{iff } & \text{there is } a^\dagger . P'' \in sp(P) \text{ s.t. } apt(a^\dagger . P'', P) = \theta \text{ and } P \models_{\rho[x \mapsto apt(a^\dagger . P'', P)]} \langle\langle x \rangle \mathcal{T}_{\mathrm{BE}}(\phi', act(P), \varrho_n) \\ \text{iff } & \exists \varrho_n . \exists \rho . P \models_\rho (x : a) \langle\langle x \rangle \mathcal{T}_{\mathrm{BE}}(\phi', act(P), \varrho_n) \\ \text{iff } & \exists \varrho_n . \exists \rho . P \models_\rho \mathcal{T}_{\mathrm{BE}}(\langle a^\dagger \rangle \phi', act(P), \varrho_n) \end{aligned}$$

■