Expansion Laws for Forward-Reverse, Forward, and Reverse Bisimilarities via Proved Encodings

Marco Bernardo   Andrea Esposito   Claudio A. Mezzina
Dipartimento di Scienze Pure e Applicate, Università di Urbino, Urbino, Italy

Reversible systems exhibit both forward computations and backward computations, where the aim of the latter is to undo the effects of the former. Such systems can be compared via forward-reverse bisimilarity as well as its two components, i.e., forward bisimilarity and reverse bisimilarity. The congruence, equational, and logical properties of these equivalences have already been studied in the setting of sequential processes. In this paper we address concurrent processes and investigate compositionality and axiomatizations of forward bisimilarity, which is interleaving, and reverse and forward-reverse bisimilarities, which are truly concurrent. To uniformly derive expansion laws for the three equivalences, we develop encodings based on the proved trees approach of Degano & Priami. In the case of reverse and forward-reverse bisimilarities, we show that in the encoding every action prefix needs to be extended with the backward ready set of the reached process.

1 Introduction

A reversible system features two directions of computation. The forward one coincides with the normal way of computing. The backward one undoes the effects of the forward one so as to return to a consistent state, i.e., a state that can be encountered while moving in the forward direction. Reversible computing has attracted an increasing interest due to its applications in many areas, including low-power computing [34, 6], program debugging [30, 38], robotics [40], wireless communications [53], fault-tolerant systems [23, 55, 35, 54], biochemical modeling [49, 50], and parallel discrete-event simulation [44, 52].

Returning to a consistent state is not an easy task to accomplish in a concurrent system, because the undo procedure necessarily starts from the last performed action and this may not be uniquely identifiable due to concurrency. The usually adopted strategy is that an action can be undone provided that all the actions it subsequently caused, if any, have been undone beforehand [22]. In this paper we focus on reversible process calculi, for which there are two approaches – later shown to be equivalent in [36] – to keep track of executed actions and revert computations in a causality-consistent way.

The dynamic approach of [22, 33] yielded RCCS (R for reversible) and its mobile variants [37, 21]. RCCS is an extension of CCS [41] that uses stack-based memories attached to processes so as to record executed actions and subprocesses discarded upon choices. A single transition relation is defined, while actions are divided into forward and backward thereby resulting in forward and backward transitions. This approach is adequate in the case of very expressive calculi as well as programming languages.

The static approach of [45] proposed a general method to reverse calculi, of which CCSK (K for keys) and its quantitative variants [10, 14, 11, 12] are a result. The idea is to retain within the process syntax all executed actions, which are suitably decorated, and all dynamic operators, which are thus made static. A forward transition relation and a backward transition relation are defined separately. Their labels are actions extended with communication keys so as to know, upon generating backward transitions, which actions synchronized with each other. This approach is very handy to deal with basic process calculi.

A systematic study of compositionality and axiomatization of strong bisimilarity in reversible process calculi has started in [13], both for nondeterministic processes and for Markovian processes. Then

To appear in EPTCS.

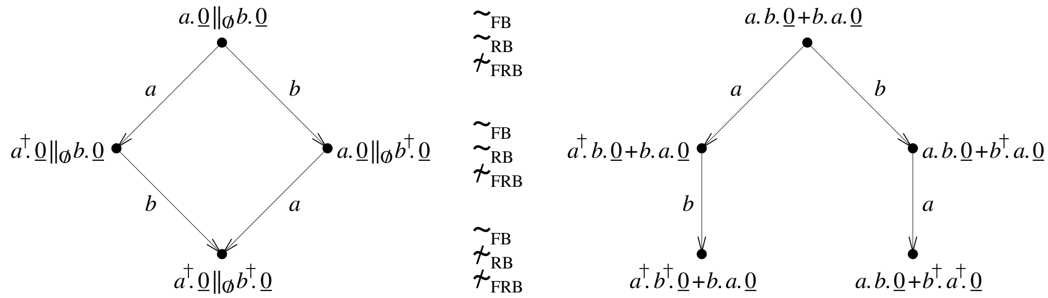


Figure 1: Forward, reverse, and forward-reverse bisimilarities at work: interleaving vs. true concurrency

compositionality and axiomatization of weak bisimilarity as well as modal logic characterizations for strong and weak bisimilarities have been investigated in [8, 9] for the nondeterministic case. That study compares the properties of forward-reverse bisimilarity $\sim_{\mathrm{FRB}}$ [45] with those of its two components, i.e., forward bisimilarity $\sim_{\mathrm{FB}}$ [43, 41] and reverse bisimilarity $\sim_{\mathrm{RB}}$. The reversible process calculus used in that study is minimal. Similar to [26], its semantics relies on a single transition relation, where the distinction between going forward or backward in the bisimulation game is made by matching outgoing or incoming transitions respectively. As a consequence, similar to [17] executed actions can be decorated uniformly, without having to resort to external stack-based memories [22] or communication keys associated with those actions [45].

A substantial limitation of [13, 8, 9] is the absence of the parallel composition operator in the calculus, motivated by the need of remaining neutral with respect to interleaving view vs. true concurrency. Unlike forward bisimilarity, as noted in [45] forward-reverse bisimilarity – and also reverse bisimilarity – does not satisfy the expansion law of parallel composition into a nondeterministic choice among all possible action sequencings. In Figure 1 we depict two labeled transition systems respectively representing a process that can perform action a in parallel with action b ($a\,.\,\underline{0} \,\|_{\emptyset}\, b\,.\,\underline{0}$ using a CSP-like parallel composition [19]) and a process that can perform either a followed by b or b followed by a ($a\,.\,b\,.\,\underline{0} + b\,.\,a\,.\,\underline{0}$ with $+$ denoting a CCS-like choice [41]), where $a \neq b$ and $\dagger$ decorates executed actions.

The forward bisimulation game yields the usual interleaving setting in which the two top states are related, the two pairs of corresponding intermediate states are related, and the three bottom states are related. However, the three bottom states are no longer related if we play the reverse bisimulation game, as the state on the left has two differently labeled incoming transitions while either state on the right has only one. The remaining pairs of states are related by reverse bisimilarity as they have identically labeled incoming transitions, whereas they are told apart by forward-reverse bisimilarity due to the failure of the interplay between outgoing and incoming transitions matching. More precisely, any two corresponding intermediate states are not forward-reverse bisimilar because their identically labeled outgoing transitions reach the aforementioned inequivalent bottom states. In turn, the two initial states are not forward-reverse bisimilar because their identically labeled outgoing transitions reach the aforementioned inequivalent intermediate states. A new level of complexity thus arises from the introduction of parallel composition.

For the sake of completeness, we recall that an interleaving view can be restored by considering computation paths (instead of states) like in the back-and-forth bisimilarity of [26]. Besides causality, this choice additionally preserves history, in the sense that backward moves are constrained to take place along the path followed in the forward direction even in the presence of concurrency. For instance, in the labeled transition system on the left, after performing a and then b it is not possible to undo a before b although there are no causality constraints between those two actions.

In this paper we add parallel composition and then extend the axiomatizations of the three strong bisimilarities examined in [13] via expansion laws. The usual technique consists of introducing normal forms, in which only action prefix and alternative composition occur, along with expansion laws, through which occurrences of parallel composition are progressively eliminated. Although this originated in the interleaving setting for forward-only calculi [32] to *identify* processes such as $a\,.\,\underline{0} \parallel_{\emptyset} b\,.\,\underline{0}$ and $a\,.\,b\,.\,\underline{0} + b\,.\,a\,.\,\underline{0}$, it was later exploited also in the truly concurrent spectrum [31, 28] to *distinguish* processes like the aforementioned two. This requires an extension of the syntax that adds suitable discriminating information within action prefixes. For example:

- Causal bisimilarity [24, 25] (corresponding to history-preserving bisimilarity [51]): every action is enriched with the set of its causing actions, each of which is expressed as a numeric backward pointer, so that the former process is expanded to $<a,\emptyset>\,.\,<b,\emptyset>\,.\,\underline{0} + <b,\emptyset>\,.\,<a,\emptyset>\,.\,\underline{0}$ while the latter process becomes $<a,\emptyset>\,.\,<b,\{1\}>\,.\,\underline{0} + <b,\emptyset>\,.\,<a,\{1\}>\,.\,\underline{0}$.
- Location bisimilarity [18] (corresponding to local history-preserving bisimilarity [20]): every action is enriched with the name of the location in which it is executed, so that the former process is expanded to $<a,l_a>\,.\,<b,l_b>\,.\,\underline{0} + <b,l_b>\,.\,<a,l_a>\,.\,\underline{0}$ while the latter process becomes $<a,l_a>\,.\,<b,l_a l_b>\,.\,\underline{0} + <b,l_b>\,.\,<a,l_b l_a>\,.\,\underline{0}$.
- Pomset bisimilarity [15]: instead of a single action, a prefix may contain the combination of several independent actions that are executed simultaneously, so that the former process is expanded to $a\,.\,b\,.\,\underline{0} + b\,.\,a\,.\,\underline{0} + (a \parallel b)\,.\,\underline{0}$ while the latter process is unchanged.

A unifying framework for addressing both interleaving and truly concurrent semantics along with their expansion laws was developed in [27]. The idea is to label every transition with a proof term [16, 17], which is an action preceded by the operators in the scope of which the action occurs. The semantics of interest then drives an observation function that maps proof terms to the required observations. In the interleaving case proof terms are reduced to the actions they contain, while in the truly concurrent case they are transformed into actions extended with discriminating information as exemplified above.

In this paper we apply the proved trees approach of [27] to develop expansion laws for forward, reverse, and forward-reverse bisimilarities. This requires understanding which additional discriminating information is needed inside prefixes for the last two equivalences. While this is rather straightforward for the truly concurrent semantics recalled above – the considered information is already present in the original transition labels – it is not obvious in our case because original transitions are labeled just with actions. However, by looking at the three bottom states in Figure 1, one can realize that they have different *backward ready sets*, i.e., sets of actions labeling incoming transitions: $\{b,a\}, \{b\}, \{a\}$.

We show that backward ready sets indeed constitute the information that is necessary to add within action prefixes for reverse and forward-reverse bisimilarities, by means of a suitable process encoding. Moreover, we provide an adequate treatment of concurrent processes in which independent actions have been executed on both sides of the parallel composition because, e.g., $a^{\dagger}\,.\,\underline{0} \parallel_{\emptyset} b^{\dagger}\,.\,\underline{0}$ cannot be expanded to something like $a^{\dagger}\,.\,b^{\dagger}\,.\,\underline{0} + b^{\dagger}\,.\,a^{\dagger}\,.\,\underline{0}$ in that only one branch of an alternative composition can be executed.

This paper is organized as follows. In Section 2 we extend the syntax of the reversible process calculus of [13] by adding a parallel composition operator, we reformulate its operational semantics by following the proved trees approach of [27], and we rephrase the definitions of forward, reverse, and forward-reverse bisimilarities of [13]. In Section 3 we illustrate the next steps of the proved trees approach, i.e., the definition of observation functions and process encodings. In Sections 4 and 5 we respectively develop axioms for forward bisimilarity, including an interleaving-style expansion law, and for reverse and forward-reverse bisimilarities, including expansion laws based on extending action prefixes with backward ready sets. In Section 6 we provide some concluding remarks.

2 From Sequential Reversible Processes to Concurrent Ones

Starting from the sequential reversible calculus considered in [13], in this section we extend its syntax with a parallel composition operator in the CSP style [19] (Section 2.1) and its semantics according to the proved trees approach [27] (Section 2.2). Then we rephrase forward, reverse, and forward-reverse bisimilarities and show that they are congruences with respect to the additional operator (Section 2.3).

2.1 Syntax of Concurrent Reversible Processes

Given a countable set A of actions including an unobservable action denoted by τ, the syntax of concurrent reversible processes extends the one in [13] as follows:

$$P ::= \underline{0} \mid a\,.\,P \mid a^{\dagger}\,.\,P \mid P + P \mid P \,\|_L\, P$$

where $a \in A$, † decorates executed actions, $L \subseteq A \setminus \{\tau\}$, and:

- $\underline{0}$ is the terminated process.
- $a\,.\,P$ is a process that can execute action a and whose forward continuation is P.
- $a^{\dagger}\,.\,P$ 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.
- $P_1 + P_2$ 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.
- $P_1 \,\|_L\, P_2$ expresses the parallel composition of P_1 and P_2, which proceed independently of each other on actions in $\bar{L} = A \setminus L$ while they have to synchronize on every action in L.

As in [13] we can characterize some important classes of processes via as many predicates. Firstly, we define *initial* processes, in which all actions are unexecuted and hence no †-decoration appears:

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

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

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

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 arise in the presence of both alternative ($+$) and parallel ($\|_L$) 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 [17, 45], so as to support reversibility. For example, in $a^{\dagger}\,.\,b\,.\,\underline{0} + c\,.\,d\,.\,\underline{0}$ the subprocess $c\,.\,d\,.\,\underline{0}$ cannot move as a was selected in the choice between a and c.

It is worth noting that:

- $\underline{0}$ is both initial and well-formed.

$$(\textsc{Act}_\mathrm{f})\ \frac{\mathit{initial}(P)}{a\,.\,P \xrightarrow{a} a^{\dagger}\,.\,P} \qquad (\textsc{Act}_\mathrm{p})\ \frac{P \xrightarrow{\theta} P'}{a^{\dagger}\,.\,P \xrightarrow{.\theta} a^{\dagger}\,.\,P'}$$

$$(\textsc{Cho}_\mathrm{l})\ \frac{P_1 \xrightarrow{\theta} P_1' \quad \mathit{initial}(P_2)}{P_1 + P_2 \xrightarrow{\ldotp\!\!+\theta} P_1' + P_2} \qquad (\textsc{Cho}_\mathrm{r})\ \frac{P_2 \xrightarrow{\theta} P_2' \quad \mathit{initial}(P_1)}{P_1 + P_2 \xrightarrow{+\!\!\ldotp\theta} P_1 + P_2'}$$

$$(\textsc{Par}_\mathrm{l})\ \frac{P_1 \xrightarrow{\theta} P_1' \quad \mathit{act}(\theta) \notin L}{P_1 \,\|_L\, P_2 \xrightarrow{\ldotp\!\|\theta} P_1' \,\|_L\, P_2} \qquad (\textsc{Par}_\mathrm{r})\ \frac{P_2 \xrightarrow{\theta} P_2' \quad \mathit{act}(\theta) \notin L}{P_1 \,\|_L\, P_2 \xrightarrow{\|\!\ldotp\theta} P_1 \,\|_L\, P_2'}$$

$$(\textsc{Syn})\ \frac{P_1 \xrightarrow{\theta_1} P_1' \quad P_2 \xrightarrow{\theta_2} P_2' \quad \mathit{act}(\theta_1) = \mathit{act}(\theta_2) \in L}{P_1 \,\|_L\, P_2 \xrightarrow{\langle \theta_1, \theta_2 \rangle} P_1' \,\|_L\, P_2'}$$

Table 1: Proved operational semantic rules for concurrent reversible processes

- Any initial process is well-formed too.
- $\mathscr{P}$ also contains processes that are not initial like, e.g., $a^{\dagger}\,.\,b\,.\,\underline{0}$, which can either do b or undo a.
- In $\mathscr{P}$ 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, $a^{\dagger}\,.\,b\,.\,\underline{0} \in \mathscr{P}$ whereas $b\,.\,a^{\dagger}\,.\,\underline{0} \notin \mathscr{P}$.
- In $\mathscr{P}$ the subprocesses of an alternative composition can be both initial, but cannot be both non-initial. As an example, $a\,.\,\underline{0} + b\,.\,\underline{0} \in \mathscr{P}$ whilst $a^{\dagger}\,.\,\underline{0} + b^{\dagger}\,.\,\underline{0} \notin \mathscr{P}$.

2.2 Proved Operational Semantics

According to [45], in the semantic rules dynamic operators such as action prefix and alternative composition have to be made static, so as to retain within the syntax all the information needed to enable reversibility. Unlike [45], we do not generate a forward transition relation and a backward one, but a single transition relation that, like in [26], we deem to be symmetric in order to enforce the *loop property* [22]: every executed action can be undone and every undone action can be redone. In our setting, a backward transition from P' to P is subsumed by the corresponding forward transition t from P to P'. As we will see in the definition of behavioral equivalences, like in [26] we view t as an *outgoing* transition of P when going forward, while we view t as an *incoming* transition of P' when going backward.

Unlike [13], as a first step based on [27] towards the derivation of expansion laws for parallel composition we provide a very concrete semantics in which every transition is labeled with a *proof term* [16, 17]. This is an action preceded by the sequence of operator symbols in the scope of which the action occurs. 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:

$$\theta \ ::= \ a \mid .\theta \mid \ldotp\!\!+\theta \mid +\!\!\ldotp\theta \mid \ldotp\!\|\theta \mid \|\!\ldotp\theta \mid \langle \theta, \theta \rangle$$

The proved semantic rules in Table 1 extend the ones in [13] and generate the proved labeled transition system $(\mathscr{P}, \Theta, \longrightarrow)$ where $\longrightarrow \subseteq \mathscr{P} \times \Theta \times \mathscr{P}$ is the proved transition relation. We denote by $\mathbb{P} \subsetneq \mathscr{P}$ the set of processes that are *reachable* from an initial one via $\longrightarrow$. Not all well-formed processes are reachable; for example, $a^{\dagger}\,.\,\underline{0} \,\|_{\{a\}}\, \underline{0}$ is not reachable from $a\,.\,\underline{0} \,\|_{\{a\}}\, \underline{0}$ as action a on the left cannot synchronize with any action on the right. We indicate with $\mathbb{P}_{\mathrm{init}}$ the set of initial processes in $\mathbb{P}$.

The first rule for action prefix ($\textsc{Act}_\mathrm{f}$ 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 ($\textsc{Act}_\mathrm{p}$ where p stands for propagation) propagates actions of inner initial subprocesses by putting a dot before them in the label for each outer executed action prefix.

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

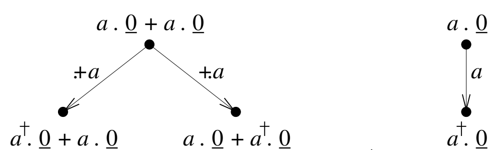
The rules for parallel composition make use of partial function $act : \Theta \rightharpoonup A$ to extract the action from a proof term θ. The function is defined by induction on the syntactical structure of θ as follows:

$$\begin{aligned} act(a) &= a \\ act(.\theta') = act(.\!+\theta') = act(+\!.\,\theta') = act(\rfloor\!\!\|\theta') = act(\|\!\!\lfloor\theta') &= act(\theta') \\ act(\langle\theta_1,\theta_2\rangle) &= act(\theta_1) \quad \text{if } act(\theta_1) = act(\theta_2) \end{aligned}$$

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. In the third rule ($\textsc{Syn}$), both subprocesses synchronize on an action in L.

Every process may have several outgoing transitions and, if it is not initial, has at least one incoming transition. Due to the decoration 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, the underlying labeled transition systems turn out to be trees, and well formedness coincides with reachability [13].

Example 2.1 The proved labeled transition systems generated by the rules in Table 1 for the two initial sequential processes $a\,.\,\underline{0} + a\,.\,\underline{0}$ and $a\,.\,\underline{0}$ are depicted below:



In the case of a forward-only process calculus, a single a-transition would be generated from $a\,.\,\underline{0} + a\,.\,\underline{0}$ to $\underline{0}$ due to the absence of action decorations within processes. ■

2.3 Forward, Reverse, and Forward-Reverse Bisimilarities

We rephrase the definitions given in [13] of forward bisimilarity [43, 41] (only *outgoing* transitions), reverse bisimilarity (only *incoming* transitions), and forward-reverse bisimilarity [45] (both kinds of transitions) because transition labels now are proof terms. Since we focus on the actions contained in those terms, the distinguishing power of the three equivalences does not change with respect to [13].

Definition 2.2 We say that $P_1, P_2 \in \mathbb{P}$ are *forward bisimilar*, written $P_1 \sim_{\mathrm{FB}} P_2$, iff $(P_1, P_2) \in \mathscr{B}$ for some forward bisimulation $\mathscr{B}$. A symmetric relation $\mathscr{B}$ over $\mathbb{P}$ is a *forward bisimulation* iff, whenever $(P_1, P_2) \in \mathscr{B}$, then:

- For each $P_1 \xrightarrow{\theta_1} P'_1$ there exists $P_2 \xrightarrow{\theta_2} P'_2$ such that $act(\theta_1) = act(\theta_2)$ and $(P'_1, P'_2) \in \mathscr{B}$. ■

Definition 2.3 We say that $P_1, P_2 \in \mathbb{P}$ are *reverse bisimilar*, written $P_1 \sim_{\mathrm{RB}} P_2$, iff $(P_1, P_2) \in \mathscr{B}$ for some reverse bisimulation $\mathscr{B}$. A symmetric relation $\mathscr{B}$ over $\mathbb{P}$ is a *reverse bisimulation* iff, whenever $(P_1, P_2) \in \mathscr{B}$, then:

- For each $P'_1 \xrightarrow{\theta_1} P_1$ there exists $P'_2 \xrightarrow{\theta_2} P_2$ such that $act(\theta_1) = act(\theta_2)$ and $(P'_1, P'_2) \in \mathscr{B}$. ■

Definition 2.4 We say that $P_1, P_2 \in \mathbb{P}$ are *forward-reverse bisimilar*, written $P_1 \sim_{\rm FRB} P_2$, iff $(P_1, P_2) \in \mathcal{B}$ for some forward-reverse bisimulation $\mathcal{B}$. A symmetric relation $\mathcal{B}$ over $\mathbb{P}$ is a *forward-reverse bisimulation* iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:

- For each $P_1 \xrightarrow{\theta_1} P_1'$ there exists $P_2 \xrightarrow{\theta_2} P_2'$ such that $act(\theta_1) = act(\theta_2)$ and $(P_1', P_2') \in \mathcal{B}$.
- For each $P_1' \xrightarrow{\theta_1} P_1$ there exists $P_2' \xrightarrow{\theta_2} P_2$ such that $act(\theta_1) = act(\theta_2)$ and $(P_1', P_2') \in \mathcal{B}$. ■

Example 2.5 The two initial processes considered in Example 2.1 are identified by all the three equivalences. This is witnessed by any bisimulation that contains the pairs $(a\,.\,\underline{0} + a\,.\,\underline{0}, a\,.\,\underline{0})$, $(a^\dagger\,.\,\underline{0} + a\,.\,\underline{0}, a^\dagger\,.\,\underline{0})$, and $(a\,.\,\underline{0} + a^\dagger\,.\,\underline{0}, a^\dagger\,.\,\underline{0})$. ■

As observed in [13], $\sim_{\rm FB}$ is not a congruence with respect to alternative composition, e.g.:

$$a^\dagger\,.\,b\,.\,\underline{0} \;\sim_{\rm FB}\; b\,.\,\underline{0} \quad \text{but} \quad a^\dagger\,.\,b\,.\,\underline{0} + c\,.\,\underline{0} \;\not\sim_{\rm FB}\; b\,.\,\underline{0} + c\,.\,\underline{0}$$

because in $a^\dagger\,.\,b\,.\,\underline{0} + c\,.\,\underline{0}$ action c is disabled by virtue of the already executed action $a^\dagger$, while in $b\,.\,\underline{0} + c\,.\,\underline{0}$ action c is enabled as there are no past actions preventing it from occurring. This problem, which does not show up for $\sim_{\rm RB}$ and $\sim_{\rm FRB}$ because they cannot identify an initial process with a non-initial one, led in [13] to the following variant of $\sim_{\rm FB}$ that is sensitive to the presence of the past.

Definition 2.6 We say that $P_1, P_2 \in \mathbb{P}$ are *past-sensitive forward bisimilar*, written $P_1 \sim_{\rm FB:ps} P_2$, iff $(P_1, P_2) \in \mathcal{B}$ for some past-sensitive forward bisimulation $\mathcal{B}$. A relation $\mathcal{B}$ over $\mathbb{P}$ is a *past-sensitive forward bisimulation* iff it is a forward bisimulation where $initial(P_1) \Longleftrightarrow initial(P_2)$ for all $(P_1, P_2) \in \mathcal{B}$. ■

Since $\sim_{\rm FB:ps}$ is sensitive to the presence of the past, we have that $a^\dagger\,.\,b\,.\,\underline{0} \not\sim_{\rm FB:ps} b\,.\,\underline{0}$, but it is still possible to identify non-initial processes having a different past like, e.g., $a_1^\dagger\,.\,P$ and $a_2^\dagger\,.\,P$. It holds that $\sim_{\rm FRB} \subsetneq \sim_{\rm FB:ps} \cap \sim_{\rm RB}$, with $\sim_{\rm FRB} = \sim_{\rm FB:ps}$ over initial processes as well as $\sim_{\rm FB:ps}$ and $\sim_{\rm RB}$ being incomparable because, e.g., for $a_1 \neq a_2$:

$$\begin{array}{ccc} a_1^\dagger\,.\,P \sim_{\rm FB:ps} a_2^\dagger\,.\,P & \text{but} & a_1^\dagger\,.\,P \not\sim_{\rm RB} a_2^\dagger\,.\,P \\ a_1\,.\,P \sim_{\rm RB} a_2\,.\,P & \text{but} & a_1\,.\,P \not\sim_{\rm FB:ps} a_2\,.\,P \end{array}$$

It is easy to establish two necessary conditions for the considered bisimilarities. Following the terminology of [42, 7], the two conditions respectively make use of the forward ready set in the forward direction and the backward ready set in the backward direction; the latter condition will be exploited when developing the expansion laws for $\sim_{\rm RB}$ and $\sim_{\rm FRB}$. We proceed by induction on the syntactical structure of $P \in \mathbb{P}$ to define its *forward ready set* $frs(P) \subseteq A$, i.e., the set of actions that P can immediately execute (labels of its outgoing transitions), as well as its *backward ready set* $brs(P) \subseteq A$, i.e., the set of actions whose execution led to P (labels of its incoming transitions):

$$\begin{array}{rclrcl}
frs(\underline{0}) & = & \emptyset & brs(\underline{0}) & = & \emptyset \\
frs(a\,.\,P') & = & \{a\} & brs(a\,.\,P') & = & \emptyset \\
frs(a^\dagger\,.\,P') & = & frs(P') & brs(a^\dagger\,.\,P') & = & \left\{ \begin{array}{ll} \{a\} & \text{if } initial(P') \\ brs(P') & \text{if } \neg initial(P') \end{array} \right.
\end{array}$$

$$frs(P_1 + P_2) = \left\{ \begin{array}{ll} frs(P_1) \cup frs(P_2) & \text{if } initial(P_1) \wedge initial(P_2) \\ frs(P_1) & \text{if } \neg initial(P_1) \wedge initial(P_2) \\ frs(P_2) & \text{if } initial(P_1) \wedge \neg initial(P_2) \end{array} \right.$$

$$brs(P_1 + P_2) = \left\{ \begin{array}{ll} \emptyset & \text{if } initial(P_1) \wedge initial(P_2) \\ brs(P_1) & \text{if } \neg initial(P_1) \wedge initial(P_2) \\ brs(P_2) & \text{if } initial(P_1) \wedge \neg initial(P_2) \end{array} \right.$$

$$\begin{array}{rcl}
frs(P_1 \,\|_L\, P_2) & = & (frs(P_1) \cap \bar{L}) \cup (frs(P_2) \cap \bar{L}) \cup (frs(P_1) \cap frs(P_2) \cap L) \\
brs(P_1 \,\|_L\, P_2) & = & (brs(P_1) \cap \bar{L}) \cup (brs(P_2) \cap \bar{L}) \cup (brs(P_1) \cap brs(P_2) \cap L)
\end{array}$$

Proposition 2.7 Let $P_1, P_2 \in \mathbb{P}$. Then:

1. If $P_1 \sim P_2$ for $\sim \in \{\sim_{\rm FB}, \sim_{\rm FB:ps}, \sim_{\rm FRB}\}$, then $\mathit{frs}(P_1) = \mathit{frs}(P_2)$.
2. If $P_1 \sim P_2$ for $\sim \in \{\sim_{\rm RB}, \sim_{\rm FRB}\}$, then $\mathit{brs}(P_1) = \mathit{brs}(P_2)$. ■

In [13] it has been shown that all these four bisimilarities are congruences with respect to action prefix, while only $\sim_{\rm FB:ps}$, $\sim_{\rm RB}$, and $\sim_{\rm FRB}$ are congruences with respect to alternative composition too, with $\sim_{\rm FB:ps}$ being the coarsest congruence with respect to $+$ contained in $\sim_{\rm FB}$. Sound and ground-complete equational characterizations have also been provided for the three congruences. Here we show that all these bisimilarities are congruences with respect to the newly added operator, i.e., parallel composition.

Theorem 2.8 Let $\sim \in \{\sim_{\rm FB}, \sim_{\rm FB:ps}, \sim_{\rm RB}, \sim_{\rm FRB}\}$ and $P_1, P_2 \in \mathbb{P}$. If $P_1 \sim P_2$ then $P_1 \,\|_L\, P \sim P_2 \,\|_L\, P$ and $P \,\|_L\, P_1 \sim P \,\|_L\, P_2$ for all $P \in \mathbb{P}$ and $L \subseteq A \setminus \{\tau\}$ such that $P_1 \,\|_L\, P, P_2 \,\|_L\, P, P \,\|_L\, P_1, P \,\|_L\, P_2 \in \mathbb{P}$. ■

3 Observation Functions and Process Encodings for Expansion Laws

Among the most important axioms there are *expansion laws*, which are useful to relate sequential specifications of systems with their concurrent implementations [41]. In the interleaving setting they can be obtained quite naturally, whereas this is not the case under true concurrency. Thanks to the proved operational semantics in Table 1, we can uniformly derive expansion laws for the interleaving bisimulation congruence $\sim_{\rm FB:ps}$ and the two truly concurrent bisimulation congruences $\sim_{\rm RB}$ and $\sim_{\rm FRB}$ by following the proved trees approach of [27].

All we have to do is the introduction of three *observation functions* $\ell_{\rm F}$, $\ell_{\rm R}$, and $\ell_{\rm FR}$ that respectively transform the proof terms labeling all proved transitions into suitable observations according to $\sim_{\rm FB:ps}$, $\sim_{\rm RB}$, and $\sim_{\rm FRB}$. In addition to a specific proof term θ, as shown in [27] each such function, say ℓ, may depend on other possible parameters in the proved labeled transition system generated by the semantic rules in Table 1. Moreover, it must preserve actions, i.e., if $\ell(\theta_1) = \ell(\theta_2)$ then $\mathit{act}(\theta_1) = \mathit{act}(\theta_2)$.

Based on the corresponding ℓ, from each of the three aforementioned congruences we can thus derive a bisimilarity in which, whenever $(P_1, P_2) \in \mathcal{B}$, the forward clause requires that:

$$\text{for each } P_1 \xrightarrow{\ell(\theta_1)} P_1' \text{ there exists } P_2 \xrightarrow{\ell(\theta_2)} P_2' \text{ such that } \ell(\theta_1) = \ell(\theta_2) \text{ and } (P_1', P_2') \in \mathcal{B}$$

while the backward clause requires that:

$$\text{for each } P_1' \xrightarrow{\ell(\theta_1)} P_1 \text{ there exists } P_2' \xrightarrow{\ell(\theta_2)} P_2 \text{ such that } \ell(\theta_1) = \ell(\theta_2) \text{ and } (P_1', P_2') \in \mathcal{B}$$

We indicate with $\sim_{{\rm FB:ps:}\ell_{\rm F}}$, $\sim_{{\rm RB:}\ell_{\rm R}}$, and $\sim_{{\rm FRB:}\ell_{\rm FR}}$ the three resulting bisimilarities.

To derive the corresponding expansion laws, the final step – left implicit in [27], see, e.g., the forthcoming Definitions 5.1 and 5.3 – consists of lifting ℓ to processes so as to encode observations within action prefixes. For a process $P \in \mathbb{P}_{\rm seq}$, the idea is to proceed by induction on the syntactical structure of P as follows, where $\sigma \in \Theta^*_{\rm seq}$ for $\Theta_{\rm seq} = \{., .{+}, {+}.\}$:

$$\begin{array}{rcl} \ell^{\sigma}(\underline{0}) & = & \underline{0} \\ \ell^{\sigma}(a\,.\,P') & = & \ell(\sigma a)\,.\,\ell^{\sigma .}(P') \\ \ell^{\sigma}(a^{\dagger}.\,P') & = & \ell(\sigma a)^{\dagger}.\,\ell^{\sigma .}(P') \\ \ell^{\sigma}(P_1 + P_2) & = & \ell^{.{+}\sigma}(P_1) + \ell^{{+}.\sigma}(P_2) \end{array}$$

Every sequential process P will thus be encoded as $\ell^{\varepsilon}(P)$, so for example $a\,.\,b\,.\,\underline{0} + b\,.\,a\,.\,\underline{0}$ will become:
$\ell^{.{+}}(a\,.\,b\,.\,\underline{0}) + \ell^{{+}.}(b\,.\,a\,.\,\underline{0}) = \ell(.{+}a)\,.\,\ell^{.{+}.}(b\,.\,\underline{0}) + \ell({+}.b)\,.\,\ell^{{+}..}(a\,.\,\underline{0}) = \ell(.{+}a)\,.\,\ell(.{+}.b)\,.\,\underline{0} + \ell({+}.b)\,.\,\ell({+}..a)\,.\,\underline{0}$

Then, given two initial sequential processes expressed as follows due to the commutativity and associativity of alternative composition (where any summation over an empty index set is $\underline{0}$):

$$P_1 \,=\, \sum_{i \in I_1} \ell(\theta_{1,i}) \,.\, P_{1,i} \quad \text{and} \quad P_2 \,=\, \sum_{i \in I_2} \ell(\theta_{2,i}) \,.\, P_{2,i}$$

the idea is to encode their parallel composition via the following expansion law (where $\underline{0} \,\|_L\, \underline{0}$ yields $\underline{0}$):

$$\begin{aligned} P_1 \,\|_L\, P_2 \;=\; & \sum_{i \in I_1, act(\theta_{1,i}) \notin L} \ell(\lfloor\!\lfloor \theta_{1,i}) \,.\, (P_{1,i} \,\|_L\, P_2) + \sum_{i \in I_2, act(\theta_{2,i}) \notin L} \ell(\rfloor\!\rfloor \theta_{2,i}) \,.\, (P_1 \,\|_L\, P_{2,i}) + \\ & \sum_{i \in I_1, act(\theta_{1,i}) \in L} \; \sum_{j \in I_2, act(\theta_{2,j}) = act(\theta_{1,i})} \ell(\langle \theta_{1,i}, \theta_{2,j} \rangle) \,.\, (P_{1,i} \,\|_L\, P_{2,j}) \end{aligned}$$

For instance, $a \,.\, \underline{0} \,\|_\emptyset\, b \,.\, \underline{0}$, represented as $\ell(a) \,.\, \underline{0} \,\|_\emptyset\, \ell(b) \,.\, \underline{0}$, will be expanded as follows:

$$\ell(\lfloor\!\lfloor_\emptyset a) \,.\, (\underline{0} \,\|_\emptyset\, \ell(b) \,.\, \underline{0}) + \ell(\rfloor\!\rfloor_\emptyset b) \,.\, (\ell(a) \,.\, \underline{0} \,\|_\emptyset\, \underline{0}) \;=\; \ell(\lfloor\!\lfloor_\emptyset a) \,.\, \ell(\rfloor\!\rfloor_\emptyset b) \,.\, \underline{0} + \ell(\rfloor\!\rfloor_\emptyset b) \,.\, \ell(\lfloor\!\lfloor_\emptyset a) \,.\, \underline{0}$$

where, compared to the encoding of $a \,.\, b \,.\, \underline{0} + b \,.\, a \,.\, \underline{0}$, in general $\ell(\dot{+} a) \neq \ell(\lfloor\!\lfloor_\emptyset a) \neq \ell(+.a)$ and $\ell(\dot{+}.b) \neq \ell(\rfloor\!\rfloor_\emptyset b) \neq \ell(+b)$. The expansion laws for the cases in which the two sequential processes are not both initial – which are specific to reversible processes and hence not addressed in [27] – are derived similarly. We will see that care must be taken when both processes are non-initial because for example $a^\dagger \,.\, \underline{0} \,\|_\emptyset\, b^\dagger \,.\, \underline{0}$ cannot be expanded to $\ell(\lfloor\!\lfloor a)^\dagger \,.\, \ell(\rfloor\!\rfloor b)^\dagger \,.\, \underline{0} + \ell(\rfloor\!\rfloor b)^\dagger \,.\, \ell(\lfloor\!\lfloor a)^\dagger \,.\, \underline{0}$ as the latter is not even well-formed due to the presence of executed actions on both sides of the alternative composition.

In the next two sections we will investigate how to define the three observation functions $\ell_{\rm F}$, $\ell_{\rm R}$, and $\ell_{\rm FR}$ in such a way that the three equivalences $\sim_{\rm FB:ps:\ell_F}$, $\sim_{\rm RB:\ell_R}$, and $\sim_{\rm FRB:\ell_{FR}}$ respectively coincide with the three congruences $\sim_{\rm FB:ps}$, $\sim_{\rm RB}$, and $\sim_{\rm FRB}$. As far as true concurrency is concerned, we point out that the observation functions developed in [27] for causal semantics and location semantics were inspired by additional information already present in the labels of the original semantics, i.e., backward pointers sets [24] and localities [18] respectively. In our case, instead, the original semantics in Table 1 features labels that are essentially actions, hence for reverse and forward-reverse bisimilarities we have to find out the additional information necessary to discriminate, e.g., the processes associated with the three bottom states in Figure 1.

4 Axioms and Expansion Law for $\sim_{\rm FB:ps}$

In this section we provide a sound and ground-complete axiomatization of forward bisimilarity over concurrent reversible processes. As already mentioned, this behavioral equivalence complies with the interleaving view of concurrency. Therefore, we can exploit the same observation function for interleaving semantics used in [27], which we express as $\ell_{\rm F}(\theta) = act(\theta)$ and immediately implies that $\sim_{\rm FB:ps:\ell_F}$ coincides with $\sim_{\rm FB:ps}$. Moreover, no additional information has to be inserted into action prefixes, i.e., the lifting to processes of the observation function is accomplished via the identity function.

The set $\mathscr{A}_{\rm F}$ of axioms for $\sim_{\rm FB:ps}$ is shown in Table 2 (where-clauses are related to $\mathbb{P}$-membership). All the axioms apart from the last one come from [13], where an axiomatization was developed over sequential reversible processes. Axioms $\mathscr{A}_{\rm F,1}$ to $\mathscr{A}_{\rm F,4}$ – associativity, commutativity, neutral element, and idempotency of alternative composition – coincide with those for forward-only processes [32]. Axioms $\mathscr{A}_{\rm F,5}$ and $\mathscr{A}_{\rm F,6}$ together establish that the presence of the past cannot be ignored, but the specific past can be neglected when moving only forward. Likewise, axiom $\mathscr{A}_{\rm F,7}$ states that a previously non-selected alternative process can be discarded when moving only forward; note that it does not subsume axioms $\mathscr{A}_{\rm F,3}$ and $\mathscr{A}_{\rm F,4}$ because P has to be non-initial.

Since due to axioms $\mathscr{A}_{\rm F,5}$ and $\mathscr{A}_{\rm F,6}$ we only need to remember whether some action has been executed in the past, axiom $\mathscr{A}_{\rm F,8}$ is the only expansion law needed for $\sim_{\rm FB:ps}$. Notation $[a^\dagger.]$ stands for the possible presence of an executed action prefix, with $a^\dagger$ being present at the beginning of the expansion iff at least one of $a_1^\dagger$ and $a_2^\dagger$ is present at the beginning of P_1 and P_2 respectively. In P_1 and P_2, as well as on the righthand side of the expansion, summations are allowed by axioms $\mathscr{A}_{\rm F,1}$ and $\mathscr{A}_{\rm F,2}$ and represent $\underline{0}$ when

$(\mathscr{A}_{\mathrm{F},1})$	$(P+Q)+R \;=\; P+(Q+R)$	where at least two among P, Q, R are initial
$(\mathscr{A}_{\mathrm{F},2})$	$P+Q \;=\; Q+P$	where at least one between P and Q is initial
$(\mathscr{A}_{\mathrm{F},3})$	$P+\underline{0} \;=\; P$	
$(\mathscr{A}_{\mathrm{F},4})$	$P+P \;=\; P$	where $\mathit{initial}(P)$
$(\mathscr{A}_{\mathrm{F},5})$	$a^{\dagger}\,.\,P \;=\; b^{\dagger}\,.\,P$	if $\mathit{initial}(P)$
$(\mathscr{A}_{\mathrm{F},6})$	$a^{\dagger}.\,P \;=\; P$	if $\neg\mathit{initial}(P)$
$(\mathscr{A}_{\mathrm{F},7})$	$P+Q \;=\; P$	if $\neg\mathit{initial}(P)$, where $\mathit{initial}(Q)$
$(\mathscr{A}_{\mathrm{F},8})$	$P_1 \,\Vert_L\, P_2 \;=\; [a^{\dagger}.]\Big(\sum_{i \in I_1, a_{1,i} \notin L} a_{1,i}\,.\,(P_{1,i} \,\Vert_L\, P'_2) + \sum_{i \in I_2, a_{2,i} \notin L} a_{2,i}\,.\,(P'_1 \,\Vert_L\, P_{2,i}) + \sum_{i \in I_1, a_{1,i} \in L} \sum_{j \in I_2, a_{2,j} = a_{1,i}} a_{1,i}\,.\,(P_{1,i} \,\Vert_L\, P_{2,j})\Big)$	
	with $P_k = [a_k^{\dagger}.]P'_k$, $P'_k = \sum_{i \in I_k} a_{k,i}\,.\,P_{k,i}$ in F-nf for $k \in \{1,2\}$ and $a^{\dagger}$ present iff so is $a_1^{\dagger}$ or $a_2^{\dagger}$	

Table 2: Axioms characterizing $\sim_{\mathrm{FB:ps}}$ over concurrent reversible processes

their index sets are empty (so that $\mathscr{A}_{\mathrm{F}} \vdash \underline{0} \,\Vert_L\, \underline{0} = \underline{0} + \underline{0} + \underline{0} = \underline{0}$ due to axiom $\mathscr{A}_{\mathrm{F},3}$, substitutivity with respect to alternative composition, and transitivity).

The deduction system based on $\mathscr{A}_{\mathrm{F}}$, whose deducibility relation we denote by $\vdash$, includes axioms and inference rules expressing reflexivity, symmetry, and transitivity (because $\sim_{\mathrm{FB:ps}}$ is an equivalence relation) as well as substitutivity with respect to the operators of the considered calculus (because $\sim_{\mathrm{FB:ps}}$ is a congruence with respect to all of those operators). Following [32], to show the soundness and ground-completeness of $\mathscr{A}_{\mathrm{F}}$ for $\sim_{\mathrm{FB:ps}}$ we introduce a suitable normal form to which every process can be reduced. The only operators that can occur in such a normal form are action prefix and alternative composition, hence all processes in normal form are sequential.

Definition 4.1 We say that $P \in \mathbb{P}$ is in *forward normal form*, written *F-nf*, iff it is equal to $[b^{\dagger}.]\sum_{i \in I} a_i\,.\,P_i$ where the executed action prefix $b^{\dagger}.\,\_$ is optional, I is a finite index set (with the summation being $\underline{0}$ when $I = \emptyset$), and each P_i is initial and in F-nf. ■

Lemma 4.2 For all (initial) $P \in \mathbb{P}$ there exists (an initial) $Q \in \mathbb{P}$ in F-nf such that $\mathscr{A}_{\mathrm{F}} \vdash P = Q$. ■

Theorem 4.3 Let $P_1, P_2 \in \mathbb{P}$. Then $P_1 \sim_{\mathrm{FB:ps}} P_2$ iff $\mathscr{A}_{\mathrm{F}} \vdash P_1 = P_2$. ■

5 Axioms and Expansion Laws for $\sim_{\mathrm{RB}}$ and $\sim_{\mathrm{FRB}}$

In this section we address the axiomatization of reverse and forward-reverse bisimilarities over concurrent reversible processes. Since these behavioral equivalences are truly concurrent, we have to provide process encodings that insert suitable additional discriminating information into action prefixes. We show that this information is the same for both semantics and is constituted by backward ready sets. Precisely, for every proved transition $P \xrightarrow{\theta} P'$, we let $\ell_{\mathrm{R}}(\theta)_{P'} = \ell_{\mathrm{FR}}(\theta)_{P'} = <\mathit{act}(\theta), \mathit{brs}(P')> \triangleq \ell_{\mathrm{brs}}(\theta)_{P'}$, where in the observation function we have indicated its primary argument θ in parentheses and its secondary argument P' as a subscript (see Section 3 for the possibility of additional parameters like P').

$$(\textsc{Act}_{\mathrm{brs,f}})\ \frac{\mathit{initial}(U)}{<a,\sqsupset>.\,U \xrightarrow{a,\sqsupset}_{\mathrm{brs}} <a^{\dagger},\sqsupset>.\,U} \qquad (\textsc{Act}_{\mathrm{brs,p}})\ \frac{U \xrightarrow{\theta,\urcorner}_{\mathrm{brs}} U'}{<a^{\dagger},\sqsupset>.\,U \xrightarrow{.\theta,\urcorner}_{\mathrm{brs}} <a^{\dagger},\sqsupset>.\,U'}$$

$$(\textsc{Cho}_{\mathrm{brs,l}})\ \frac{U_1 \xrightarrow{\theta,\sqsupset}_{\mathrm{brs}} U_1' \quad \mathit{initial}(U_2)}{U_1 + U_2 \xrightarrow{+\theta,\sqsupset}_{\mathrm{brs}} U_1' + U_2} \qquad (\textsc{Cho}_{\mathrm{brs,r}})\ \frac{U_2 \xrightarrow{\theta,\sqsupset}_{\mathrm{brs}} U_2' \quad \mathit{initial}(U_1)}{U_1 + U_2 \xrightarrow{+\theta,\sqsupset}_{\mathrm{brs}} U_1 + U_2'}$$

Table 3: Proved operational semantic rules for $\mathbb{P}_{\mathrm{brs}}$ ($\sqsupset, \urcorner \in 2^A$)

By virtue of Proposition 2.7(2), the distinguishing power of $\sim_{\mathrm{RB}}$ and $\sim_{\mathrm{FRB}}$ does not change if, in the related definitions of bisimulation, we additionally require that $\mathit{brs}(P_1) = \mathit{brs}(P_2)$ for all $(P_1, P_2) \in \mathscr{B}$. As a consequence, it is straightforward to realize that $\sim_{\mathrm{RB}:\ell_{\mathrm{brs}}}$ and $\sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}}$ (see page 8) respectively coincide with $\sim_{\mathrm{RB}}$ and $\sim_{\mathrm{FRB}}$ over $\mathbb{P}$. Moreover, $\sim_{\mathrm{RB}:\ell_{\mathrm{brs}}}$ and $\sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}}$ also apply to the encoding target $\mathbb{P}_{\mathrm{brs}}$, i.e., the set of processes obtained from $\mathbb{P}_{\mathrm{seq}}$ by extending every action prefix with a subset of A.

The syntax of $\mathbb{P}_{\mathrm{brs}}$ processes is defined as follows where $\sqsupset \in 2^A$:

$$U \ ::= \ \underline{0} \mid <a,\sqsupset>.\,U \mid <a^{\dagger},\sqsupset>.\,U \mid U + U$$

The proved operational semantic rules for $\mathbb{P}_{\mathrm{brs}}$ shown in Table 3 generate the proved labeled transition system $(\mathbb{P}_{\mathrm{brs}}, \Theta \times 2^A, \longrightarrow_{\mathrm{brs}})$. With respect to those in Table 1, the rules in Table 3 additionally label the produced transitions with the action sets occurring in the action prefixes inside the source processes. Given a symmetric relation $\mathscr{B}$ over $\mathbb{P}_{\mathrm{brs}}$, whenever $(U_1, U_2) \in \mathscr{B}$ the forward clause of $\sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}}$ can be rephrased as:

for each $U_1 \xrightarrow{\theta_1,\sqsupset}_{\mathrm{brs}} U_1'$ there exists $U_2 \xrightarrow{\theta_2,\sqsupset}_{\mathrm{brs}} U_2'$ such that $\mathit{act}(\theta_1) = \mathit{act}(\theta_2)$ and $(U_1', U_2') \in \mathscr{B}$

while the backward clauses of $\sim_{\mathrm{RB}:\ell_{\mathrm{brs}}}$ and $\sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}}$ can be rephrased as:

for each $U_1' \xrightarrow{\theta_1,\sqsupset}_{\mathrm{brs}} U_1$ there exists $U_2' \xrightarrow{\theta_2,\sqsupset}_{\mathrm{brs}} U_2$ such that $\mathit{act}(\theta_1) = \mathit{act}(\theta_2)$ and $(U_1', U_2') \in \mathscr{B}$

Following the proved trees approach as described in Section 3, we now lift ℓ_{brs} so as to encode $\mathbb{P}$ into $\mathbb{P}_{\mathrm{brs}}$. The objective is to extend each action prefix with the backward ready set of the reached process. While in the case of processes in $\mathbb{P}_{\mathrm{seq}}$ it is just a matter of extending any action prefix with a singleton containing the action itself, backward ready sets may contain several actions when handling processes not in $\mathbb{P}_{\mathrm{seq}}$. To account for this, the lifting of ℓ_{brs} has to make use of a secondary argument, which we call environment process and will be written as a subscript by analogy with the secondary argument of the observation function.

The environment process is progressively updated as prefixes are turned into pairs so as to represent the process reached so far, i.e., the process yielding the backward ready set. The environment process E for P embodies P, in the sense that it is initially P and then its forward behavior is updated upon each action prefix extension by decorating the action as executed, where the action is located within E by a proof term. To correctly handle the extension of already executed prefixes, (part of) E has to be brought back by replacing P inside E with the process $\mathit{to\_initial}(P)$ obtained from P by removing all †-decorations. Function $\mathit{to\_initial} : \mathbb{P} \to \mathbb{P}_{\mathrm{init}}$ is defined by induction on the syntactical structure of $P \in \mathbb{P}$ as follows:

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

In Definitions 5.1 and 5.3 we develop the lifting of ℓ_{brs} and denote by $\widetilde{P}$ the result of its application.

We recall that $\ell_{\text{brs}}(\theta)_{P'} = <act(\theta), brs(P')>$ and we let $\ell_{\text{brs}}(\theta)^{\dagger}_{P'} = <act(\theta)^{\dagger}, brs(P')>$. We further recall that $\Theta_{\text{seq}} = \{., .\!+, +\!.\}$.

Definition 5.1 Let $P \in \mathbb{P}$, $E \in \mathbb{P}$ be such that P is a subprocess of E, and $\ddot{E}$ be obtained from E by replacing P with *to_initial*(P). The ℓ_{brs}-*encoding* of P is $\widetilde{P} = \ell^{\varepsilon}_{\text{brs}}(P)_P$, where $\ell^{\sigma}_{\text{brs}} : \mathbb{P} \times \mathbb{P} \to \mathbb{P}_{\text{brs}}$ for $\sigma \in \Theta^*_{\text{seq}}$ is defined by induction on the syntactical structure of its primary argument $P \in \mathbb{P}$ (its secondary argument is $E \in \mathbb{P}$) as follows:

$$\begin{array}{rcl}
\ell^{\sigma}_{\text{brs}}(\underline{0})_E & = & \underline{0} \\
\ell^{\sigma}_{\text{brs}}(a\,.\,P')_E & = & \ell_{\text{brs}}(\sigma a)_{upd(E,\sigma a)} \,.\, \ell^{\sigma .}_{\text{brs}}(P')_{upd(E,\sigma a)} \\
\ell^{\sigma}_{\text{brs}}(a^{\dagger}.\,P')_E & = & \ell_{\text{brs}}(\sigma a)^{\dagger}_{upd(\ddot{E},\sigma a)} \,.\, \ell^{\sigma .}_{\text{brs}}(P')_E \\
\ell^{\sigma}_{\text{brs}}(P_1 + P_2)_E & = & \ell^{\sigma .\!+}_{\text{brs}}(P_1)_E + \ell^{\sigma +\!.}_{\text{brs}}(P_2)_E \\
\ell^{\sigma}_{\text{brs}}(P_1 \,\|_L\, P_2)_E & = & e\ell^{\sigma}_{\text{brs}}(\widetilde{P}_1, \widetilde{P}_2, L)_E
\end{array}$$

where function $e\ell^{\sigma}_{\text{brs}}$ will be defined later on while function $upd : \mathbb{P} \times \Theta \to \mathbb{P}$ is defined by induction on the syntactical structural of its first argument $E \in \mathbb{P}$ as follows:

$$\begin{array}{rcl}
upd(\underline{0}, \theta) & = & \underline{0} \\
upd(a\,.\,E', \theta) & = & \begin{cases} a^{\dagger}.\,E' & \text{if } \theta = a \\ a\,.\,E' & \text{otherwise} \end{cases} \\
upd(a^{\dagger}.\,E', \theta) & = & \begin{cases} a^{\dagger}.\,upd(E', \theta') & \text{if } \theta = .\theta' \\ a^{\dagger}.\,E' & \text{otherwise} \end{cases} \\
upd(E_1 + E_2, \theta) & = & \begin{cases} upd(E_1, \theta') + E_2 & \text{if } \theta = .\!+\theta' \\ E_1 + upd(E_2, \theta') & \text{if } \theta = +\!.\theta' \\ E_1 + E_2 & \text{otherwise} \end{cases} \\
upd(E_1 \,\|_L\, E_2, \theta) & = & \begin{cases} upd(E_1, \theta') \,\|_L\, E_2 & \text{if } \theta = \lfloor\!\mid\theta' \\ E_1 \,\|_L\, upd(E_2, \theta') & \text{if } \theta = \mid\!\rfloor\theta' \\ upd(E_1, \theta_1) \,\|_L\, upd(E_2, \theta_2) & \text{if } \theta = \langle \theta_1, \theta_2 \rangle \\ E_1 \,\|_L\, E_2 & \text{otherwise} \end{cases}
\end{array}$$

■

Example 5.2 Encoding sequential processes (for them function $e\ell^{\sigma}_{\text{brs}}$ does not come into play):

- Let P be the initial sequential process $a\,.\,b\,.\,\underline{0} + b\,.\,a\,.\,\underline{0}$. Then:

$$\begin{array}{rcl}
\widetilde{P} = \ell^{\varepsilon}_{\text{brs}}(P)_P & = & \ell^{.\!+}_{\text{brs}}(a\,.\,b\,.\,\underline{0})_{a.b.\underline{0}+b.a.\underline{0}} + \ell^{+\!.}_{\text{brs}}(b\,.\,a\,.\,\underline{0})_{a.b.\underline{0}+b.a.\underline{0}} \\
& = & \ell_{\text{brs}}(.\!+a)_{a^{\dagger}.b.\underline{0}+b.a.\underline{0}} \,.\, \ell^{.\!+.}_{\text{brs}}(b\,.\,\underline{0})_{a^{\dagger}.b.\underline{0}+b.a.\underline{0}} + \\
& & \ell_{\text{brs}}(+\!.b)_{a.b.\underline{0}+b^{\dagger}.a.\underline{0}} \,.\, \ell^{+\!..}_{\text{brs}}(a\,.\,\underline{0})_{a.b.\underline{0}+b^{\dagger}.a.\underline{0}} \\
& = & <a,\{a\}>.\,\ell_{\text{brs}}(.\!+.b)_{a^{\dagger}.b^{\dagger}.\underline{0}+b.a.\underline{0}} \,.\, \ell^{.\!+..}_{\text{brs}}(\underline{0})_{a^{\dagger}.b^{\dagger}.\underline{0}+b.a.\underline{0}} + \\
& & <b,\{b\}>.\,\ell_{\text{brs}}(+\!..a)_{a.b.\underline{0}+b^{\dagger}.a^{\dagger}.\underline{0}} \,.\, \ell^{+\!...}_{\text{brs}}(\underline{0})_{a.b.\underline{0}+b^{\dagger}.a^{\dagger}.\underline{0}} \\
& = & <a,\{a\}>.<b,\{b\}>.\,\underline{0} + <b,\{b\}>.<a,\{a\}>.\,\underline{0}
\end{array}$$

- Let P be the non-initial sequential process $a^{\dagger}.\,b^{\dagger}.\,\underline{0}$. Then:

$$\begin{array}{rcl}
\widetilde{P} = \ell^{\varepsilon}_{\text{brs}}(P)_P & = & \ell_{\text{brs}}(a)^{\dagger}_{a^{\dagger}.b.\underline{0}} \,.\, \ell^{.}_{\text{brs}}(b^{\dagger}.\,\underline{0})_{a^{\dagger}.b^{\dagger}.\underline{0}} \;= \\
& = & <a^{\dagger},\{a\}>.\,\ell_{\text{brs}}(.b)^{\dagger}_{a^{\dagger}.b^{\dagger}.\underline{0}} \,.\, \ell^{..}_{\text{brs}}(\underline{0})_{a^{\dagger}.b^{\dagger}.\underline{0}} \;=\; <a^{\dagger},\{a\}>.<b^{\dagger},\{b\}>.\,\underline{0}
\end{array}$$

  Definition 5.1 yields $a\,.\,b\,.\,\underline{0}$ as $\ddot{P}$ after the second $=$ (before it, P is a subprocess of the environment P) and $a^{\dagger}.\,b\,.\,\underline{0}$ as $\ddot{P}$ after the third $=$ (before it, $b^{\dagger}.\,\underline{0}$ is a subprocess of the environment P). ■

While for sequential processes the backward ready set added to every action prefix is a singleton containing the action itself, this is no longer the case when dealing with processes of the form $P_1 \,\|_L\, P_2$. We thus complete the encoding by providing the definition of $e\ell^{\sigma}_{\text{brs}}$. When P_1 and P_2 are not both initial, in the expansion we have to reconstruct all possible alternative action sequencings that have not been undertaken – which yield as many initial subprocesses – because under the forward-reverse semantics each of them could be selected after a rollback. In the subcase in which both P_1 and P_2 are non-initial and their executed actions are not in L – e.g., $a^\dagger . \underline{0} \parallel_\emptyset b^\dagger . \underline{0}$ – care must be taken because executed actions cannot appear on both sides of an alternative composition – e.g., the expansion cannot be $a^\dagger . b^\dagger . \underline{0} + b^\dagger . a^\dagger . \underline{0}$ in that not well-formed. To overcome this, based on a total order $\leq_\dagger$ over Θ induced by the trace of actions executed so far, the expansion builds the corresponding sequencing of already executed actions plus all the aforementioned unexecuted action sequencings – e.g., $a^\dagger . b^\dagger . \underline{0} + b . a . \underline{0}$ or $b^\dagger . a^\dagger . \underline{0} + a . b . \underline{0}$ depending on whether $\rfloor\!\!\rfloor a \leq_\dagger \lfloor\!\!\lfloor b$ or $\lfloor\!\!\lfloor b \leq_\dagger \rfloor\!\!\rfloor a$ respectively.

Definition 5.3 Let $P_1, P_2 \in \mathbb{P}$, $L \subseteq A \setminus \{\tau\}$, $E_1, E_2, E \in \mathbb{P}$ be such that $P_1 \parallel_L P_2, E_1 \parallel_L E_2 \in \mathbb{P}$, P_1 is a subprocess of E_1, P_2 is a subprocess of E_2, and $E_1 \parallel_L E_2$ is a subprocess of E. Then $e\ell^\sigma_{\rm brs} : \mathbb{P}_{\rm brs} \times \mathbb{P}_{\rm brs} \times 2^{A \setminus \{\tau\}} \times \mathbb{P} \rightarrow \mathbb{P}_{\rm brs}$ for $\sigma \in \Theta^*_{\rm seq}$ is inductively defined as follows, where square brackets enclose optional subprocesses as already done in Section 4 and every summation over an empty index set is taken to be $\underline{0}$ (and for simplicity is omitted within a choice unless all alternative subprocesses inside that choice are $\underline{0}$, in which case the whole choice boils down to $\underline{0}$):

- If $\widetilde{P}_1$ and $\widetilde{P}_2$ are both initial, say $\widetilde{P}_k = \sum_{i \in I_k} \ell_{\rm brs}(\theta_{k,i})_{upd(P_k, \theta_{k,i})} . \widetilde{P}_{k,i}$ for $k \in \{1,2\}$, let $e\ell^\sigma_{\rm brs}(\widetilde{P}_1, \widetilde{P}_2, L)_E$
$$\begin{aligned} = & \sum_{i \in I_1, act(\theta_{1,i}) \notin L} \ell_{\rm brs}(\sigma \rfloor\!\!\rfloor \theta_{1,i})_{upd(E, \sigma \rfloor\!\rfloor \theta_{1,i})} . e\ell^\sigma_{\rm brs}(\widetilde{P}_{1,i}, \widetilde{P}_2, L)_{upd(E, \sigma \rfloor\!\rfloor \theta_{1,i})} + \\ & \sum_{i \in I_2, act(\theta_{2,i}) \notin L} \ell_{\rm brs}(\sigma \lfloor\!\!\lfloor \theta_{2,i})_{upd(E, \sigma \lfloor\!\lfloor \theta_{2,i})} . e\ell^\sigma_{\rm brs}(\widetilde{P}_1, \widetilde{P}_{2,i}, L)_{upd(E, \sigma \lfloor\!\lfloor \theta_{2,i})} + \\ & \sum_{i \in I_1, act(\theta_{1,i}) \in L} \; \sum_{j \in I_2, act(\theta_{2,j}) = act(\theta_{1,i})} \ell_{\rm brs}(\sigma \langle \theta_{1,i}, \theta_{2,j} \rangle)_{upd(E, \sigma \langle \theta_{1,i}, \theta_{2,j} \rangle)} . e\ell^\sigma_{\rm brs}(\widetilde{P}_{1,i}, \widetilde{P}_{2,j}, L))_{upd(E, \sigma \langle \theta_{1,i}, \theta_{2,j} \rangle)} \end{aligned}$$
where each of the three summation-shaped subprocesses on the right is an initial process.
- If $\widetilde{P}_1$ is not initial while $\widetilde{P}_2$ is initial, say $\widetilde{P}_1 = \ell_{\rm brs}(\theta_1)^\dagger_{upd(to\_initial(P_1), \theta_1)} . \widetilde{P}'_1 \, [+ \widetilde{P}''_1]$ where $act(\theta_1) \notin L$ and $\widetilde{P}''_1$ is initial, say $\widetilde{P}''_1 = \sum_{i \in I_1} \ell_{\rm brs}(\theta_{1,i})_{upd(P''_1, \theta_{1,i})} . \widetilde{P}''_{1,i}$, and $\widetilde{P}_2 = \sum_{i \in I_2} \ell_{\rm brs}(\theta_{2,i})_{upd(P_2, \theta_{2,i})} . \widetilde{P}_{2,i}$, for $\ddot{E}$ obtained from E by replacing P_1 with $to\_initial(P_1)$ let $e\ell^\sigma_{\rm brs}(\widetilde{P}_1, \widetilde{P}_2, L)_E$
$$\begin{aligned} = \; & \ell_{\rm brs}(\sigma \rfloor\!\!\rfloor \theta_1)^\dagger{}_{upd(\ddot{E}, \sigma \rfloor\!\rfloor \theta_1)} . e\ell^\sigma_{\rm brs}(\widetilde{P}'_1, \widetilde{P}_2, L)_E + \\ & [\textstyle\sum_{i \in I_1, act(\theta_{1,i}) \notin L} \ell_{\rm brs}(\sigma \rfloor\!\!\rfloor \theta_{1,i})_{upd(\ddot{E}, \sigma \rfloor\!\rfloor \theta_{1,i})} . e\ell^\sigma_{\rm brs}(\widetilde{P}''_{1,i}, \widetilde{P}_2, L)_{upd(\ddot{E}, \sigma \rfloor\!\rfloor \theta_{1,i})} +] \\ & \textstyle\sum_{i \in I_2, act(\theta_{2,i}) \notin L} \ell_{\rm brs}(\sigma \lfloor\!\!\lfloor \theta_{2,i})_{upd(\ddot{E}, \sigma \lfloor\!\lfloor \theta_{2,i})} . e\ell^\sigma_{\rm brs}(to\_initial(\widetilde{P}_1), \widetilde{P}_{2,i}, L)_{upd(\ddot{E}, \sigma \lfloor\!\lfloor \theta_{2,i})} + \\ & [\sum_{i \in I_1, act(\theta_{1,i}) \in L} \; \sum_{j \in I_2, act(\theta_{2,j}) = act(\theta_{1,i})} \ell_{\rm brs}(\sigma \langle \theta_{1,i}, \theta_{2,j} \rangle)_{upd(\ddot{E}, \sigma \langle \theta_{1,i}, \theta_{2,j} \rangle)} . e\ell^\sigma_{\rm brs}(\widetilde{P}''_{1,i}, \widetilde{P}_{2,j}, L))_{upd(\ddot{E}, \sigma \langle \theta_{1,i}, \theta_{2,j} \rangle)}] \end{aligned}$$
where each of the last three summation-shaped subprocesses on the right is an initial process needed by the forward-reverse semantics, with the presence of the first one and the third one depending on the presence of $\widetilde{P}''_1$.
- The case in which $\widetilde{P}_1$ is initial while $\widetilde{P}_2$ is not initial is like the previous one.
- If $\widetilde{P}_1$ and $\widetilde{P}_2$ are both non-initial, say $\widetilde{P}_k = \ell_{\rm brs}(\theta_k)^\dagger_{upd(to\_initial(P_k), \theta_k)} . \widetilde{P}'_k \, [+ \widetilde{P}''_k]$ where $\widetilde{P}''_k$ is initial, say $\widetilde{P}''_k = \sum_{i \in I_k} \ell_{\rm brs}(\theta_{k,i})_{upd(P''_k, \theta_{k,i})} . \widetilde{P}''_{k,i}$, for $k \in \{1,2\}$, for $\ddot{E}$ obtained from E by replacing each P_k with $to\_initial(P_k)$ there are three subcases:
  - If $act(\theta_1) \notin L \wedge (act(\theta_2) \in L \vee \sigma \rfloor\!\!\rfloor \theta_1 \leq_\dagger \sigma \lfloor\!\!\lfloor \theta_2)$, let $e\ell^\sigma_{\rm brs}(\widetilde{P}_1, \widetilde{P}_2, L)_E$
$$\begin{aligned} = \; & \ell_{\rm brs}(\sigma \rfloor\!\!\rfloor \theta_1)^\dagger{}_{upd(\ddot{E}, \sigma \rfloor\!\rfloor \theta_1)} . e\ell^\sigma_{\rm brs}(\widetilde{P}'_1, \widetilde{P}_2, L)_E + \\ & [\ell_{\rm brs}(\sigma \lfloor\!\!\lfloor \theta_2)_{upd(\ddot{E}, \sigma \lfloor\!\lfloor \theta_2)} . e\ell^\sigma_{\rm brs}(to\_initial(\widetilde{P}_1), to\_initial(\widetilde{P}'_2), L)_{upd(\ddot{E}, \sigma \lfloor\!\lfloor \theta_2)} +] \\ & [\sum_{i \in I_1, act(\theta_{1,i}) \notin L} \ell_{\rm brs}(\sigma \rfloor\!\!\rfloor \theta_{1,i})_{upd(\ddot{E}, \sigma \rfloor\!\rfloor \theta_{1,i})} . e\ell^\sigma_{\rm brs}(\widetilde{P}''_{1,i}, to\_initial(\widetilde{P}_2), L)_{upd(\ddot{E}, \sigma \rfloor\!\rfloor \theta_{1,i})} +] \\ & [\sum_{i \in I_2, act(\theta_{2,i}) \notin L} \ell_{\rm brs}(\sigma \lfloor\!\!\lfloor \theta_{2,i})_{upd(\ddot{E}, \sigma \lfloor\!\lfloor \theta_{2,i})} . e\ell^\sigma_{\rm brs}(to\_initial(\widetilde{P}_1), \widetilde{P}''_{2,i}, L)_{upd(\ddot{E}, \sigma \lfloor\!\lfloor \theta_{2,i})} +] \\ & [\sum_{i \in I_1, act(\theta_{1,i}) \in L} \; \sum_{j \in I_2, act(\theta_{2,j}) = act(\theta_{1,i})} \ell_{\rm brs}(\sigma \langle \theta_{1,i}, \theta_{2,j} \rangle)_{upd(\ddot{E}, \sigma \langle \theta_{1,i}, \theta_{2,j} \rangle)} . e\ell^\sigma_{\rm brs}(\widetilde{P}''_{1,i}, \widetilde{P}''_{2,j}, L))_{upd(\ddot{E}, \sigma \langle \theta_{1,i}, \theta_{2,j} \rangle)}] \end{aligned}$$

where each of the last four subprocesses on the right is an initial process needed by the forward-reverse semantics, with the first one being present only if $act(\theta_2) \notin L$ and the presence of the subsequent three respectively depending on the presence of $\widetilde{P}''_1$, $\widetilde{P}''_2$, or both.

- The subcase $act(\theta_2) \notin L \wedge (act(\theta_1) \in L \vee \sigma\lfloor\!\lfloor\theta_2 \leq_{\dagger} \sigma\rfloor\!\rfloor\theta_1)$ is like the previous one.
- If $act(\theta_1) = act(\theta_2) \in L$, let $e\ell^{\sigma}_{\rm brs}(\widetilde{P}_1, \widetilde{P}_2, L)_E$

$$= \ell_{\rm brs}(\sigma\langle\theta_1,\theta_2\rangle)^{\dagger}{}_{upd(\ddot{E},\sigma\langle\theta_1,\theta_2\rangle)} \,.\, e\ell^{\sigma}_{\rm brs}(\widetilde{P}'_1, \widetilde{P}'_2, L))_E \, +$$
$$[\sum_{i\in I_1, act(\theta_{1,i})\notin L} \ell_{\rm brs}(\sigma\rfloor\!\rfloor\theta_{1,i})_{upd(\ddot{E},\sigma\rfloor\!\rfloor\theta_{1,i})} \,.\, e\ell^{\sigma}_{\rm brs}(\widetilde{P}''_{1,i}, to\_initial(\widetilde{P}_2), L)_{upd(\ddot{E},\sigma\rfloor\!\rfloor\theta_{1,i})} \, +]$$
$$[\sum_{i\in I_2, act(\theta_{2,i})\notin L} \ell_{\rm brs}(\sigma\lfloor\!\lfloor\theta_{2,i})_{upd(\ddot{E},\sigma\lfloor\!\lfloor\theta_{2,i})} \,.\, e\ell^{\sigma}_{\rm brs}(to\_initial(\widetilde{P}_1), \widetilde{P}''_{2,i}, L)_{upd(\ddot{E},\sigma\lfloor\!\lfloor\theta_{2,i})} \, +]$$
$$[\sum_{i\in I_1, act(\theta_{1,i})\in L} \; \sum_{j\in I_2, act(\theta_{2,j})=act(\theta_{1,i})} \ell_{\rm brs}(\sigma\langle\theta_{1,i},\theta_{2,j}\rangle)_{upd(\ddot{E},\sigma\langle\theta_{1,i},\theta_{2,j}\rangle)} \,.\, e\ell^{\sigma}_{\rm brs}(\widetilde{P}''_{1,i}, \widetilde{P}''_{2,j}, L))_{upd(\ddot{E},\sigma\langle\theta_{1,i},\theta_{2,j}\rangle)}]$$

where each of the last three summation-shaped subprocesses on the right is an initial process needed by the forward-reverse semantics, with their presence respectively depending on the presence of $\widetilde{P}''_1$, $\widetilde{P}''_2$, or both. ■

Example 5.4 Let P be $P_1 \,\|_{\emptyset}\, P_2$, where P_1 and P_2 are the initial sequential processes $a\,.\,\underline{0}$ and $b\,.\,\underline{0}$ so that $\widetilde{P}_1 = \ell_{\rm brs}(a)_{a^{\dagger}.\underline{0}} \,.\, \widetilde{\underline{0}}$ and $\widetilde{P}_2 = \ell_{\rm brs}(b)_{b^{\dagger}.\underline{0}} \,.\, \widetilde{\underline{0}}$. Then:

$$\begin{aligned} \widetilde{P} = \ell^{\varepsilon}_{\rm brs}(P)_P \;&=\; e\ell^{\varepsilon}_{\rm brs}(\widetilde{P}_1, \widetilde{P}_2, \emptyset)_P \\ &=\; \ell_{\rm brs}(\rfloor\!\rfloor a)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{\underline{0}}, \widetilde{P}_2, \emptyset)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b.\underline{0}} \, + \\ &\quad\;\; \ell_{\rm brs}(\lfloor\!\lfloor b)_{a.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{P}_1, \widetilde{\underline{0}}, \emptyset)_{a.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \\ &=\; <a,\{a\}> .\, \ell_{\rm brs}(\lfloor\!\lfloor b)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \, + \\ &\quad\;\; <b,\{b\}> .\, \ell_{\rm brs}(\rfloor\!\rfloor a)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \\ &=\; <a,\{a\}> .<b,\{a,b\}> .\,\underline{0} + <b,\{b\}> .<a,\{a,b\}> .\,\underline{0} \end{aligned}$$

which is different from the encoding of $a\,.\,b\,.\,\underline{0} + b\,.\,a\,.\,\underline{0}$ shown in Example 5.2, unless $a = b$ as in that case the backward ready set $\{a,b\}$ collapses to $\{a\}$.

If instead P_1 is the non-initial sequential process $a^{\dagger}.\,\underline{0}$ and P_2 is the initial sequential process $b\,.\,\underline{0}$, so that $\widetilde{P}_1 = \ell_{\rm brs}(a)^{\dagger}_{a^{\dagger}.\underline{0}} \,.\, \widetilde{\underline{0}}$ and $\widetilde{P}_2 = \ell_{\rm brs}(b)_{b^{\dagger}.\underline{0}} \,.\, \widetilde{\underline{0}}$, then:

$$\begin{aligned} \widetilde{P} = \ell^{\varepsilon}_{\rm brs}(P)_P \;&=\; e\ell^{\varepsilon}_{\rm brs}(\widetilde{P}_1, \widetilde{P}_2, \emptyset)_P \\ &=\; \ell_{\rm brs}(\rfloor\!\rfloor a)^{\dagger}_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{\underline{0}}, \widetilde{P}_2, \emptyset)_P \, + \\ &\quad\;\; \ell_{\rm brs}(\lfloor\!\lfloor b)_{a.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\ell_{\rm brs}(a)_{a^{\dagger}.\underline{0}} \,.\, \widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{a.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \\ &=\; <a^{\dagger},\{a\}> .\, \ell_{\rm brs}(\lfloor\!\lfloor b)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \, + \\ &\quad\;\; <b,\{b\}> .\, \ell_{\rm brs}(\rfloor\!\rfloor a)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \\ &=\; <a^{\dagger},\{a\}> .<b,\{a,b\}> .\,\underline{0} + <b,\{b\}> .<a,\{b,a\}> .\,\underline{0} \end{aligned}$$

If finally P_1 is the non-initial sequential process $a^{\dagger}.\,\underline{0}$ and P_2 is the non-initial sequential process $b^{\dagger}.\,\underline{0}$, so that $\widetilde{P}_1 = \ell_{\rm brs}(a)^{\dagger}_{a^{\dagger}.\underline{0}} \,.\, \widetilde{\underline{0}}$ and $\widetilde{P}_2 = \ell_{\rm brs}(b)^{\dagger}_{b^{\dagger}.\underline{0}} \,.\, \widetilde{\underline{0}}$, then for $\rfloor\!\rfloor a \leq_{\dagger} \lfloor\!\lfloor b$:

$$\begin{aligned} \widetilde{P} = \ell^{\varepsilon}_{\rm brs}(P)_P \;&=\; e\ell^{\varepsilon}_{\rm brs}(\widetilde{P}_1, \widetilde{P}_2, \emptyset)_P \\ &=\; \ell_{\rm brs}(\rfloor\!\rfloor a)^{\dagger}_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{\underline{0}}, \widetilde{P}_2, \emptyset)_P \, + \\ &\quad\;\; \ell_{\rm brs}(\lfloor\!\lfloor b)_{a.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\ell_{\rm brs}(a)_{a^{\dagger}.\underline{0}} \,.\, \widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{a.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \\ &=\; <a^{\dagger},\{a\}> .\, \ell_{\rm brs}(\lfloor\!\lfloor b)^{\dagger}_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \, + \\ &\quad\;\; <b,\{b\}> .\, \ell_{\rm brs}(\rfloor\!\rfloor a)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \,.\, e\ell^{\varepsilon}_{\rm brs}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{a^{\dagger}.\underline{0}\,\|_{\emptyset}\, b^{\dagger}.\underline{0}} \\ &=\; <a^{\dagger},\{a\}> .<b^{\dagger},\{a,b\}> .\,\underline{0} + <b,\{b\}> .<a,\{b,a\}> .\,\underline{0} \end{aligned}$$

■

We now investigate the correctness of the $\ell_{\rm brs}$-encoding. After some compositionality properties, we show that the encoding preserves initiality and – to a large extent – backward ready sets.

$(\mathscr{A}_{\mathrm{R},1})$	$\widetilde{(P+Q)+R}$	$=$	$\widetilde{P+(Q+R)}$	where at least two among P, Q, R are initial
$(\mathscr{A}_{\mathrm{R},2})$	$\widetilde{P+Q}$	$=$	$\widetilde{Q+P}$	where at least one between P and Q is initial
$(\mathscr{A}_{\mathrm{R},3})$	$\widetilde{a\,.\,P}$	$=$	$\widetilde{P}$	where $initial(P)$
$(\mathscr{A}_{\mathrm{R},4})$	$\widetilde{P+Q}$	$=$	$\widetilde{P}$	if $initial(Q)$
$(\mathscr{A}_{\mathrm{R},5})$	$\widetilde{P_1 \parallel_L P_2}$	$=$	$e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P}_1, \widetilde{P}_2, L)_{P_1 \parallel_L P_2}$	with P_k in R-nf for $k \in \{1,2\}$
$(\mathscr{A}_{\mathrm{FR},1})$	$\widetilde{(P+Q)+R}$	$=$	$\widetilde{P+(Q+R)}$	where at least two among P, Q, R are initial
$(\mathscr{A}_{\mathrm{FR},2})$	$\widetilde{P+Q}$	$=$	$\widetilde{Q+P}$	where at least one between P and Q is initial
$(\mathscr{A}_{\mathrm{FR},3})$	$\widetilde{P+\underline{0}}$	$=$	$\widetilde{P}$	
$(\mathscr{A}_{\mathrm{FR},4})$	$\widetilde{P+Q}$	$=$	$\widetilde{P}$	if $initial(Q) \wedge to\_initial(P) = Q$
$(\mathscr{A}_{\mathrm{FR},5})$	$\widetilde{P_1 \parallel_L P_2}$	$=$	$e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P}_1, \widetilde{P}_2, L)_{P_1 \parallel_L P_2}$	with P_k in FR-nf for $k \in \{1,2\}$

Table 4: Axioms characterizing $\sim_{\mathrm{RB}}$ and $\sim_{\mathrm{FRB}}$ via the ℓ_{brs}-encoding into $\mathbb{P}_{\mathrm{brs}}$ processes

Lemma 5.5 Let $a \in A$ and $P, P_1, P_2 \in \mathbb{P}$ be such that $a\,.\,P, P_1 + P_2 \in \mathbb{P}$. Then:

1. $\widetilde{a\,.\,P} = {<}a, \{a\}{>}\,.\,\widetilde{P}$.
2. $\widetilde{a^{\dagger}\,.\,P} = {<}a^{\dagger}, brs(a^{\dagger}\,.\,P){>}\,.\,\widetilde{P}$, with $brs(a^{\dagger}\,.\,P) = \{a\}$ if P is initial.
3. $\widetilde{P_1 + P_2} = \widetilde{P}_1 + \widetilde{P}_2$. ■

Proposition 5.6 Let $P \in \mathbb{P}$. Then:

1. $initial(\widetilde{P})$ iff $initial(P)$.
2. $brs(\widetilde{P}) = brs(P)$ if P has no subprocesses of the form $P_1 \parallel_L P_2$ such that P_1 and P_2 are non-initial and the last executed action $b_1^{\dagger}$ in $\widetilde{P}_1$ is different from the last executed action $b_2^{\dagger}$ in $\widetilde{P}_2$ with $b_1, b_2 \notin L$. ■

As an example, for P given by $a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}$ we have that $\widetilde{P} = {<}a^{\dagger}, \{a\}{>}\,.\,{<}b^{\dagger}, \{a,b\}{>}\,.\,\underline{0} + {<}b, \{b\}{>}\,.\,{<}a, \{a,b\}{>}\,.\,\underline{0}$ when the last executed actions satisfy $\rfloor\!\rfloor a \leq_{\dagger} \rfloor\!\rfloor b$ (see end of Example 5.4), hence $brs(P) = \{a,b\}$ but $brs(\widetilde{P}) = \{b\}$ for $a \neq b$. However, in $\widetilde{P}$ the backward ready set $\{a,b\}$ occurs next to the last executed action $b^{\dagger}$, hence it will label the related transition in $\longrightarrow_{\mathrm{brs}}$ (see Table 3). Indeed, the ℓ_{brs}-encoding is correct in the following sense.

Theorem 5.7 Let $P, P' \in \mathbb{P}$ and $\theta \in \Theta$. Then $P \xrightarrow{\theta} P'$ iff $\widetilde{P} \xrightarrow{\ell_{\mathrm{brs}}(\theta)_{P'}}_{\mathrm{brs}} \widetilde{P}'$. ■

Corollary 5.8 Let $P_1, P_2 \in \mathbb{P}$ and $B \in \{\mathrm{RB}, \mathrm{FRB}\}$. Then $P_1 \sim_B P_2$ iff $\widetilde{P}_1 \sim_{B:\ell_{\mathrm{brs}}} \widetilde{P}_2$. ■

The set $\mathscr{A}_{\mathrm{R}}$ of axioms for $\sim_{\mathrm{RB}}$ is shown in the upper part of Table 4. All the axioms apart from the last one come from the axiomatization developed in [13] over sequential processes. Axiom $\mathscr{A}_{\mathrm{R},3}$ establishes that the future can be completely canceled when moving only backward. Likewise, axiom $\mathscr{A}_{\mathrm{R},4}$ states that a previously non-selected alternative can be discarded when moving only backward; note that this axiom subsumes both $\widetilde{P+\underline{0}} = \widetilde{P}$ and $\widetilde{P+P} = \widetilde{P}$. The new axiom $\mathscr{A}_{\mathrm{R},5}$ concisely expresses via $e\ell_{\mathrm{brs}}$ the expansion laws for reverse bisimilarity, where P_k is $\underline{0}$ or the $+$-free sequential process $a_k^{\dagger}\,.\,P_k'$ featuring only executed actions for $k \in \{1,2\}$.

Definition 5.9 We say that $P \in \mathbb{P}$ is in *reverse normal form*, written *R-nf*, iff it is equal to $\underline{0}$ or $a^\dagger . P'$ where P' is in R-nf. This extends to $\widetilde{P} \in \mathbb{P}_{\rm brs}$ in the expected way. ■

Lemma 5.10 For all (initial) $P \in \mathbb{P}$ there exists (an initial) $\widetilde{Q} \in \mathbb{P}_{\rm brs}$ in R-nf (which is $\widetilde{\underline{0}}$) such that $\mathscr{A}_{\rm R} \vdash \widetilde{P} = \widetilde{Q}$. ■

Theorem 5.11 Let $P_1, P_2 \in \mathbb{P}$. Then $\widetilde{P_1} \sim_{{\rm RB}:\ell_{\rm brs}} \widetilde{P_2}$ iff $\mathscr{A}_{\rm R} \vdash \widetilde{P_1} = \widetilde{P_2}$. ■

The set $\mathscr{A}_{\rm FR}$ of axioms for $\sim_{\rm FRB}$ is shown in the lower part of Table 4. All the axioms apart from the last one come from the axiomatization developed in [13] over sequential processes. Axiom $\mathscr{A}_{\rm FR,4}$ is a variant of idempotency appeared for the first time in [39], with P and Q coinciding like in axiom $\mathscr{A}_{\rm F,4}$ when they are both initial. The new axiom $\mathscr{A}_{\rm FR,5}$ concisely expresses via $e\ell_{\rm brs}$ the expansion laws for forward-reverse bisimilarity, where P_k is the sequential process $[a_k^\dagger . P'_k +] \sum_{i \in I_k} a_{k,i} . P_{k,i}$ for $k \in \{1,2\}$.

Definition 5.12 We say that $P \in \mathbb{P}$ is in *forward-reverse normal form*, written *FR-nf*, iff it is equal to $[b^\dagger . P' +] \sum_{i \in I} a_i . P_i$ where $b^\dagger . P'$ is optional, P' is in FR-nf, I is a finite index set (with the summation being $\underline{0}$ – or disappearing in the presence of $b^\dagger . P'$ – when $I = \emptyset$), and each P_i is initial and in FR-nf. This extends to $\widetilde{P} \in \mathbb{P}_{\rm brs}$ in the expected way. ■

Lemma 5.13 For all (initial) $P \in \mathbb{P}$ there exists (an initial) $\widetilde{Q} \in \mathbb{P}_{\rm brs}$ in FR-nf such that $\mathscr{A}_{\rm FR} \vdash \widetilde{P} = \widetilde{Q}$. ■

Theorem 5.14 Let $P_1, P_2 \in \mathbb{P}$. Then $\widetilde{P_1} \sim_{{\rm FRB}:\ell_{\rm brs}} \widetilde{P_2}$ iff $\mathscr{A}_{\rm FR} \vdash \widetilde{P_1} = \widetilde{P_2}$. ■

6 Conclusions

In this paper we have exhibited expansion laws for forward bisimilarity, which is interleaving, and reverse and forward-reverse bisimilarities, which are truly concurrent. To uniformly develop them, we have resorted to the proved trees approach of [27], which has turned out to be effective also in our setting. With respect to other truly concurrent semantics to which the approach was applied, such as causal and location bisimilarities, the operational semantics of our reversible calculus does not carry the additional discriminating information within transition labels. However, we have been able to derive it from those labels and shown to consist of backward ready sets for both reverse and forward-reverse bisimilarities. Another technical difficulty that we have faced is the encoding of concurrent processes in which both subprocesses have executed non-synchronizing actions, because their expansions cannot contain executed actions on both sides of an alternative composition. For completeness we mention that in [1] proved semantics has already been employed in a reversible setting, for a different purpose though.

As for future work, an obvious direction is to exploit the same approach to find out expansion laws for the weak versions of forward, reverse, and forward-reverse bisimilarities, i.e., their versions capable of abstracting from τ-actions [8].

A more interesting direction is to show that forward-reverse bisimilarity augmented with a clause for backward ready *multi*sets equality corresponds to hereditary history-preserving bisimilarity [5], thus yielding for the latter an operational characterization, an axiomatization alternative to [29], and logical characterizations alternative to [48, 4]. These two bisimilarities were shown to coincide in [5, 46, 47, 2] in the absence of autoconcurrency. In fact, if $a = b$ in Figure 1, the two processes turn out to be forward-reverse bisimilar, with the backward ready sets of the three bottom states collapsing to $\{a\}$, but not hereditary history-preserving bisimilar, because *identifying* executed actions is important [3] (as done also in CCSK via communication keys [45]). However, if backward ready multisets are used instead, then the bottom state on the left can be distinguished from the two bottom states on the right. Thus, *counting* executed actions that label incoming transitions may be enough.

Acknowledgments. We would like to thank Irek Ulidowski, Ilaria Castellani, and Pierpaolo Degano for the valuable discussions. 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] C. Aubert (2022): *Concurrencies in Reversible Concurrent Calculi*. In: Proc. of the 14th Int. Conf. on Reversible Computation (RC 2022), LNCS 13354, Springer, pp. 146–163, doi:10.1007/978-3-031-09005-9_10.

[2] C. Aubert & I. Cristescu (2017): *Contextual Equivalences in Configuration Structures and Reversibility*. Journal of Logical and Algebraic Methods in Programming 86, pp. 77–106, doi:10.1016/j.jlamp.2016.08.004.

[3] C. Aubert & I. Cristescu (2020): *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 171, pp. 7:1–7:23, doi:10.4230/LIPIcs.CONCUR.2020.7.

[4] P. Baldan & S. Crafa (2014): *A Logic for True Concurrency*. Journal of the ACM 61, pp. 24:1–24:36, doi:10.1145/2629638.

[5] M.A. Bednarczyk (1991): *Hereditary History Preserving Bisimulations or What Is the Power of the Future Perfect in Program Logics*. Technical Report, Polish Academy of Sciences, Gdansk.

[6] C.H. Bennett (1973): *Logical Reversibility of Computation*. IBM Journal of Research and Development 17, pp. 525–532, doi:10.1147/rd.176.0525.

[7] J.A. Bergstra, J.W. Klop & E.-R. Olderog (1988): *Readies and Failures in the Algebra of Communicating Processes*. SIAM Journal on Computing 17, pp. 1134–1177, doi:10.1137/0217073.

[8] M. Bernardo & A. Esposito (2023): *On the Weak Continuation of Reverse Bisimilarity vs. Forward Bisimilarity*. In: Proc. of the 24th Italian Conf. on Theoretical Computer Science (ICTCS 2023), CEUR-WS 3587, pp. 44–58.

[9] M. Bernardo & A. Esposito (2023): *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 390, pp. 67–81, doi:10.4204/EPTCS.390.5.

[10] M. Bernardo & C.A. Mezzina (2023): *Bridging Causal Reversibility and Time Reversibility: A Stochastic Process Algebraic Approach*. Logical Methods in Computer Science 19(2), pp. 6:1–6:27, doi:10.46298/lmcs-19(2:6)2023.

[11] M. Bernardo & C.A. Mezzina (2023): *Causal Reversibility for Timed Process Calculi with Lazy/Eager Durationless Actions and Time Additivity*. In: Proc. of the 21st Int. Conf. on Formal Modeling and Analysis of Timed Systems (FORMATS 2023), LNCS 14138, Springer, pp. 15–32, doi:10.1007/978-3-031-42626-1_2.

[12] M. Bernardo & C.A. Mezzina (2024): *Reversibility in Process Calculi with Nondeterminism and Probabilities*. In: Proc. of the 21st Int. Coll. on Theoretical Aspects of Computing (ICTAC 2024), LNCS, Springer.

[13] M. Bernardo & S. Rossi (2023): *Reverse Bisimilarity vs. Forward Bisimilarity*. In: Proc. of the 26th Int. Conf. on Foundations of Software Science and Computation Structures (FOSSACS 2023), LNCS 13992, Springer, pp. 265–284, doi:10.1007/978-3-031-30829-1_13.

[14] L. Bocchi, I. Lanese, C.A. Mezzina & S. Yuen (2024): *revTPL: The Reversible Temporal Process Language*. Logical Methods in Computer Science 20(1), pp. 11:1–11:35, doi:10.46298/lmcs-20(1:11)2024.

[15] G. Boudol & I. Castellani (1988): *Concurrency and Atomicity*. Theoretical Computer Science 59, pp. 25–84, doi:10.1016/0304-3975(88)90096-5.

[16] G. Boudol & I. Castellani (1988): *A Non-Interleaving Semantics for CCS Based on Proved Transitions*. *Fundamenta Informaticae* 11, pp. 433–452, doi:10.3233/FI-1988-11406.

[17] G. Boudol & I. Castellani (1994): *Flow Models of Distributed Computations: Three Equivalent Semantics for CCS*. *Information and Computation* 114, pp. 247–314, doi:10.1006/inco.1994.1088.

[18] G. Boudol, I. Castellani, M. Hennessy & A. Kiehn (1994): *A Theory of Processes with Localities*. *Formal Aspects of Computing* 6, pp. 165–200, doi:10.1007/BF01221098.

[19] S.D. Brookes, C.A.R. Hoare & A.W. Roscoe (1984): *A Theory of Communicating Sequential Processes*. *Journal of the ACM* 31, pp. 560–599, doi:10.1145/828.833.

[20] I. Castellani (1995): *Observing Distribution in Processes: Static and Dynamic Localities*. *Foundations of Computer Science* 6, pp. 353–393, doi:10.1142/S0129054195000196.

[21] I. Cristescu, J. Krivine & D. Varacca (2013): *A Compositional Semantics for the Reversible P-Calculus*. In: *Proc. of the 28th ACM/IEEE Symp. on Logic in Computer Science (LICS 2013)*, IEEE-CS Press, pp. 388–397, doi:10.1109/LICS.2013.45.

[22] V. Danos & J. Krivine (2004): *Reversible Communicating Systems*. In: *Proc. of the 15th Int. Conf. on Concurrency Theory (CONCUR 2004)*, *LNCS* 3170, Springer, pp. 292–307, doi:10.1007/978-3-540-28644-8_19.

[23] V. Danos & J. Krivine (2005): *Transactions in RCCS*. In: *Proc. of the 16th Int. Conf. on Concurrency Theory (CONCUR 2005)*, *LNCS* 3653, Springer, pp. 398–412, doi:10.1007/11539452_31.

[24] Ph. Darondeau & P. Degano (1989): *Causal Trees*. In: *Proc. of the 16th Int. Coll. on Automata, Languages and Programming (ICALP 1989)*, *LNCS* 372, Springer, pp. 234–248, doi:10.1007/BFb0035764.

[25] Ph. Darondeau & P. Degano (1990): *Causal Trees: Interleaving + Causality*. In: *Proc. of the LITP Spring School on Theoretical Computer Science: Semantics of Systems of Concurrent Processes*, *LNCS* 469, Springer, pp. 239–255, doi:10.1007/3-540-53479-2_10.

[26] R. De Nicola, U. Montanari & F. Vaandrager (1990): *Back and Forth Bisimulations*. In: *Proc. of the 1st Int. Conf. on Concurrency Theory (CONCUR 1990)*, *LNCS* 458, Springer, pp. 152–165, doi:10.1007/BFb0039058.

[27] P. Degano & C. Priami (1992): *Proved Trees*. In: *Proc. of the 19th Int. Coll. on Automata, Languages and Programming (ICALP 1992)*, *LNCS* 623, Springer, pp. 629–640, doi:10.1007/3-540-55719-9_110.

[28] H. Fecher (2004): *A Completed Hierarchy of True Concurrent Equivalences*. *Information Processing Letters* 89, pp. 261–265, doi:10.1016/j.ipl.2003.11.008.

[29] S. Fröschle & S. Lasota (2005): *Decomposition and Complexity of Hereditary History Preserving Bisimulation on BPP*. In: *Proc. of the 16th Int. Conf. on Concurrency Theory (CONCUR 2005)*, *LNCS* 3653, Springer, pp. 263–277, doi:10.1007/11539452_22.

[30] E. Giachino, I. Lanese & C.A. Mezzina (2014): *Causal-Consistent Reversible Debugging*. In: *Proc. of the 17th Int. Conf. on Fundamental Approaches to Software Engineering (FASE 2014)*, *LNCS* 8411, Springer, pp. 370–384, doi:10.1007/978-3-642-54804-8_26.

[31] R.J. van Glabbeek & U. Goltz (2001): *Refinement of Actions and Equivalence Notions for Concurrent Systems*. *Acta Informatica* 37, pp. 229–327, doi:10.1007/s002360000041.

[32] M. Hennessy & R. Milner (1985): *Algebraic Laws for Nondeterminism and Concurrency*. *Journal of the ACM* 32, pp. 137–162, doi:10.1145/2455.2460.

[33] J. Krivine (2012): *A Verification Technique for Reversible Process Algebra*. In: *Proc. of the 4th Int. Workshop on Reversible Computation (RC 2012)*, *LNCS* 7581, Springer, pp. 204–217, doi:10.1007/978-3-642-36315-3_17.

[34] R. Landauer (1961): *Irreversibility and Heat Generation in the Computing Process*. *IBM Journal of Research and Development* 5, pp. 183–191, doi:10.1147/rd.53.0183.

[35] I. Lanese, M. Lienhardt, C.A. Mezzina, A. Schmitt & J.-B. Stefani (2013): *Concurrent Flexible Reversibility*. In: *Proc. of the 22nd European Symp. on Programming (ESOP 2013)*, *LNCS* 7792, Springer, pp. 370–390, doi:10.1007/978-3-642-37036-6_21.

[36] I. Lanese, D. Medić & C.A. Mezzina (2021): *Static versus Dynamic Reversibility in CCS*. *Acta Informatica* 58, pp. 1–34, doi:10.1007/s00236-019-00346-6.

[37] I. Lanese, C.A. Mezzina & J.-B. Stefani (2010): *Reversing Higher-Order Pi*. In: *Proc. of the 21st Int. Conf. on Concurrency Theory (CONCUR 2010)*, *LNCS* 6269, Springer, pp. 478–493, doi:10.1007/978-3-642-15375-4_33.

[38] I. Lanese, N. Nishida, A. Palacios & G. Vidal (2018): *CauDEr: A Causal-Consistent Reversible Debugger for Erlang*. In: *Proc. of the 14th Int. Symp. on Functional and Logic Programming (FLOPS 2018)*, *LNCS* 10818, Springer, pp. 247–263, doi:10.1007/978-3-319-90686-7_16.

[39] I. Lanese & I. Phillips (2021): *Forward-Reverse Observational Equivalences in CCSK*. In: *Proc. of the 13th Int. Conf. on Reversible Computation (RC 2021)*, *LNCS* 12805, Springer, pp. 126–143, doi:10.1007/978-3-030-79837-6_8.

[40] J.S. Laursen, L.-P. Ellekilde & U.P. Schultz (2018): *Modelling Reversible Execution of Robotic Assembly*. *Robotica* 36, pp. 625–654, doi:10.1017/S0263574717000613.

[41] R. Milner (1989): *Communication and Concurrency*. Prentice Hall.

[42] E.-R. Olderog & C.A.R. Hoare (1986): *Specification-Oriented Semantics for Communicating Processes*. *Acta Informatica* 23, pp. 9–66, doi:10.1007/BF00268075.

[43] D. Park (1981): *Concurrency and Automata on Infinite Sequences*. In: *Proc. of the 5th GI Conf. on Theoretical Computer Science*, *LNCS* 104, Springer, pp. 167–183, doi:10.1007/BFb0017309.

[44] K.S. Perumalla & A.J. Park (2014): *Reverse Computation for Rollback-Based Fault Tolerance in Large Parallel Systems – Evaluating the Potential Gains and Systems Effects*. *Cluster Computing* 17, pp. 303–313, doi:10.1007/s10586-013-0277-4.

[45] I. Phillips & I. Ulidowski (2007): *Reversing Algebraic Process Calculi*. *Journal of Logic and Algebraic Programming* 73, pp. 70–96, doi:10.1016/j.jlap.2006.11.002.

[46] I. Phillips & I. Ulidowski (2007): *Reversibility and Models for Concurrency*. In: *Proc. of the 4th Int. Workshop on Structural Operational Semantics (SOS 2007)*, *ENTCS* 192(1), Elsevier, pp. 93–108, doi:10.1016/j.entcs.2007.08.018.

[47] I. Phillips & I. Ulidowski (2012): *A Hierarchy of Reverse Bisimulations on Stable Configuration Structures*. *Mathematical Structures in Computer Science* 22, pp. 333–372, doi:10.1017/S0960129511000429.

[48] I. Phillips & I. Ulidowski (2014): *Event Identifier Logic*. *Mathematical Structures in Computer Science* 24(2), pp. 1–51, doi:10.1017/S0960129513000510.

[49] I. Phillips, I. Ulidowski & S. Yuen (2012): *A Reversible Process Calculus and the Modelling of the ERK Signalling Pathway*. In: *Proc. of the 4th Int. Workshop on Reversible Computation (RC 2012)*, *LNCS* 7581, Springer, pp. 218–232, doi:10.1007/978-3-642-36315-3_18.

[50] G.M. Pinna (2017): *Reversing Steps in Membrane Systems Computations*. In: *Proc. of the 18th Int. Conf. on Membrane Computing (CMC 2017)*, *LNCS* 10725, Springer, pp. 245–261, doi:10.1007/978-3-319-73359-3_16.

[51] A.M. Rabinovich & B.A. Trakhtenbrot (1988): *Behavior Structures and Nets*. *Fundamenta Informaticae* 11, pp. 357–404, doi:10.3233/FI-1988-11404.

[52] M. Schordan, T. Oppelstrup, D.R. Jefferson & P.D. Barnes Jr. (2018): *Generation of Reversible C++ Code for Optimistic Parallel Discrete Event Simulation*. *New Generation Computing* 36, pp. 257–280, doi:10.1007/s00354-018-0038-2.

[53] H. Siljak, K. Psara & A. Philippou (2019): *Distributed Antenna Selection for Massive MIMO Using Reversing Petri Nets*. *IEEE Wireless Communication Letters* 8, pp. 1427–1430, doi:10.1109/LWC.2019.2920128.

[54] M. Vassor & J.-B. Stefani (2018): *Checkpoint/Rollback vs Causally-Consistent Reversibility*. In: *Proc. of the 10th Int. Conf. on Reversible Computation (RC 2018)*, *LNCS* 11106, Springer, pp. 286–303, doi:10.1007/978-3-319-99498-7_20.

[55] E. de Vries, V. Koutavas & M. Hennessy (2010): *Communicating Transactions*. In: *Proc. of the 21st Int. Conf. on Concurrency Theory (CONCUR 2010)*, *LNCS* 6269, Springer, pp. 569–583, doi:10.1007/978-3-642-15375-4_39.

A Proofs of Results

Proof of Proposition 2.7. A straightforward consequence of the definitions of the considered bisimilarities. ■

Proof of Theorem 2.8. Let $\mathscr{B}$ be a $\sim$-bisimulation containing the pair (P_1, P_2). Then:

$$\mathscr{B}' = \{(Q_1 \,\|_L\, Q, Q_2 \,\|_L\, Q) \in \mathbb{P} \times \mathbb{P} \mid (Q_1, Q_2) \in \mathscr{B} \land Q \in \mathbb{P}\}$$

is a $\sim$-bisimulation too because whenever $(Q_1 \,\|_L\, Q, Q_2 \,\|_L\, Q) \in \mathscr{B}'$:

- If $\sim$ considers outgoing transitions, then $Q_1 \,\|_L\, Q \xrightarrow{\theta_1} Q'_1 \,\|_L\, Q$ or $Q_1 \,\|_L\, Q \xrightarrow{\theta_1} Q_1 \,\|_L\, Q'$ for $act(\theta_1) \notin L$ or $Q_1 \,\|_L\, Q \xrightarrow{\theta_1} Q'_1 \,\|_L\, Q'$ for $act(\theta_1) \in L$ is resp. matched by $Q_2 \,\|_L\, Q \xrightarrow{\theta_2} Q'_2 \,\|_L\, Q$ or $Q_2 \,\|_L\, Q \xrightarrow{\theta_2} Q_2 \,\|_L\, Q'$ for $act(\theta_2) \notin L$ or $Q_2 \,\|_L\, Q \xrightarrow{\theta_2} Q'_2 \,\|_L\, Q'$ for $act(\theta_2) \in L$. In the first case and the third case the reason is that, since $(Q_1, Q_2) \in \mathscr{B}$, for all $Q_1 \xrightarrow{\theta'_1} Q'_1$ there exists $Q_2 \xrightarrow{\theta'_2} Q'_2$ such that $act(\theta'_1) = act(\theta'_2)$ and $(Q'_1, Q'_2) \in \mathscr{B}$.
- If $\sim$ considers incoming transitions, then $Q'_1 \,\|_L\, Q \xrightarrow{\theta_1} Q_1 \,\|_L\, Q$ or $Q_1 \,\|_L\, Q' \xrightarrow{\theta_1} Q_1 \,\|_L\, Q$ for $act(\theta_1) \notin L$ or $Q'_1 \,\|_L\, Q' \xrightarrow{\theta_1} Q_1 \,\|_L\, Q$ for $act(\theta_1) \in L$ is resp. matched by $Q'_2 \,\|_L\, Q \xrightarrow{\theta_2} Q_2 \,\|_L\, Q$ or $Q_2 \,\|_L\, Q' \xrightarrow{\theta_2} Q_2 \,\|_L\, Q$ for $act(\theta_2) \notin L$ or $Q'_2 \,\|_L\, Q' \xrightarrow{\theta_2} Q_2 \,\|_L\, Q$ for $act(\theta_2) \in L$. In the first case and the third case the reason is that, since $(Q_1, Q_2) \in \mathscr{B}$, for all $Q'_1 \xrightarrow{\theta'_1} Q_1$ there exists $Q'_2 \xrightarrow{\theta'_2} Q_2$ such that $act(\theta'_1) = act(\theta'_2)$ and $(Q'_1, Q'_2) \in \mathscr{B}$.
- If $\sim$ considers initiality, $initial(Q_1) \Longleftrightarrow initial(Q_2)$ implies $initial(Q_1 \,\|_L\, Q) \Longleftrightarrow initial(Q_2 \,\|_L\, Q)$.

Likewise $\mathscr{B}'' = \{(Q \,\|_L\, Q_1, Q \,\|_L\, Q_2) \in \mathbb{P} \times \mathbb{P} \mid (Q_1, Q_2) \in \mathscr{B} \land Q \in \mathbb{P}\}$ is a $\sim$-bisimulation too. ■

Proof of Lemma 4.2. We proceed by induction on the syntactical structure of $P \in \mathbb{P}$:

- If P is $\underline{0}$, then the result follows by taking Q equal to $\underline{0}$ due to reflexivity.
- If P is $a\,.\,P'$ where P' is initial, then by the induction hypothesis there exists Q' initial and in F-nf such that $\mathscr{A}_{\rm F} \vdash P' = Q'$. The result follows by taking Q equal to $a\,.\,Q'$ – which is in F-nf because Q' is initial and in F-nf – due to substitutivity with respect to action prefix.
- If P is $a^{\dagger}\,.\,P'$, then by the induction hypothesis there exists Q' in F-nf such that $\mathscr{A}_{\rm F} \vdash P' = Q'$. There are two cases:
  - If P' and Q' are both initial, then the result follows by taking Q equal to $a^{\dagger}\,.\,Q'$ – which is in F-nf because Q' is initial and in F-nf – due to substitutivity with respect to executed action prefix.
  - Let P' and Q' be both non-initial. Since Q' is in F-nf and hence features a single executed action prefix at the beginning, i.e., Q' is $b^{\dagger}\,.\,Q''$ with Q'' initial and in F-nf, the result follows by taking Q equal to Q' by virtue of $\mathscr{A}_{\rm F} \vdash P = a^{\dagger}\,.\,Q'$ due to substitutivity with respect to executed action prefix, $\mathscr{A}_{\rm F} \vdash a^{\dagger}\,.\,Q' = Q'$ due to axiom $\mathscr{A}_{\rm F,6}$, and transitivity.
- If P is $P_1 + P_2$, then by the induction hypothesis there exist Q_1 and Q_2 in F-nf such that $\mathscr{A}_{\rm F} \vdash P_1 = Q_1$ and $\mathscr{A}_{\rm F} \vdash P_2 = Q_2$, hence $\mathscr{A}_{\rm F} \vdash P = Q_1 + Q_2$ due to substitutivity with respect to alternative composition. There are three cases:
  - If P_1 and P_2 are both initial, then Q_1 and Q_2 are both initial too and hence the result follows by taking Q equal to $Q_1 + Q_2$, up to an application of axiom $\mathscr{A}_{\rm F,3}$ in the case that $Q_1 + Q_2$ is not in F-nf because Q_1 and Q_2 are not different from $\underline{0}$ (possibly preceded by an application of axiom $\mathscr{A}_{\rm F,2}$ to move the $\underline{0}$ subprocess to the right of $+$) and transitivity.

- If only P_2 is initial, then only Q_2 is initial too and hence the result follows by taking Q equal to Q_1 by virtue of $\mathscr{A}_{\rm F} \vdash Q_1 + Q_2 = Q_1$ due to axiom $\mathscr{A}_{\rm F,7}$ and transitivity.
- If only P_1 is initial, then only Q_1 is initial too and hence the result follows by taking Q equal to Q_2 by virtue of $\mathscr{A}_{\rm F} \vdash Q_1 + Q_2 = Q_2 + Q_1$ due to axiom $\mathscr{A}_{\rm F,2}$, $\mathscr{A}_{\rm F} \vdash Q_2 + Q_1 = Q_2$ due to axiom $\mathscr{A}_{\rm F,7}$, and transitivity.

- If P is $P_1 \,\|_L\, P_2$, then by the induction hypothesis there exist Q_1 and Q_2 in F-nf – say $Q_1 = [a_1^{\dagger}.]Q_1'$ with $Q_1' = \sum_{i \in I_1} a_{1,i} \,.\, Q_{1,i}$ and $Q_2 = [a_2^{\dagger}.]Q_2'$ with $Q_2' = \sum_{i \in I_2} a_{2,i} \,.\, Q_{2,i}$ – such that $\mathscr{A}_{\rm F} \vdash P_1 = Q_1$ and $\mathscr{A}_{\rm F} \vdash P_2 = Q_2$, hence $\mathscr{A}_{\rm F} \vdash P = Q_1 \,\|_L\, Q_2$ due to substitutivity with respect to parallel composition. As a consequence we have that $\mathscr{A}_{\rm F} \vdash P = [a^{\dagger}.](\sum_{i \in I_1, a_{1,i} \notin L} a_{1,i} \,.\, (Q_{1,i} \,\|_L\, Q_2') + \sum_{i \in I_2, a_{2,i} \notin L} a_{2,i} \,.\, (Q_1' \,\|_L\, Q_{2,i}) + \sum_{i \in I_1, a_{1,i} \in L} \sum_{j \in I_2, a_{2,j} = a_{1,i}} a_{1,i} \,.\, (Q_{1,i} \,\|_L\, Q_{2,j}))$ due to axiom $\mathscr{A}_{\rm F,8}$ and transitivity. We recall that Q_1', Q_2', and every $Q_{1,i}$ and $Q_{2,i}$ are all initial and in F-nf. Moreover, thanks to axiom $\mathscr{A}_{\rm F,5}$ we can assume that either $a_1, a_2 \notin L$ or $a_1 = a_2 \in L$ so as to ensure that $Q_1 \,\|_L\, Q_2 \in \mathbb{P}$.
  We now prove that, if $O_1, O_2 \in \mathbb{P}$ are (initial and) in F-nf and such that $O_1 \,\|_L\, O_2 \in \mathbb{P}$, then there exists $O \in \mathbb{P}$ (initial and) in F-nf such that $\mathscr{A}_{\rm F} \vdash O_1 \,\|_L\, O_2 = O$, from which the result will follow due to substitutivity with respect to action prefix, alternative composition, and executed action prefix if any. Let us define the size of $P \in \mathbb{P}$ – an upper bound to the depth of the transition system whose initial state is P – by induction on the syntactical structure of P as follows:
$$\begin{array}{rcl} size(\underline{0}) & = & 0 \\ size(a \,.\, P') & = & 1 + size(P') \\ size(a^{\dagger} \,.\, P') & = & 1 + size(P') \\ size(P_1 + P_2) & = & \max(size(P_1), size(P_2)) \\ size(P_1 \,\|_L\, P_2) & = & size(P_1) + size(P_2) \end{array}$$
  Since the parallel processes that we will encounter are not subprocesses of $O_1 \,\|_L\, O_2$, we proceed by induction on $size(O_1 \,\|_L\, O_2)$:
  - If $size(O_1 \,\|_L\, O_2) = 0$, then $O_1 \,\|_L\, O_2$ is $\underline{0} \,\|_L\, \underline{0}$ where $\mathscr{A}_{\rm F} \vdash \underline{0} \,\|_L\, \underline{0} = \underline{0} + \underline{0} + \underline{0}$ due to axiom $\mathscr{A}_{\rm F,8}$. The result follows by taking O equal to $\underline{0}$ due to axiom $\mathscr{A}_{\rm F,3}$ applied twice, substitutivity with respect to alternative composition, and transitivity.
  - If $size(O_1 \,\|_L\, O_2) > 0$, then $O_1 = [b_1^{\dagger}.]O_1'$ with $O_1' = \sum_{i \in J_1} b_{1,i} \,.\, O_{1,i}$ and $O_2 = [b_2^{\dagger}.]O_2'$ with $O_2' = \sum_{i \in J_2} b_{2,i} \,.\, O_{2,i}$, where at least one of the following holds: $b_1^{\dagger}$ present, $J_1 \neq \emptyset$, $b_2^{\dagger}$ present, $J_2 \neq \emptyset$. Thus $\mathscr{A}_{\rm F} \vdash O_1 \,\|_L\, O_2 = [b^{\dagger}.] \sum_{i \in J_1, b_{1,i} \notin L} b_{1,i} \,.\, (O_{1,i} \,\|_L\, O_2') + \sum_{i \in J_2, b_{2,i} \notin L} b_{2,i} \,.\, (O_1' \,\|_L\, O_{2,i}) + \sum_{i \in J_1, b_{1,i} \in L} \sum_{j \in J_2, b_{2,j} = b_{1,i}} b_{1,i} \,.\, (O_{1,i} \,\|_L\, O_{2,j})$ due to axiom $\mathscr{A}_{\rm F,8}$. The result follows by applying the induction hypothesis to every $O_{1,i} \,\|_L\, O_2'$, $O_1' \,\|_L\, O_{2,i}$, $O_{1,i} \,\|_L\, O_{2,j}$ due to substitutivity with respect to action prefix, alternative composition, and executed action prefix if any, with possible applications of axiom $\mathscr{A}_{\rm F,3}$ (each possibly preceded by an application of axiom $\mathscr{A}_{\rm F,2}$ to move the $\underline{0}$ subprocess to the right of $+$). ■

Proof of Theorem 4.3. Soundness, i.e., $\mathscr{A}_{\rm F} \vdash P_1 = P_2 \Longrightarrow P_1 \sim_{\rm FB:ps} P_2$, is a straightforward consequence of the axioms and inference rules behind $\vdash$ (reflexivity, symmetry, transitivity, substitutivity) together with $\sim_{\rm FB:ps}$ being an equivalence relation and a congruence, plus the fact that the lefthand side process of each additional axiom in $\mathscr{A}_{\rm F}$ is $\sim_{\rm FB:ps}$-equivalent to the righthand side process of the same axiom.
Let us address ground completeness, i.e., $P_1 \sim_{\rm FB:ps} P_2 \Longrightarrow \mathscr{A}_{\rm F} \vdash P_1 = P_2$. We suppose that P_1 and P_2 are both in F-nf and proceed by induction on the syntactical structure of P_1:

- If P_1 is $\underline{0}$, then from $P_1 \sim_{\rm FB:ps} P_2$ and P_2 in F-nf we derive that P_2 can only be $\underline{0}$, from which the result follows by reflexivity.

- If P_1 is $[a_1^\dagger.]\sum_{i \in I_1} a_{1,i} \,.\, P_{1,i}$ with $a_1^\dagger$ present or $I_1 \neq \emptyset$, then from $P_1 \sim_{\rm FB:ps} P_2$ and P_2 in F-nf we derive that P_2 can only be $[a_2^\dagger.]\sum_{i \in I_2} a_{2,i} \,.\, P_{2,i}$ with $a_2^\dagger$ present iff $a_1^\dagger$ present and $I_2 \neq \emptyset$ iff $I_1 \neq \emptyset$. We recall that every $P_{1,i}$ and every $P_{2,i}$ is initial and in F-nf.
  Since $P_1 \sim_{\rm FB:ps} P_2$, for each $i_1 \in I_1$ there exists $i_2 \in I_2$ such that $a_{1,i_1} = a_{2,i_2}$ and $P_{1,i_1} \sim_{\rm FB:ps} P_{2,i_2}$, and vice versa. From the induction hypothesis we obtain that $\mathscr{A}_{\rm F} \vdash P_{1,i_1} = P_{2,i_2}$. It then follows that:
  - $\mathscr{A}_{\rm F} \vdash a_{1,i_1} \,.\, P_{1,i_1} = a_{2,i_2} \,.\, P_{2,i_2}$ due to substitutivity with respect to action prefix.
  - $\mathscr{A}_{\rm F} \vdash \sum_{i \in I_1} a_{1,i} \,.\, P_{1,i} = \sum_{i \in I_2} a_{2,i} \,.\, P_{2,i}$ due to substitutivity with respect to alternative composition as well as axiom $\mathscr{A}_{\rm F,4}$ and transitivity in the presence of identical summands on the same side that are absent on the other side (possibly preceded by applications of axioms $\mathscr{A}_{\rm F,1}$ and $\mathscr{A}_{\rm F,2}$ to move identical summands next to each other).
  - $\mathscr{A}_{\rm F} \vdash [a^\dagger.]\sum_{i \in I_1} a_{1,i} \,.\, P_{1,i} = [a^\dagger.]\sum_{i \in I_2} a_{2,i} \,.\, P_{2,i}$ due to substitutivity with respect to executed action prefix.
  - $\mathscr{A}_{\rm F} \vdash [a_1^\dagger.]\sum_{i \in I_1} a_{1,i} \,.\, P_{1,i} = [a_2^\dagger.]\sum_{i \in I_2} a_{2,i} \,.\, P_{2,i}$ due to axiom $\mathscr{A}_{\rm F,5}$ and transitivity.

If P_1 and P_2 are not both in F-nf, thanks to Lemma 4.2 we can find Q_1 and Q_2 in F-nf, each of which is initial iff so is its corresponding original process, such that $\mathscr{A}_{\rm F} \vdash P_1 = Q_1$ and $\mathscr{A}_{\rm F} \vdash P_2 = Q_2$, hence $\mathscr{A}_{\rm F} \vdash Q_2 = P_2$ by symmetry. Due to soundness, we get $P_1 \sim_{\rm FB:ps} Q_1$, hence $Q_1 \sim_{\rm FB:ps} P_1$ as $\sim_{\rm FB:ps}$ is symmetric, and $P_2 \sim_{\rm FB:ps} Q_2$. Since $P_1 \sim_{\rm FB:ps} P_2$, we also get $Q_1 \sim_{\rm FB:ps} Q_2$ as $\sim_{\rm FB:ps}$ is transitive. By virtue of what has been shown above, from $Q_1 \sim_{\rm FB:ps} Q_2$ with Q_1 and Q_2 in F-nf it follows that $\mathscr{A}_{\rm F} \vdash Q_1 = Q_2$ and hence $\mathscr{A}_{\rm F} \vdash P_1 = P_2$ by transitivity. ■

Proof of Lemma 5.5. From Definition 5.1 it follows that:

- $\widetilde{a\,.\,P} = \ell_{\rm brs}(a)_{a^\dagger.P} \,.\, \ell_{\rm brs}(P)_{a^\dagger.P} = \,<\!a, \{a\}\!>\, .\, \ell^{\varepsilon}_{\rm brs}(P)_P = \,<\!a, \{a\}\!>\, .\, \widetilde{P}$ because P is the immediate subprocess of $a\,.\,P$ and, once the environment $a^\dagger\,.\,P$ reduces to P, the symbol . is no longer necessary in the superscript. The fact that $brs(a^\dagger.P) = \{a\}$ stems from the definition of brs as well as the initiality of P otherwise $a\,.\,P \notin \mathbb{P}$.
- $\widetilde{a^\dagger.P} = \ell_{\rm brs}(a)^\dagger_{a^\dagger.P} \,.\, \ell_{\rm brs}(P)_{a^\dagger.P} = \,<\!a^\dagger, brs(a^\dagger.P)\!>\, .\, \ell^{\varepsilon}_{\rm brs}(P)_P = \,<\!a^\dagger, brs(a^\dagger.P)\!>\, .\, \widetilde{P}$ because P is the immediate subprocess of $a^\dagger\,.\,P$ and, once the environment $a^\dagger.P$ reduces to P, the symbol . is no longer necessary in the superscript. The fact that $brs(a^\dagger.P) = \{a\}$ if P is initial stems from the definition of brs.
- $\widetilde{P_1 + P_2} = \ell^{\dot{+}}_{\rm brs}(P_1)_{P_1+P_2} + \ell^{+\!\cdot}_{\rm brs}(P_2)_{P_1+P_2} = \ell^{\varepsilon}_{\rm brs}(P_1)_{P_1} + \ell^{\varepsilon}_{\rm brs}(P_2)_{P_2} = \widetilde{P_1} + \widetilde{P_2}$ because P_1 and P_2 are the immediate subprocesses of $P_1 + P_2$ and, once the environment $P_1 + P_2$ reduces to P_1 (resp. P_2), the symbol $\dot{+}$ (resp. $+\!\cdot$) is no longer necessary. ■

Proof of Proposition 5.6. After recalling that $\widetilde{P_1}$ and $\widetilde{P_2}$ are sequential, hence it makes sense to talk about their last executed action, we proceed by induction on the syntactical structure of $P \in \mathbb{P}$ to prove both properties simultaneously:

- If P is $\underline{0}$, then $\widetilde{P} = \underline{0}$ by Definition 5.1. They are both initial and $brs(\widetilde{P}) = brs(P) = \emptyset$.
- If P is $a\,.\,P'$, then $\widetilde{P} = \,<\!a, \{a\}\!>\, .\, \widetilde{P'}$ by Lemma 5.5. They are both initial and $brs(\widetilde{P}) = brs(P) = \emptyset$.
- If P is $a^\dagger.\,P'$, then $\widetilde{P} = \,<\!a^\dagger, brs(a^\dagger.P')\!>\, .\, \widetilde{P'}$ by Lemma 5.5, where $initial(\widetilde{P'})$ iff $initial(P')$ and $brs(\widetilde{P'}) = brs(P')$ by the induction hypothesis. P and $\widetilde{P}$ are both non-initial and $brs(\widetilde{P}) = brs(P)$ because the two sets are equal to $\{a\}$ when P' and $\widetilde{P'}$ are both initial or $brs(P')$ when P' and $\widetilde{P'}$ are both non-initial.

- If P is $P_1 + P_2$, then $\widetilde{P} = \widetilde{P_1} + \widetilde{P_2}$ by Lemma 5.5, where $initial(\widetilde{P_k})$ iff $initial(P_k)$ and $brs(\widetilde{P_k}) = brs(P_k)$ for $k \in \{1,2\}$ by the induction hypothesis. Then $initial(\widetilde{P})$ iff $initial(P)$ and $brs(\widetilde{P}) = brs(P)$ because the two sets are equal to $\emptyset$ when P_1, P_2, $\widetilde{P_1}$, $\widetilde{P_2}$ are all initial, $brs(P_1)$ when P_1 and $\widetilde{P_1}$ are non-initial while P_2 and $\widetilde{P_2}$ are initial, or $brs(P_2)$ when P_1 and $\widetilde{P_1}$ are initial while P_2 and $\widetilde{P_2}$ are non-initial.
- If P is $P_1 \,\|_L\, P_2$, then $\widetilde{P} = e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P_1}, \widetilde{P_2}, L)_P$ by Definition 5.1, where $initial(\widetilde{P_k})$ iff $initial(P_k)$ and $brs(\widetilde{P_k}) = brs(P_k)$ for $k \in \{1,2\}$ by the induction hypothesis. There are two cases:
  - If P_1 and P_2 are both initial – hence P is initial – then so are $\widetilde{P_1}$ and $\widetilde{P_2}$ – hence $\widetilde{P}$ is initial by Definition 5.3 – and vice versa. In this case $brs(\widetilde{P}) = brs(P) = \emptyset$.
  - If P_1 and P_2 are not both initial – hence P is non-initial – then so are $\widetilde{P_1}$ and $\widetilde{P_2}$ – hence $\widetilde{P}$ is non-initial by Definition 5.3 – and vice versa. As far as backward ready set preservation is concerned, there are three subcases:
    - ∗ If only P_1 and $\widetilde{P_1}$ are non-initial, say $\widetilde{P_1} = {<}a_1^{\dagger}, brs(a_1^{\dagger}\,.\,P_1'){>}\,.\,\widetilde{P_1'}\,[+\widetilde{P_1''}]$ where $a_1 \notin L$ and the optional P_1'' is initial, then $brs(\widetilde{P_1}) = brs(P_1) = brs(a_1^{\dagger}\,.\,P_1')$ and $brs(\widetilde{P_2}) = brs(P_2) = \emptyset$. Therefore $brs(\widetilde{P}) = brs(\widetilde{P_1}) = brs(P_1) = brs(P)$ as P_2 and $\widetilde{P_2}$ are initial.
    - ∗ The subcase in which only P_2 and $\widetilde{P_2}$ are non-initial is like the previous one.
    - ∗ Let P_1 and P_2 be both non-initial, say $\widetilde{P_k} = {<}a_k^{\dagger}, brs(a_k^{\dagger}\,.\,P_k'){>}\,.\,\widetilde{P_k'}\,[+\widetilde{P_k''}]$ where the optional P_k'' is initial for $k \in \{1,2\}$. Since by hypothesis it is not the case that the last executed action $b_1^{\dagger}$ in $\widetilde{P_1}$ is different from the last executed action $b_2^{\dagger}$ in $\widetilde{P_2}$ with $b_1, b_2 \notin L$ – and the same is true for all possible subprocesses of P_1 and P_2 of the form $Q_1 \,\|_{L'}\, Q_2$ with Q_1 and Q_2 non-initial – it holds that $brs(P_1) = \{b_1\}$ and $brs(P_2) = \{b_2\}$. Recalling that $brs(P_1 \,\|_L\, P_2) = (brs(P_1) \cap \overline{L}) \cup (brs(P_2) \cap \overline{L}) \cup (brs(P_1) \cap brs(P_2) \cap L)$, there are four further subcases:
      - · If $b_1, b_2 \notin L$, then from the aforementioned hypothesis it follows that $b_1 = b_2 \triangleq b$ and hence $brs(\widetilde{P}) = brs(P) = (brs(P_1) \cap \overline{L}) \cup (brs(P_2) \cap \overline{L}) \cup \emptyset = \{b\}$.
      - · If $b_1, b_2 \in L$, then from $P \in \mathbb{P}$ it follows that $b_1 = b_2 \triangleq b$ and hence $brs(\widetilde{P}) = brs(P) = \emptyset \cup \emptyset \cup (brs(P_1) \cap brs(P_2) \cap L) = \{b\}$.
      - · If $b_1 \in L$ and $b_2 \notin L$, then $brs(\widetilde{P}) = brs(P) = \emptyset \cup (brs(P_2) \cap \overline{L}) \cup \emptyset = \{b_2\}$.
      - · If $b_1 \notin L$ and $b_2 \in L$, then $brs(\widetilde{P}) = brs(P) = (brs(P_1) \cap \overline{L}) \cup \emptyset \cup \emptyset = \{b_1\}$. ■

Proof of Theorem 5.7. We proceed by induction on the number $n \in \mathbb{N}_{\geq 1}$ of applications of operational semantic rules that are necessary to derive the considered transitions:

- If $n = 1$, then P is $a\,.\,Q$, with $initial(Q)$, and $\widetilde{P} = {<}a, \{a\}{>}\,.\,\widetilde{Q}$ by Lemma 5.5. According to the rules $\textsc{Act}_{\mathrm{f}}$ in Table 1 and $\textsc{Act}_{\mathrm{brs,f}}$ in Table 3, their only outgoing transitions are respectively $P \xrightarrow{a} a^{\dagger}\,.\,Q$ and $\widetilde{P} \xrightarrow{a,\{a\}}_{\mathrm{brs}} {<}a^{\dagger}, \{a\}{>}\,.\,\widetilde{Q}$, with $\ell_{\mathrm{brs}}(a)_{a^{\dagger}\,.\,Q} = {<}a, \{a\}{>}$ and $\widetilde{a^{\dagger}\,.\,Q} = {<}a^{\dagger}, \{a\}{>}\,.\,\widetilde{Q}$ by Lemma 5.5 because $initial(Q)$.
- If $n > 1$, there are three cases:
  - Let P be $a^{\dagger}\,.\,Q$. Then $P \xrightarrow{.\theta'} a^{\dagger}\,.\,Q'$ implies $Q \xrightarrow{\theta'} Q'$ by rule $\textsc{Act}_{\mathrm{p}}$ in Table 1. By the induction hypothesis this is equivalent to $\widetilde{Q} \xrightarrow{\ell_{\mathrm{brs}}(\theta')_{Q'}}_{\mathrm{brs}} \widetilde{Q'}$, which implies ${<}a^{\dagger}, brs(a^{\dagger}\,.\,Q){>}\,.\,\widetilde{Q}$ $\xrightarrow{\ell_{\mathrm{brs}}(\theta')_{Q'}}_{\mathrm{brs}} {<}a^{\dagger}, brs(a^{\dagger}\,.\,Q'){>}\,.\,\widetilde{Q'}$ by rule $\textsc{Act}_{\mathrm{brs,p}}$ in Table 3 – as $brs(a^{\dagger}\,.\,Q') = brs(Q')$ due to $\neg initial(Q')$ – with ${<}a^{\dagger}, brs(a^{\dagger}\,.\,Q){>}\,.\,\widetilde{Q} = \widetilde{P}$ and ${<}a^{\dagger}, brs(a^{\dagger}\,.\,Q'){>}\,.\,\widetilde{Q'} = \widetilde{a^{\dagger}\,.\,Q'}$ by Lemma 5.5. The proof starting from $\widetilde{P} \xrightarrow{\ell_{\mathrm{brs}}(.\theta')_{a^{\dagger}\,.\,Q'}}_{\mathrm{brs}} \widetilde{a^{\dagger}\,.\,Q'}$ is similar.

- Let P be $P_1 + P_2$. There are two subcases:
  - If P_1 moves, i.e., $P \xrightarrow{+\theta'} P'_1 + P_2$ with $initial(P_2)$, then $P_1 \xrightarrow{\theta'} P'_1$ by rule $\textsc{Cho}_l$ in Table 1. By the induction hypothesis this is equivalent to $\widetilde{P_1} \xrightarrow{\ell_{\mathrm{brs}}(\theta')_{P'_1}}_{\mathrm{brs}} \widetilde{P'_1}$, which implies $\widetilde{P_1} + \widetilde{P_2} \xrightarrow{\ell_{\mathrm{brs}}(+\theta')_{P'_1+P_2}}_{\mathrm{brs}} \widetilde{P'_1} + \widetilde{P_2}$ by rule $\textsc{Cho}_{\mathrm{brs,l}}$ in Table 3 – as $brs(P'_1 + P_2) = brs(P'_1)$ due to $initial(P_2)$ – with $\widetilde{P_1} + \widetilde{P_2} = \widetilde{P}$ and $\widetilde{P'_1} + \widetilde{P_2} = \widetilde{P'_1 + P_2}$ by Lemma 5.5.

    The proof starting from $\widetilde{P} \xrightarrow{\ell_{\mathrm{brs}}(+\theta')_{P'_1+P_2}}_{\mathrm{brs}} \widetilde{P'_1 + P_2}$ is similar.
  - The subcase in which P_2 moves and P_1 is initial is like the previous one.
- Let P be $P_1 \parallel_L P_2$. There are three subcases:
  - If $act(\theta) \notin L$ and P_1 moves, i.e., $P \xrightarrow{\theta} P'_1 \parallel_L P_2$ with $\theta = \rfloor\!\rfloor\theta'$, then $P_1 \xrightarrow{\theta'} P'_1$ by rule $\textsc{Par}_\mathrm{l}$ in Table 1. By the induction hypothesis this is equivalent to $\widetilde{P_1} \xrightarrow{\ell_{\mathrm{brs}}(\theta')_{P'_1}}_{\mathrm{brs}} \widetilde{P'_1}$. By Definition 5.3 this implies that $\widetilde{P}$, after a possible sequence of executed actions, has a maximal initial subprocess with a summand of the form $<\!act(\rfloor\!\rfloor\theta'), brs(P'_1 \parallel_L P_2)\!> . \widetilde{P'_1 \parallel_L P_2}$, hence $\widetilde{P} \xrightarrow{act(\rfloor\!\rfloor\theta'), brs(P'_1 \parallel_L P_2)}_{\mathrm{brs}} \widetilde{P'_1 \parallel_L P_2}$ where $<\!act(\rfloor\!\rfloor\theta'), brs(P'_1 \parallel_L P_2)\!> = \ell_{\mathrm{brs}}(\rfloor\!\rfloor\theta')_{P'_1 \parallel_L P_2}$.

    The proof starting from $\widetilde{P} \xrightarrow{\ell_{\mathrm{brs}}(\theta)_{P'_1 \parallel_L P_2}}_{\mathrm{brs}} \widetilde{P'_1 \parallel_L P_2}$ is similar.
  - The subcase in which $act(\theta) \notin L$ and P_2 moves is like the previous one.
  - If $act(\theta) \in L$, i.e., $P \xrightarrow{\theta} P'_1 \parallel_L P'_2$ with $\theta = \langle \theta_1, \theta_2 \rangle$, then $P_1 \xrightarrow{\theta_1} P'_1$ and $P_2 \xrightarrow{\theta_2} P'_2$ by rule $\textsc{Syn}$ in Table 1. By the induction hypothesis this is equivalent to $\widetilde{P_1} \xrightarrow{\ell_{\mathrm{brs}}(\theta_1)_{P'_1}}_{\mathrm{brs}} \widetilde{P'_1}$ and $\widetilde{P_2} \xrightarrow{\ell_{\mathrm{brs}}(\theta_2)_{P'_2}}_{\mathrm{brs}} \widetilde{P'_2}$. By Definition 5.3 this implies that $\widetilde{P}$, after a possible sequence of executed actions, has a maximal initial subprocess with a summand of the form $<\!act(\langle \theta_1, \theta_2 \rangle), brs(P'_1 \parallel_L P'_2)\!> . \widetilde{P'_1 \parallel_L P'_2}$, hence $\widetilde{P} \xrightarrow{act(\langle \theta_1, \theta_2 \rangle), brs(P'_1 \parallel_L P'_2)}_{\mathrm{brs}} \widetilde{P'_1 \parallel_L P'_2}$ where $<\!act(\langle \theta_1, \theta_2 \rangle), brs(P'_1 \parallel_L P'_2)\!> = \ell_{\mathrm{brs}}(\langle \theta_1, \theta_2 \rangle)_{P'_1 \parallel_L P'_2}$.

    The proof starting from $\widetilde{P} \xrightarrow{\ell_{\mathrm{brs}}(\theta)_{P'_1 \parallel_L P'_2}}_{\mathrm{brs}} \widetilde{P'_1 \parallel_L P'_2}$ is similar. ■

Proof of Corollary 5.8. The proof is divided into two parts:

- Suppose $P_1 \sim_B P_2$ and let $\mathscr{B}$ be a $\sim_B$-bisimulation containing the pair (P_1, P_2). The results follows by proving that $\mathscr{B}' = \{(\widetilde{Q_1}, \widetilde{Q_2}) \mid (Q_1, Q_2) \in \mathscr{B}\}$ is a $\sim_{B:\ell_{\mathrm{brs}}}$ bisimulation. Let $(\widetilde{Q_1}, \widetilde{Q_2}) \in \mathscr{B}'$:
  - If $B = \mathrm{FRB}$ and $\widetilde{Q_1} \xrightarrow{\ell_{\mathrm{brs}}(\theta_1)_{Q'_1}}_{\mathrm{brs}} \widetilde{Q'_1}$, then $Q_1 \xrightarrow{\theta_1} Q'_1$ due to Theorem 5.7. From $(Q_1, Q_2) \in \mathscr{B}$ it follows that $Q_2 \xrightarrow{\theta_2} Q'_2$ with $act(\theta_1) = act(\theta_2)$ and $(Q'_1, Q'_2) \in \mathscr{B}$; moreover $brs(Q'_1) = brs(Q'_2)$ due to Proposition 2.7. Thus $\widetilde{Q_2} \xrightarrow{\ell_{\mathrm{brs}}(\theta_2)_{Q'_2}}_{\mathrm{brs}} \widetilde{Q'_2}$ due to Theorem 5.7, with $act(\theta_1) = act(\theta_2)$, $brs(Q'_1) = brs(Q'_2)$, and $(\widetilde{Q'_1}, \widetilde{Q'_2}) \in \mathscr{B}'$.
  - If $\widetilde{Q'_1} \xrightarrow{\ell_{\mathrm{brs}}(\theta_1)_{Q_1}}_{\mathrm{brs}} \widetilde{Q_1}$ the proof is like the previous one where Proposition 2.7 yields $brs(Q_1) = brs(Q_2)$.
- Suppose $\widetilde{P_1} \sim_{B:\ell_{\mathrm{brs}}} \widetilde{P_2}$ and let $\mathscr{B}$ be a $\sim_{B:\ell_{\mathrm{brs}}}$-bisimulation containing the pair $(\widetilde{P_1}, \widetilde{P_2})$. The results follows by proving that $\mathscr{B}' = \{(Q_1, Q_2) \mid (\widetilde{Q_1}, \widetilde{Q_2}) \in \mathscr{B}\}$ is a $\sim_B$-bisimulation. Let $(Q_1, Q_2) \in \mathscr{B}'$:

- If $B = \text{FRB}$ and $Q_1 \xrightarrow{\theta_1} Q'_1$, then $\widetilde{Q}_1 \xrightarrow{\ell_{\text{brs}}(\theta_1)_{Q'_1}}_{\text{brs}} \widetilde{Q}'_1$ due to Theorem 5.7. From $(\widetilde{Q}_1, \widetilde{Q}_2) \in \mathscr{B}$ it follows that $\widetilde{Q}_2 \xrightarrow{\ell_{\text{brs}}(\theta_2)_{Q'_2}}_{\text{brs}} \widetilde{Q}'_2$ with $act(\theta_1) = act(\theta_2)$, $brs(Q'_1) = brs(Q'_2)$, and $(\widetilde{Q}'_1, \widetilde{Q}'_2) \in \mathscr{B}$. Therefore $Q_2 \xrightarrow{\theta_2} Q'_2$ due to Theorem 5.7, with $act(\theta_1) = act(\theta_2)$ and $(Q'_1, Q'_2) \in \mathscr{B}'$.
- If $Q'_1 \xrightarrow{\theta_1} Q_1$ the proof is like the previous one. ■

Proof of Lemma 5.10. We proceed by induction on the syntactical structure of $P \in \mathbb{P}$:

- If P is $\underline{0}$, then the result follows by taking $\widetilde{Q}$ equal to $\widetilde{\underline{0}}$ due to reflexivity.
- If P is $a \,.\, P'$ where P' is initial, then by the induction hypothesis there exists $\widetilde{Q}'$ initial and in R-nf such that $\mathscr{A}_{\text{R}} \vdash \widetilde{P}' = \widetilde{Q}'$. The result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q}'$ by virtue of $\mathscr{A}_{\text{R}} \vdash \widetilde{P} = \widetilde{a \,.\, Q'}$ due to substitutivity with respect to action prefix, $\mathscr{A}_{\text{R}} \vdash \widetilde{a \,.\, Q'} = \widetilde{Q}'$ due to axiom $\mathscr{A}_{\text{R},3}$, and transitivity.
- If P is $a^\dagger . P'$, then by the induction hypothesis there exists $\widetilde{Q}'$ in R-nf such that $\mathscr{A}_{\text{R}} \vdash \widetilde{P}' = \widetilde{Q}'$. The result follows by taking $\widetilde{Q}$ equal to $\widetilde{a^\dagger . Q'}$ – which is in R-nf because so is $\widetilde{Q}'$ – due to substitutivity with respect to executed action prefix.
- If P is $P_1 + P_2$, then by the induction hypothesis there exist $\widetilde{Q}_1$ and $\widetilde{Q}_2$ in R-nf such that $\mathscr{A}_{\text{R}} \vdash \widetilde{P}_1 = \widetilde{Q}_1$ and $\mathscr{A}_{\text{R}} \vdash \widetilde{P}_2 = \widetilde{Q}_2$. Thus $\mathscr{A}_{\text{R}} \vdash \widetilde{P}_1 + \widetilde{P}_2 = \widetilde{Q}_1 + \widetilde{Q}_2$ due to substitutivity with respect to alternative composition, from which it follows that $\mathscr{A}_{\text{R}} \vdash \widetilde{P} = \widetilde{Q_1 + Q_2}$ due to Lemma 5.5 applied to both sides and transitivity. There are three cases:
  - If P_1 and P_2 are both initial, then $\widetilde{Q}_1$ and $\widetilde{Q}_2$ are both initial too and hence the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q}_1$ by virtue of $\mathscr{A}_{\text{R}} \vdash \widetilde{Q_1 + Q_2} = \widetilde{Q}_1$ due to axiom $\mathscr{A}_{\text{R},4}$ and transitivity.
  - If only P_2 is initial, then only $\widetilde{Q}_2$ is initial too and hence the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q}_1$ for the same reason as the previous case.
  - If only P_1 is initial, then only $\widetilde{Q}_1$ is initial too and hence the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q}_2$ by virtue of $\mathscr{A}_{\text{R}} \vdash \widetilde{Q_1 + Q_2} = \widetilde{Q_2 + Q_1}$ due to axiom $\mathscr{A}_{\text{R},2}$, $\mathscr{A}_{\text{R}} \vdash \widetilde{Q_2 + Q_1} = \widetilde{Q}_2$ due to axiom $\mathscr{A}_{\text{R},4}$, and transitivity.
- If P is $P_1 \,\|_L\, P_2$, then by the induction hypothesis there exist $\widetilde{Q}_1$ and $\widetilde{Q}_2$ in R-nf such that $\mathscr{A}_{\text{R}} \vdash \widetilde{P}_1 = \widetilde{Q}_1$ and $\mathscr{A}_{\text{R}} \vdash \widetilde{P}_2 = \widetilde{Q}_2$. Thus $\mathscr{A}_{\text{R}} \vdash e\ell^{\varepsilon}_{\text{brs}}(\widetilde{P}_1, \widetilde{P}_2, L)_{P_1 \,\|_L\, P_2} = e\ell^{\varepsilon}_{\text{brs}}(\widetilde{Q}_1, \widetilde{Q}_2, L)_{Q_1 \,\|_L\, Q_2}$ due to substitutivity with respect to action prefix and alternative composition, from which it follows that $\mathscr{A}_{\text{R}} \vdash \widetilde{P} = \widetilde{Q_1 \,\|_L\, Q_2}$ due to axiom $\mathscr{A}_{\text{R},5}$ applied to both sides and transitivity. There are four cases:
  - If $\widetilde{Q}_1$ and $\widetilde{Q}_2$ are both $\widetilde{\underline{0}}$, then the result follows by taking $\widetilde{Q}$ equal to $\widetilde{\underline{0}}$ by virtue of $\mathscr{A}_{\text{R}} \vdash \widetilde{Q_1 \,\|_L\, Q_2} = \widetilde{\underline{0}}$ due to axiom $\mathscr{A}_{\text{R},5}$ along with Definition 5.3 and transitivity.
  - If only $\widetilde{Q}_2$ is $\widetilde{\underline{0}}$, then the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q}_1$ – note that none of its executed actions belongs to L otherwise it could not have been executed – by virtue of $\mathscr{A}_{\text{R}} \vdash \widetilde{Q_1 \,\|_L\, Q_2} = \widetilde{Q}_1$ due to axiom $\mathscr{A}_{\text{R},5}$ along with Definition 5.3 and transitivity.
  - The case in which only $\widetilde{Q}_1$ is $\widetilde{\underline{0}}$ is like the previous one.
  - If $\widetilde{Q}_1$ and $\widetilde{Q}_2$ are both different from $\widetilde{\underline{0}}$, say $\widetilde{Q}_k$ of the form $\widetilde{a^\dagger_k . Q'_k}$ with $\widetilde{Q}'_k$ in R-nf for $k \in \{1, 2\}$, then the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q_1 \,\|_L\, Q_2}$, up to the applications of axiom $\mathscr{A}_{\text{R},4}$ necessary to obtain the R-nf in the presence of rebuilt initial alternatives within $\widetilde{Q_1 \,\|_L\, Q_2}$ (see Definition 5.3) and transitivity. ■

Proof of Theorem 5.11. Soundness, i.e., $\mathscr{A}_{\mathrm{R}} \vdash \widetilde{P_1} = \widetilde{P_2} \implies \widetilde{P_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{P_2}$, is a straightforward consequence of the axioms and inference rules behind $\vdash$ (reflexivity, symmetry, transitivity, substitutivity) together with $\sim_{\mathrm{RB}:\ell_{\mathrm{brs}}}$ being an equivalence relation and a congruence, plus the fact that the lefthand side process of each additional axiom in $\mathscr{A}_{\mathrm{R}}$ is $\sim_{\mathrm{RB}:\ell_{\mathrm{brs}}}$-equivalent to the righthand side process of the same axiom.
Let us address ground completeness, i.e., $\widetilde{P_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{P_2} \implies \mathscr{A}_{\mathrm{R}} \vdash \widetilde{P_1} = \widetilde{P_2}$. We suppose that $\widetilde{P_1}$ and $\widetilde{P_2}$ are both in R-nf and proceed by induction on the syntactical structure of $\widetilde{P_1}$:

- If $\widetilde{P_1}$ is $\widetilde{\underline{0}}$, then from $\widetilde{P_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{P_2}$ and $\widetilde{P_2}$ in R-nf we derive that $\widetilde{P_2}$ can only be $\widetilde{\underline{0}}$, from which the result follows by reflexivity.
- If $\widetilde{P_1}$ is $\widetilde{a_1^{\dagger}\,.\,P_1'}$, then from $\widetilde{P_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{P_2}$ and $\widetilde{P_2}$ in R-nf we derive that $\widetilde{P_2}$ can only be $\widetilde{a_2^{\dagger}\,.\,P_2'}$. We recall that $\widetilde{P_1'}$ and $\widetilde{P_2'}$ are both in R-nf.
From $\widetilde{P_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{P_2}$ and $\widetilde{P_1}$ and $\widetilde{P_2}$ both in R-nf and different from $\widetilde{\underline{0}}$ it follows that $\widetilde{P_1}$ and $\widetilde{P_2}$ consist of the same sequence of executed actions, hence in particular $a_1 = a_2$ and $\widetilde{P_1'} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{P_2'}$. From the induction hypothesis we obtain that $\mathscr{A}_{\mathrm{R}} \vdash \widetilde{P_1'} = \widetilde{P_2'}$, hence $\mathscr{A}_{\mathrm{R}} \vdash \widetilde{a_1^{\dagger}\,.\,P_1'} = \widetilde{a_2^{\dagger}\,.\,P_2'}$ due to substitutivity with respect to executed action prefix.

If $\widetilde{P_1}$ and $\widetilde{P_2}$ are not both in R-nf, thanks to Lemma 5.10 we can find $\widetilde{Q_1}$ and $\widetilde{Q_2}$ in R-nf such that $\mathscr{A}_{\mathrm{R}} \vdash \widetilde{P_1} = \widetilde{Q_1}$ and $\mathscr{A}_{\mathrm{R}} \vdash \widetilde{P_2} = \widetilde{Q_2}$, hence $\mathscr{A}_{\mathrm{R}} \vdash \widetilde{Q_2} = \widetilde{P_2}$ by symmetry. Due to soundness, we get $\widetilde{P_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{Q_1}$, hence $\widetilde{Q_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{P_1}$ as $\sim_{\mathrm{RB}:\ell_{\mathrm{brs}}}$ is symmetric, and $\widetilde{P_2} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{Q_2}$. Since $\widetilde{P_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{P_2}$, we also get $\widetilde{Q_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{Q_2}$ as $\sim_{\mathrm{RB}:\ell_{\mathrm{brs}}}$ is transitive. By virtue of what has been shown above, from $\widetilde{Q_1} \sim_{\mathrm{RB}:\ell_{\mathrm{brs}}} \widetilde{Q_2}$ with $\widetilde{Q_1}$ and $\widetilde{Q_2}$ in R-nf it follows that $\mathscr{A}_{\mathrm{R}} \vdash \widetilde{Q_1} = \widetilde{Q_2}$ and hence $\mathscr{A}_{\mathrm{R}} \vdash \widetilde{P_1} = \widetilde{P_2}$ by transitivity. ∎

Proof of Lemma 5.13. We proceed by induction on the syntactical structure of $P \in \mathbb{P}$:

- If P is $\underline{0}$, then the result follows by taking $\widetilde{Q}$ equal to $\widetilde{\underline{0}}$ due to reflexivity.
- If P is $a\,.\,P'$ where P' is initial, then by the induction hypothesis there exists $\widetilde{Q'}$ initial and in FR-nf such that $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P'} = \widetilde{Q'}$. The result follows by taking $\widetilde{Q}$ equal to $\widetilde{a\,.\,Q'}$ – which is in FR-nf because $\widetilde{Q'}$ is initial and in FR-nf – due to substitutivity with respect to action prefix.
- If P is $a^{\dagger}\,.\,P'$, then by the induction hypothesis there exists $\widetilde{Q'}$ in FR-nf such that $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P'} = \widetilde{Q'}$. The result follows by taking $\widetilde{Q}$ equal to $\widetilde{a^{\dagger}\,.\,Q'}$ – which is in FR-nf because so is $\widetilde{Q'}$ – due to substitutivity with respect to executed action prefix.
- If P is $P_1 + P_2$, then by the induction hypothesis there exist $\widetilde{Q_1}$ and $\widetilde{Q_2}$ in FR-nf such that $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P_1} = \widetilde{Q_1}$ and $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P_2} = \widetilde{Q_2}$. Thus $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P_1} + \widetilde{P_2} = \widetilde{Q_1} + \widetilde{Q_2}$ due to substitutivity with respect to alternative composition, from which it follows that $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P} = \widetilde{Q_1 + Q_2}$ due to Lemma 5.5 applied to both sides and transitivity. There are three cases:
  – If P_1 and P_2 are both initial, then $\widetilde{Q_1}$ and $\widetilde{Q_2}$ are both initial too and hence the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q_1 + Q_2}$, up to an application of axiom $\mathscr{A}_{\mathrm{FR},3}$ in the case that $\widetilde{Q_1 + Q_2}$ is not in FR-nf because $\widetilde{Q_1}$ and $\widetilde{Q_2}$ are not different from $\widetilde{\underline{0}}$ (possibly preceded by an application of axiom $\mathscr{A}_{\mathrm{FR},2}$ to move the $\widetilde{\underline{0}}$ subprocess to the right of $+$) and transitivity.
  – If only P_2 is initial, then only $\widetilde{Q_2}$ is initial too and hence the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q_1 + Q_2}$, up to an application of axiom $\mathscr{A}_{\mathrm{FR},3}$ in the case that $\widetilde{Q_1 + Q_2}$ is not in FR-nf because $\widetilde{Q_2}$ is not different from $\widetilde{\underline{0}}$, and transitivity.

  - If only P_1 is initial, then only $\widetilde{Q}_1$ is initial too and hence the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q_2 + Q_1}$ by virtue of $\mathscr{A}_{\rm FR} \vdash \widetilde{Q_1 + Q_2} = \widetilde{Q_2 + Q_1}$ due to axiom $\mathscr{A}_{\rm FR,2}$ and transitivity, up to an application of axiom $\mathscr{A}_{\rm FR,3}$ in the case that $\widetilde{Q_2 + Q_1}$ is not in FR-nf because $\widetilde{Q}_1$ is not different from $\widetilde{\underline{0}}$, and transitivity.

- If P is $P_1 \,\|_L\, P_2$, then by the induction hypothesis there exist $\widetilde{Q}_1$ and $\widetilde{Q}_2$ in FR-nf such that $\mathscr{A}_{\rm FR} \vdash \widetilde{P}_1 = \widetilde{Q}_1$ and $\mathscr{A}_{\rm FR} \vdash \widetilde{P}_2 = \widetilde{Q}_2$. Thus $\mathscr{A}_{\rm FR} \vdash e\ell^{\varepsilon}_{\rm brs}(\widetilde{P}_1, \widetilde{P}_2, L)_{P_1 \,\|_L\, P_2} = e\ell^{\varepsilon}_{\rm brs}(\widetilde{Q}_1, \widetilde{Q}_2, L)_{Q_1 \,\|_L\, Q_2}$ due to substitutivity with respect to action prefix and alternative composition, from which it follows that $\mathscr{A}_{\rm FR} \vdash \widetilde{P} = \widetilde{Q_1 \,\|_L\, Q_2}$ due to axiom $\mathscr{A}_{\rm FR,5}$ applied to both sides and transitivity. There are four cases:
  - If $\widetilde{Q}_1$ and $\widetilde{Q}_2$ are both $\widetilde{\underline{0}}$, then the result follows by taking $\widetilde{Q}$ equal to $\widetilde{\underline{0}}$ by virtue of $\mathscr{A}_{\rm FR} \vdash \widetilde{Q_1 \,\|_L\, Q_2} = \widetilde{\underline{0}}$ due to axiom $\mathscr{A}_{\rm FR,5}$ along with Definition 5.3 and transitivity.
  - If only $\widetilde{Q}_2$ is $\widetilde{\underline{0}}$, then the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q}_1$ – note that none of its executed actions belongs to L otherwise it could not have been executed – by virtue of $\mathscr{A}_{\rm FR} \vdash \widetilde{Q_1 \,\|_L\, Q_2} = \widetilde{Q}_1$ due to axiom $\mathscr{A}_{\rm FR,5}$ along with Definition 5.3 and transitivity.
  - The case in which only $\widetilde{Q}_1$ is $\widetilde{\underline{0}}$ is like the previous one.
  - If $\widetilde{Q}_1$ and $\widetilde{Q}_2$ are both different from $\widetilde{\underline{0}}$, say $\widetilde{Q}_k$ of the form $[\widetilde{a_k^{\dagger} \,.\, Q'_k} +] \sum_{i \in I_k} \widetilde{a_{k,i} \,.\, Q_{k,i}}$ with $\widetilde{Q}'_k$ and every $\widetilde{Q}_{k,i}$ in FR-nf for $k \in \{1,2\}$, then the result follows by taking $\widetilde{Q}$ equal to $\widetilde{Q_1 \,\|_L\, Q_2}$.

■

Proof of Theorem 5.14. Soundness, i.e., $\mathscr{A}_{\rm FR} \vdash \widetilde{P}_1 = \widetilde{P}_2 \Longrightarrow \widetilde{P}_1 \sim_{{\rm FRB}:\ell_{\rm brs}} \widetilde{P}_2$, is a straightforward consequence of the axioms and inference rules behind $\vdash$ (reflexivity, symmetry, transitivity, substitutivity) together with $\sim_{{\rm FRB}:\ell_{\rm brs}}$ being an equivalence relation and a congruence, plus the fact that the lefthand side process of each additional axiom in $\mathscr{A}_{\rm FR}$ is $\sim_{{\rm FRB}:\ell_{\rm brs}}$-equivalent to the righthand side process of the same axiom.
Let us address ground completeness, i.e., $\widetilde{P}_1 \sim_{{\rm FRB}:\ell_{\rm brs}} \widetilde{P}_2 \Longrightarrow \mathscr{A}_{\rm FR} \vdash \widetilde{P}_1 = \widetilde{P}_2$. We suppose that $\widetilde{P}_1$ and $\widetilde{P}_2$ are both in FR-nf and proceed by induction on the syntactical structure of $\widetilde{P}_1$:

- If $\widetilde{P}_1$ is $\widetilde{\underline{0}}$, then from $\widetilde{P}_1 \sim_{{\rm FRB}:\ell_{\rm brs}} \widetilde{P}_2$ and $\widetilde{P}_2$ in FR-nf we derive that $\widetilde{P}_2$ can only be $\widetilde{\underline{0}}$, from which the result follows by reflexivity.

- If $\widetilde{P}_1$ is $[\widetilde{a_1^{\dagger} \,.\, P'_1} +] \sum_{i \in I_1} \widetilde{a_{1,i} \,.\, P_{1,i}}$ with $\widetilde{a_1^{\dagger} \,.\, P'_1}$ present or $I_1 \neq \emptyset$, then from $\widetilde{P}_1 \sim_{{\rm FRB}:\ell_{\rm brs}} \widetilde{P}_2$ and $\widetilde{P}_2$ in FR-nf we derive that $\widetilde{P}_2$ can only be $[\widetilde{a_2^{\dagger} \,.\, P'_2} +] \sum_{i \in I_2} \widetilde{a_{2,i} \,.\, P_{2,i}}$ with $\widetilde{a_2^{\dagger} \,.\, P'_2}$ present iff $\widetilde{a_1^{\dagger} \,.\, P'_1}$ present and – if they are absent – $I_2 \neq \emptyset \neq I_1$. We recall that $\widetilde{P}'_1$, $\widetilde{P}'_2$, every $\widetilde{P}_{1,i}$, and every $\widetilde{P}_{2,i}$ are all in FR-nf.
In the presence of $\widetilde{a_1^{\dagger} \,.\, P'_1}$ and $\widetilde{a_2^{\dagger} \,.\, P'_2}$, it is not necessarily the case that $I_2 \neq \emptyset$ iff $I_1 \neq \emptyset$. However, if for example $I_1 = \emptyset$ and $I_2 \neq \emptyset$, then in order for $\widetilde{P}_1 \sim_{{\rm FRB}:\ell_{\rm brs}} \widetilde{P}_2$ it must be the case that $\mathit{to\_initial}(\widetilde{a_2^{\dagger} \,.\, P'_2}) = \sum_{i \in I_2} \widetilde{a_{2,i} \,.\, P_{2,i}}$, in which case $\mathscr{A}_{\rm FR} \vdash \widetilde{P}_2 = \widetilde{a_2^{\dagger} \,.\, P'_2}$ due to axiom $\mathscr{A}_{\rm FR,4}$. Therefore we can suppose that $I_2 \neq \emptyset$ iff $I_1 \neq \emptyset$ when examining the two main summands of $\widetilde{P}_1$ and $\widetilde{P}_2$.
If $\widetilde{a_1^{\dagger} \,.\, P'_1}$ and $\widetilde{a_2^{\dagger} \,.\, P'_2}$ are present, from the fact that they are the only summands in $\widetilde{P}_1$ and $\widetilde{P}_2$ that can move it follows that $a_1 = a_2$ and $\widetilde{P}'_1 \sim_{{\rm FRB}:\ell_{\rm brs}} \widetilde{P}'_2$, otherwise $\widetilde{P}_1 \sim_{{\rm FRB}:\ell_{\rm brs}} \widetilde{P}_2$ cannot hold. From the induction hypothesis we obtain that $\mathscr{A}_{\rm FR} \vdash \widetilde{P}'_1 = \widetilde{P}'_2$ and hence $\mathscr{A}_{\rm FR} \vdash \widetilde{a_1^{\dagger} \,.\, P'_1} = \widetilde{a_2^{\dagger} \,.\, P'_2}$ due to substitutivity with respect to executed action prefix.

If $I_1 \neq \emptyset \neq I_2$, when going back to $to\_initial(\widetilde{P}_1)$ and $to\_initial(\widetilde{P}_2)$ also $\sum_{i \in I_1} \widetilde{a_{1,i} \,.\, P_{1,i}}$ and $\sum_{i \in I_2} \widetilde{a_{2,i} \,.\, P_{2,i}}$ can move. Suppose that $to\_initial(\widetilde{a_1^{\dagger} \,.\, P_1'}) \neq \sum_{i \in I_1} \widetilde{a_{1,i} \,.\, P_{1,i}}$ and $to\_initial(\widetilde{a_2^{\dagger} \,.\, P_2'}) \neq \sum_{i \in I_2} \widetilde{a_{2,i} \,.\, P_{2,i}}$ so as not to fall back into the previous case. Since $\widetilde{P}_1 \sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}} \widetilde{P}_2$, for each $i_1 \in I_1$ there exists $i_2 \in I_2$ such that $a_{1,i_1} = a_{2,i_2}$ and $\widetilde{P}_{1,i_1} \sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}} \widetilde{P}_{2,i_2}$, and vice versa. From the induction hypothesis we obtain that $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P}_{1,i_1} = \widetilde{P}_{2,i_2}$. It then follows that $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{a_{1,i_1} \,.\, P_{1,i_1}} = \widetilde{a_{2,i_2} \,.\, P_{2,i_2}}$ due to substitutivity with respect to action prefix and hence $\mathscr{A}_{\mathrm{FR}} \vdash \sum_{i \in I_1} \widetilde{a_{1,i} \,.\, P_{1,i}} = \sum_{i \in I_2} \widetilde{a_{2,i} \,.\, P_{2,i}}$ due to substitutivity with respect to alternative composition as well as axiom $\mathscr{A}_{\mathrm{FR},4}$ and transitivity in the presence of identical summands on the same side that are absent on the other side (possibly preceded by applications of axioms $\mathscr{A}_{\mathrm{FR},1}$ and $\mathscr{A}_{\mathrm{FR},2}$ to move identical summands next to each other).

When $\widetilde{a_1^{\dagger} \,.\, P_1'}$ and $\widetilde{a_2^{\dagger} \,.\, P_2'}$ are present and $I_1 \neq \emptyset \neq I_2$, the result stems from substitutivity with respect to alternative composition.

If $\widetilde{P}_1$ and $\widetilde{P}_2$ are not both in FR-nf, thanks to Lemma 5.13 we can find $\widetilde{Q}_1$ and $\widetilde{Q}_2$ in FR-nf, each of which is initial iff so is its corresponding process, such that $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P}_1 = \widetilde{Q}_1$ and $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P}_2 = \widetilde{Q}_2$, hence $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{Q}_2 = \widetilde{P}_2$ by symmetry. Due to soundness, we get $\widetilde{P}_1 \sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}} \widetilde{Q}_1$, hence $\widetilde{Q}_1 \sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}} \widetilde{P}_1$ as $\sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}}$ is symmetric, and $\widetilde{P}_2 \sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}} \widetilde{Q}_2$. Since $\widetilde{P}_1 \sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}} \widetilde{P}_2$, we also get $\widetilde{Q}_1 \sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}} \widetilde{Q}_2$ as $\sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}}$ is transitive. By virtue of what has been shown above, from $\widetilde{Q}_1 \sim_{\mathrm{FRB}:\ell_{\mathrm{brs}}} \widetilde{Q}_2$ with $\widetilde{Q}_1$ and $\widetilde{Q}_2$ in FR-nf it follows that $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{Q}_1 = \widetilde{Q}_2$ and hence $\mathscr{A}_{\mathrm{FR}} \vdash \widetilde{P}_1 = \widetilde{P}_2$ by transitivity. ■

B Further Examples of Encoding of Non-Sequential Processes

Example B.1 Encoding initial processes $P \in \mathbb{P}$ containing subprocesses of the form $P_1 \parallel_L P_2$, where as a consequence both P_1 and P_2 are initial too:

- Let P be $P_1 \parallel_\emptyset P_2$ with P_1 being the initial sequential process $a\,.\,\underline{0} + c\,.\,\underline{0}$ and P_2 being the initial sequential process $b\,.\,\underline{0} + d\,.\,\underline{0}$ so that:

$$\begin{aligned}\widetilde{P_1} &= \ell_{\mathrm{brs}}(.{+}a)_{a^\dagger.\underline{0}+c.\underline{0}} \cdot \widetilde{\underline{0}} + \ell_{\mathrm{brs}}({+}.c)_{a.\underline{0}+c^\dagger.\underline{0}} \cdot \widetilde{\underline{0}}\\ \widetilde{P_2} &= \ell_{\mathrm{brs}}(.{+}b)_{b^\dagger.\underline{0}+d.\underline{0}} \cdot \widetilde{\underline{0}} + \ell_{\mathrm{brs}}({+}.d)_{b.\underline{0}+d^\dagger.\underline{0}} \cdot \widetilde{\underline{0}}\end{aligned}$$

Then:

$$\begin{aligned}\widetilde{P} &= e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P_1}, \widetilde{P_2}, \emptyset)_P\\
&= \ell_{\mathrm{brs}}(\|\!\rfloor .{+}a)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{P_2}, \emptyset)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b.\underline{0}+d.\underline{0})} +\\
&\quad \ell_{\mathrm{brs}}(\|\!\rfloor {+}.c)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{P_2}, \emptyset)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b.\underline{0}+d.\underline{0})} +\\
&\quad \ell_{\mathrm{brs}}(\lfloor\!\| .{+}b)_{(a.\underline{0}+c.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P_1}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0}+c.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})} +\\
&\quad \ell_{\mathrm{brs}}(\lfloor\!\| {+}.d)_{(a.\underline{0}+c.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P_1}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0}+c.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})}\\
&= <a,\{a\}>.\,(\ell_{\mathrm{brs}}(\lfloor\!\| .{+}b)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})} +\\
&\qquad\qquad \ell_{\mathrm{brs}}(\lfloor\!\| {+}.d)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})}) +\\
&\quad <c,\{c\}>.\,(\ell_{\mathrm{brs}}(\lfloor\!\| .{+}b)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})} +\\
&\qquad\qquad \ell_{\mathrm{brs}}(\lfloor\!\| {+}.d)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})}) +\\
&\quad <b,\{b\}>.\,(\ell_{\mathrm{brs}}(\|\!\rfloor .{+}a)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})} +\\
&\qquad\qquad \ell_{\mathrm{brs}}(\|\!\rfloor {+}.c)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b^\dagger.\underline{0}+d.\underline{0})}) +\\
&\quad <d,\{d\}>.\,(\ell_{\mathrm{brs}}(\|\!\rfloor .{+}a)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^\dagger.\underline{0}+c.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})} +\\
&\qquad\qquad \ell_{\mathrm{brs}}(\|\!\rfloor {+}.c)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0}+c^\dagger.\underline{0}) \parallel_\emptyset (b.\underline{0}+d^\dagger.\underline{0})})\\
&= <a,\{a\}>.\,(<b,\{a,b\}>.\,\underline{0} + <d,\{a,d\}>.\,\underline{0}) + <c,\{c\}>.\,(<b,\{c,b\}>.\,\underline{0} + <d,\{c,d\}>.\,\underline{0}) +\\
&\quad <b,\{b\}>.\,(<a,\{b,a\}>.\,\underline{0} + <c,\{b,c\}>.\,\underline{0}) + <d,\{d\}>.\,(<a,\{d,a\}>.\,\underline{0} + <c,\{d,c\}>.\,\underline{0})\end{aligned}$$

- Let P be $(P_a \parallel_\emptyset P_c) + (P_b \parallel_\emptyset P_d)$ with P_a being the initial sequential process $a\,.\,\underline{0}$, P_c being the initial sequential process $c\,.\,\underline{0}$, P_b being the initial sequential process $b\,.\,\underline{0}$, and P_d being the initial sequential process $d\,.\,\underline{0}$. Then:

$$\begin{aligned}\widetilde{P} &= e\ell^{.+}_{\mathrm{brs}}(\widetilde{P_a}, \widetilde{P_c}, \emptyset)_P + e\ell^{+.}_{\mathrm{brs}}(\widetilde{P_b}, \widetilde{P_d}, \emptyset)_P\\
&= \ell_{\mathrm{brs}}(.{+}\|\!\rfloor a)_{(a^\dagger.\underline{0} \parallel_\emptyset c.\underline{0})+(b.\underline{0} \parallel_\emptyset d.\underline{0})} \cdot e\ell^{.+}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{P_c}, \emptyset)_{(a^\dagger.\underline{0} \parallel_\emptyset c.\underline{0})+(b.\underline{0} \parallel_\emptyset d.\underline{0})} +\\
&\quad \ell_{\mathrm{brs}}(.{+}\lfloor\!\| c)_{(a.\underline{0} \parallel_\emptyset c^\dagger.\underline{0})+(b.\underline{0} \parallel_\emptyset d.\underline{0})} \cdot e\ell^{.+}_{\mathrm{brs}}(\widetilde{P_a}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0} \parallel_\emptyset c^\dagger.\underline{0})+(b.\underline{0} \parallel_\emptyset d.\underline{0})} +\\
&\quad \ell_{\mathrm{brs}}({+}.\|\!\rfloor b)_{(a.\underline{0} \parallel_\emptyset c.\underline{0})+(b^\dagger.\underline{0} \parallel_\emptyset d.\underline{0})} \cdot e\ell^{+.}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{P_d}, \emptyset)_{(a.\underline{0} \parallel_\emptyset c.\underline{0})+(b^\dagger.\underline{0} \parallel_\emptyset d.\underline{0})} +\\
&\quad \ell_{\mathrm{brs}}({+}.\lfloor\!\| d)_{(a.\underline{0} \parallel_\emptyset c.\underline{0})+(b.\underline{0} \parallel_\emptyset d^\dagger.\underline{0})} \cdot e\ell^{+.}_{\mathrm{brs}}(\widetilde{P_b}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0} \parallel_\emptyset c.\underline{0})+(b.\underline{0} \parallel_\emptyset d^\dagger.\underline{0})}\\
&= <a,\{a\}>.\,\ell_{\mathrm{brs}}(.{+}\lfloor\!\| c)_{(a^\dagger.\underline{0} \parallel_\emptyset c^\dagger.\underline{0})+(b.\underline{0} \parallel_\emptyset d.\underline{0})} \cdot e\ell^{.+}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^\dagger.\underline{0} \parallel_\emptyset c^\dagger.\underline{0})+(b.\underline{0} \parallel_\emptyset d.\underline{0})} +\\
&\quad <c,\{c\}>.\,\ell_{\mathrm{brs}}(.{+}\|\!\rfloor a)_{(a^\dagger.\underline{0} \parallel_\emptyset c^\dagger.\underline{0})+(b.\underline{0} \parallel_\emptyset d.\underline{0})} \cdot e\ell^{.+}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^\dagger.\underline{0} \parallel_\emptyset c^\dagger.\underline{0})+(b.\underline{0} \parallel_\emptyset d.\underline{0})} +\\
&\quad <b,\{b\}>.\,\ell_{\mathrm{brs}}({+}.\lfloor\!\| d)_{(a.\underline{0} \parallel_\emptyset c.\underline{0})+(b^\dagger.\underline{0} \parallel_\emptyset d^\dagger.\underline{0})} \cdot e\ell^{+.}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0} \parallel_\emptyset c.\underline{0})+(b^\dagger.\underline{0} \parallel_\emptyset d^\dagger.\underline{0})} +\\
&\quad <d,\{d\}>.\,\ell_{\mathrm{brs}}({+}.\|\!\rfloor b)_{(a.\underline{0} \parallel_\emptyset c.\underline{0})+(b^\dagger.\underline{0} \parallel_\emptyset d^\dagger.\underline{0})} \cdot e\ell^{+.}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0} \parallel_\emptyset c.\underline{0})+(b^\dagger.\underline{0} \parallel_\emptyset d^\dagger.\underline{0})}\\
&= <a,\{a\}>.<c,\{a,c\}>.\,\underline{0} + <c,\{c\}>.<a,\{c,a\}>.\,\underline{0} +\\
&\quad <b,\{b\}>.<d,\{b,d\}>.\,\underline{0} + <d,\{d\}>.<b,\{d,b\}>.\,\underline{0}\end{aligned}$$

where, unlike the previous example, $\sigma \neq \varepsilon$ in $e\ell^{\sigma}_{\mathrm{brs}}$ at the beginning, precisely $\sigma = .{+}$ and $\sigma = {+}.$.

- Let P be $P_{a,b} \parallel_{\emptyset} P_c$ with $P_{a,b}$ being the concurrent process $P_a \parallel_{\emptyset} P_b$, P_a being the initial sequential process $a\,.\,\underline{0}$, P_b being the initial sequential process $b\,.\,\underline{0}$, and P_c being the initial sequential process $c\,.\,\underline{0}$. Since as shown in Example 5.4:

$$\begin{aligned}
\widetilde{P}_{a,b} &= e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P}_a, \widetilde{P}_b, \emptyset)_{P_{a,b}} \\
&= \ell_{\mathrm{brs}}(\rfloor a)_{a^{\dagger}.\underline{0} \parallel_{\emptyset} b.\underline{0}} \cdot \ell_{\mathrm{brs}}(\lfloor b)_{a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}} + \\
& \ell_{\mathrm{brs}}(\lfloor b)_{a.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}} \cdot \ell_{\mathrm{brs}}(\rfloor a)_{a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}
\end{aligned}$$

we have that:

$$\begin{aligned}
\widetilde{P} &= e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P}_{a,b}, \widetilde{P}_c, \emptyset)_P \\
&= \ell_{\mathrm{brs}}(\rfloor\rfloor a)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b.\underline{0}) \parallel_{\emptyset} c.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\ell_{\mathrm{brs}}(\lfloor b)_{a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}, \widetilde{P}_c, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b.\underline{0}) \parallel_{\emptyset} c.\underline{0}} + \\
& \ell_{\mathrm{brs}}(\rfloor\lfloor b)_{(a.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\ell_{\mathrm{brs}}(\rfloor a)_{a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}, \widetilde{P}_c, \emptyset)_{(a.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c.\underline{0}} + \\
& \ell_{\mathrm{brs}}(\lfloor c)_{(a.\underline{0} \parallel_{\emptyset} b.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P}_{a,b}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0} \parallel_{\emptyset} b.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \\
&= <a,\{a\}>\,.\,(\ell_{\mathrm{brs}}(\rfloor\lfloor b)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{P}_c, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c.\underline{0}} + \\
& \qquad \ell_{\mathrm{brs}}(\lfloor c)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\ell_{\mathrm{brs}}(\lfloor b)_{a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}}) + \\
& <b,\{b\}>\,.\,(\ell_{\mathrm{brs}}(\rfloor\rfloor a)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{P}_c, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c.\underline{0}} + \\
& \qquad \ell_{\mathrm{brs}}(\lfloor c)_{(a.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\ell_{\mathrm{brs}}(\rfloor a)_{a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}}) + \\
& <c,\{c\}>\,.\,(\ell_{\mathrm{brs}}(\rfloor\rfloor a)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\ell_{\mathrm{brs}}(\lfloor b)_{a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} + \\
& \qquad \ell_{\mathrm{brs}}(\rfloor\lfloor b)_{(a.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\ell_{\mathrm{brs}}(\rfloor a)_{a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}}) \\
&= <a,\{a\}>\,.\,(<b,\{a,b\}>\,.\,\ell_{\mathrm{brs}}(\lfloor c)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} + \\
& \qquad <c,\{a,c\}>\,.\,\ell_{\mathrm{brs}}(\rfloor\lfloor b)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}}) + \\
& <b,\{b\}>\,.\,(<a,\{b,a\}>\,.\,\ell_{\mathrm{brs}}(\lfloor c)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} + \\
& \qquad <c,\{b,c\}>\,.\,\ell_{\mathrm{brs}}(\rfloor\rfloor a)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}}) + \\
& <c,\{c\}>\,.\,(<a,\{c,a\}>\,.\,\ell_{\mathrm{brs}}(\rfloor\lfloor b)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} + \\
& \qquad <b,\{c,b\}>\,.\,\ell_{\mathrm{brs}}(\rfloor\rfloor a)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \widetilde{\underline{0}}, \emptyset)_{(a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}) \parallel_{\emptyset} c^{\dagger}.\underline{0}}) \\
&= <a,\{a\}>\,.\,(<b,\{a,b\}>\,.\,<c,\{a,b,c\}>\,.\,\underline{0} + <c,\{a,c\}>\,.\,<b,\{a,c,b\}>\,.\,\underline{0}) + \\
& <b,\{b\}>\,.\,(<a,\{b,a\}>\,.\,<c,\{b,a,c\}>\,.\,\underline{0} + <c,\{b,c\}>\,.\,<a,\{b,c,a\}>\,.\,\underline{0}) + \\
& <c,\{c\}>\,.\,(<a,\{c,a\}>\,.\,<b,\{c,a,b\}>\,.\,\underline{0} + <b,\{c,b\}>\,.\,<a,\{c,b,a\}>\,.\,\underline{0}) \qquad \blacksquare
\end{aligned}$$

Example B.2 Encoding non-initial processes $P \in \mathbb{P}$ containing subprocesses of the form $P_1 \parallel_L P_2$ where either P_1 or P_2 is initial:

- Let P be $P_1 \parallel_{\emptyset} P_2$ with P_1 being the initial sequential process $a\,.\,\underline{0} + c\,.\,\underline{0}$ and P_2 being the non-initial sequential process $b^{\dagger}\,.\,\underline{0} + d\,.\,\underline{0}$ so that:

$$\begin{aligned}
\widetilde{P}_1 &= \ell_{\mathrm{brs}}({\cdot\!+}a)_{a^{\dagger}.\underline{0}+c.\underline{0}} \cdot \widetilde{\underline{0}} + \ell_{\mathrm{brs}}({+\!\cdot}c)_{a.\underline{0}+c^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}} \\
\widetilde{P}_2 &= \ell_{\mathrm{brs}}({\cdot\!+}b)^{\dagger}_{b^{\dagger}.\underline{0}+d.\underline{0}} \cdot \widetilde{\underline{0}} + \ell_{\mathrm{brs}}({+\!\cdot}d)_{b.\underline{0}+d^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}
\end{aligned}$$

Then:

$$\begin{aligned}
\widetilde{P} &= e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P}_1, \widetilde{P}_2, \emptyset)_P \\
&= \ell_{\mathrm{brs}}(\lfloor{\cdot\!+}b)^{\dagger}_{(a.\underline{0}+c.\underline{0}) \parallel_{\emptyset} (b^{\dagger}.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P}_1, \widetilde{\underline{0}}, \emptyset)_P + \\
& \ell_{\mathrm{brs}}(\lfloor{+\!\cdot}d)_{(a.\underline{0}+c.\underline{0}) \parallel_{\emptyset} (b.\underline{0}+d^{\dagger}.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P}_1, \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0}+c.\underline{0}) \parallel_{\emptyset} (b.\underline{0}+d^{\dagger}.\underline{0})} + \\
& \ell_{\mathrm{brs}}(\rfloor{\cdot\!+}a)_{(a^{\dagger}.\underline{0}+c.\underline{0}) \parallel_{\emptyset} (b.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \ell_{\mathrm{brs}}({\cdot\!+}b)_{b^{\dagger}.\underline{0}+d.\underline{0}} \cdot \widetilde{\underline{0}} + \ell_{\mathrm{brs}}({+\!\cdot}d)_{b.\underline{0}+d^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}, \emptyset)_{(a^{\dagger}.\underline{0}+c.\underline{0}) \parallel_{\emptyset} (b.\underline{0}+d.\underline{0})} + \\
& \ell_{\mathrm{brs}}(\rfloor{+\!\cdot}c)_{(a.\underline{0}+c^{\dagger}.\underline{0}) \parallel_{\emptyset} (b.\underline{0}+d.\underline{0})} \cdot e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}}, \ell_{\mathrm{brs}}({\cdot\!+}b)_{b^{\dagger}.\underline{0}+d.\underline{0}} \cdot \widetilde{\underline{0}} + \ell_{\mathrm{brs}}({+\!\cdot}d)_{b.\underline{0}+d^{\dagger}.\underline{0}} \cdot \widetilde{\underline{0}}, \emptyset)_{(a.\underline{0}+c^{\dagger}.\underline{0}) \parallel_{\emptyset} (b.\underline{0}+d.\underline{0})}
\end{aligned}$$

$$
\begin{aligned}
= \; & <b^{\dagger},\{b\}>.\,(\ell_{\mathrm{brs}}(\rrfloor\dot{+}a)_{(a^{\dagger}.\underline{0}+c.\underline{0})\,\|_{\emptyset}\,(b^{\dagger}.\underline{0}+d.\underline{0})}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\widetilde{\underline{0}},\emptyset)_{(a^{\dagger}.\underline{0}+c.\underline{0})\,\|_{\emptyset}\,(b^{\dagger}.\underline{0}+d.\underline{0})}\; + \\
& \qquad \ell_{\mathrm{brs}}(\rrfloor\dot{+}c)_{(a.\underline{0}+c^{\dagger}.\underline{0})\,\|_{\emptyset}\,(b^{\dagger}.\underline{0}+d.\underline{0})}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\widetilde{\underline{0}},\emptyset)_{(a.\underline{0}+c^{\dagger}.\underline{0})\,\|_{\emptyset}\,(b^{\dagger}.\underline{0}+d.\underline{0})})\; + \\
& <d,\{d\}>.\,(\ell_{\mathrm{brs}}(\rrfloor\dot{+}a)_{(a^{\dagger}.\underline{0}+c.\underline{0})\,\|_{\emptyset}\,(b.\underline{0}+d^{\dagger}.\underline{0})}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\widetilde{\underline{0}},\emptyset)_{(a^{\dagger}.\underline{0}+c.\underline{0})\,\|_{\emptyset}\,(b.\underline{0}+d^{\dagger}.\underline{0})}\; + \\
& \qquad \ell_{\mathrm{brs}}(\rrfloor\dot{+}c)_{(a.\underline{0}+c^{\dagger}.\underline{0})\,\|_{\emptyset}\,(b.\underline{0}+d^{\dagger}.\underline{0})}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\widetilde{\underline{0}},\emptyset)_{(a.\underline{0}+c^{\dagger}.\underline{0})\,\|_{\emptyset}\,(b.\underline{0}+d^{\dagger}.\underline{0})})\; + \\
& <a,\{a\}>.\,(\ell_{\mathrm{brs}}(\llfloor\dot{+}b)_{(a^{\dagger}.\underline{0}+c.\underline{0})\,\|_{\emptyset}\,(b^{\dagger}.\underline{0}+d.\underline{0})}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\widetilde{\underline{0}},\emptyset)_{(a^{\dagger}.\underline{0}+c.\underline{0})\,\|_{\emptyset}\,(b^{\dagger}.\underline{0}+d.\underline{0})}\; + \\
& \qquad \ell_{\mathrm{brs}}(\llfloor\dot{+}d)_{(a^{\dagger}.\underline{0}+c.\underline{0})\,\|_{\emptyset}\,(b.\underline{0}+d^{\dagger}.\underline{0})}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\widetilde{\underline{0}},\emptyset)_{(a^{\dagger}.\underline{0}+c.\underline{0})\,\|_{\emptyset}\,(b.\underline{0}+d^{\dagger}.\underline{0})})\; + \\
& <c,\{c\}>.\,(\ell_{\mathrm{brs}}(\llfloor\dot{+}b)_{(a.\underline{0}+c^{\dagger}.\underline{0})\,\|_{\emptyset}\,(b^{\dagger}.\underline{0}+d.\underline{0})}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\widetilde{\underline{0}},\emptyset)_{(a.\underline{0}+c^{\dagger}.\underline{0})\,\|_{\emptyset}\,(b^{\dagger}.\underline{0}+d.\underline{0})}\; + \\
& \qquad \ell_{\mathrm{brs}}(\llfloor\dot{+}d)_{(a.\underline{0}+c^{\dagger}.\underline{0})\,\|_{\emptyset}\,(b.\underline{0}+d^{\dagger}.\underline{0})}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\widetilde{\underline{0}},\emptyset)_{(a.\underline{0}+c^{\dagger}.\underline{0})\,\|_{\emptyset}\,(b.\underline{0}+d^{\dagger}.\underline{0})}) \\
= \; & <b^{\dagger},\{b\}>.\,(<a,\{b,a\}>.\underline{0}+<c,\{b,c\}>.\underline{0})\; + \\
& <d,\{d\}>.\,(<a,\{d,a\}>.\underline{0}+<c,\{d,c\}>.\underline{0})\; + \\
& <a,\{a\}>.\,(<b,\{a,b\}>.\underline{0}+<d,\{a,d\}>.\underline{0})\; + \\
& <c,\{c\}>.\,(<b,\{c,b\}>.\underline{0}+<d,\{c,d\}>.\underline{0})
\end{aligned}
$$

Note that $\llfloor\dot{+}b \leq_{\dagger} \rrfloor\dot{+}a$ and $\llfloor\dot{+}b \leq_{\dagger} \rrfloor\dot{+}c$ as only b has been executed. ∎

Example B.3 Encoding non-initial processes $P \in \mathbb{P}$ containing subprocesses of the form $P_1 \,\|_L\, P_2$ where both P_1 and P_2 are non-initial:

- Let P be $P_1 \,\|_{\{c\}}\, P_2$ with P_1 being the non-initial sequential process $a^{\dagger}.c^{\dagger}.\underline{0}$ and P_2 being the non-initial sequential process $c^{\dagger}.b.\underline{0}$ so that:

  $$
  \begin{aligned}
  \widetilde{P_1} &= \ell_{\mathrm{brs}}(a)^{\dagger}_{a^{\dagger}.c.\underline{0}}\,.\,\ell_{\mathrm{brs}}(.c)^{\dagger}_{a^{\dagger}.c^{\dagger}.\underline{0}}\,.\,\widetilde{\underline{0}} \\
  \widetilde{P_2} &= \ell_{\mathrm{brs}}(c)^{\dagger}_{c^{\dagger}.b.\underline{0}}\,.\,\ell_{\mathrm{brs}}(.b)_{c^{\dagger}.b^{\dagger}.\underline{0}}\,.\,\widetilde{\underline{0}}
  \end{aligned}
  $$

  Then:

  $$
  \begin{aligned}
  \widetilde{P} &= e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P_1},\widetilde{P_2},\{c\})_P \\
  &= \ell_{\mathrm{brs}}(\rrfloor a)^{\dagger}_{a^{\dagger}.c.\underline{0}\,\|_{\{c\}}\,c.b.\underline{0}}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\ell_{\mathrm{brs}}(.c)^{\dagger}_{a^{\dagger}.c^{\dagger}.\underline{0}}\,.\,\widetilde{\underline{0}},\widetilde{P_2},\{c\})_P \\
  &= <a^{\dagger},\{a\}>.\,\ell_{\mathrm{brs}}(\langle .c,c\rangle)^{\dagger}_{a^{\dagger}.c^{\dagger}.\underline{0}\,\|_{\{c\}}\,c^{\dagger}.b.\underline{0}}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\ell_{\mathrm{brs}}(.b)_{c^{\dagger}.b^{\dagger}.\underline{0}}\,.\,\widetilde{\underline{0}},\{c\})_P \\
  &= <a^{\dagger},\{a\}>.<c^{\dagger},\{c\}>.\,\ell_{\mathrm{brs}}(\llfloor .b)_{a^{\dagger}.c^{\dagger}.\underline{0}\,\|_{\{c\}}\,c^{\dagger}.b^{\dagger}.\underline{0}}\,.\,e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{\underline{0}},\widetilde{\underline{0}},\{c\})_{a^{\dagger}.c^{\dagger}.\underline{0}\,\|_{\{c\}}\,c^{\dagger}.b^{\dagger}.\underline{0}} \\
  &= <a^{\dagger},\{a\}>.<c^{\dagger},\{c\}>.<b,\{b\}>.\underline{0}
  \end{aligned}
  $$

- Let P be $P_1 \,\|_{\{c\}}\, P_2$ with P_1 being the non-initial sequential process $a^{\dagger}.c^{\dagger}.a^{\dagger}.\underline{0}$ and P_2 being the non-initial sequential process $b^{\dagger}.c^{\dagger}.b^{\dagger}.\underline{0}$ so that:

  $$
  \begin{aligned}
  \widetilde{P_1} &= \ell_{\mathrm{brs}}(a)^{\dagger}_{a^{\dagger}.c.a.\underline{0}}\,.\,\ell_{\mathrm{brs}}(.c)^{\dagger}_{a^{\dagger}.c^{\dagger}.a.\underline{0}}\,.\,\ell_{\mathrm{brs}}(..a)^{\dagger}_{a^{\dagger}.c^{\dagger}.a^{\dagger}.\underline{0}}\,.\,\widetilde{\underline{0}} \\
  \widetilde{P_2} &= \ell_{\mathrm{brs}}(b)^{\dagger}_{b^{\dagger}.c.b.\underline{0}}\,.\,\ell_{\mathrm{brs}}(.c)^{\dagger}_{b^{\dagger}.c^{\dagger}.b.\underline{0}}\,.\,\ell_{\mathrm{brs}}(..b)^{\dagger}_{b^{\dagger}.c^{\dagger}.b^{\dagger}.\underline{0}}\,.\,\widetilde{\underline{0}}
  \end{aligned}
  $$

  Then for $\rrfloor a \leq_{\dagger} \llfloor b$ and $\llfloor ..b \leq_{\dagger} \rrfloor ..a$:

  $$
  \begin{aligned}
  \widetilde{P} &= e\ell^{\varepsilon}_{\mathrm{brs}}(\widetilde{P_1},\widetilde{P_2},\{c\})_P = \ldots = \\
  &= <a^{\dagger},\{a\}>.<b^{\dagger},\{a,b\}>.\underline{0}.<c^{\dagger},\{c\}>.\,(<b^{\dagger},\{b\}>.<a^{\dagger},\{b,a\}>.\underline{0}\; + \\
  &\qquad\qquad <a,\{a\}>.<b,\{a,b\}>.\underline{0})\; + \\
  &\quad <b,\{b\}>.<a,\{a,b\}>.\underline{0}.<c,\{c\}>.\,(<a,\{a\}>.<b,\{a,b\}>.\underline{0}\; + \\
  &\qquad\qquad <b,\{b\}>.<a,\{b,a\}>.\underline{0})
  \end{aligned}
  $$

  ∎