

PRPC: SEMANTICS, LOGICS, AXIOMS

A PROCESS ALGEBRAIC THEORY OF REVERSIBLE COMPUTING

Marco Bernardo

University of Urbino – Italy

PRIN 2020 project *NiRvAna*

Concurrency: Nondeterminism vs. Irreversibility

- Systems composed of **several interconnected computing parts** that communicate by exchanging information or simply synchronizing.
- Models: shared memory, message passing, web services, cloud, ...
- Types: centralized/distributed/decentralized, static/dynamic/mobile.
- Aspects: functionality, security, reliability, performance, ...

Concurrency: Nondeterminism vs. Irreversibility

- Systems composed of **several interconnected computing parts** that communicate by exchanging information or simply synchronizing.
- Models: shared memory, message passing, web services, cloud, ...
- Types: centralized/distributed/decentralized, static/dynamic/mobile.
- Aspects: functionality, security, reliability, performance, ...
- **Nondeterminism: the input does not uniquely define the output.**
- Different advancing speeds, scheduling policies, ...

Concurrency: Nondeterminism vs. Irreversibility

- Systems composed of **several interconnected computing parts** that communicate by exchanging information or simply synchronizing.
- Models: shared memory, message passing, web services, cloud, ...
- Types: centralized/distributed/decentralized, static/dynamic/mobile.
- Aspects: functionality, security, reliability, performance, ...
- **Nondeterminism: the input does not uniquely define the output.**
- Different advancing speeds, scheduling policies, ...
- **What if the output does not uniquely define the input?**
- **Irreversibility:** typical of functions that are *not invertible*.

Concurrency: Nondeterminism vs. Irreversibility

- Systems composed of **several interconnected computing parts** that communicate by exchanging information or simply synchronizing.
- Models: shared memory, message passing, web services, cloud, ...
- Types: centralized/distributed/decentralized, static/dynamic/mobile.
- Aspects: functionality, security, reliability, performance, ...
- **Nondeterminism: the input does not uniquely define the output.**
- Different advancing speeds, scheduling policies, ...
- **What if the output does not uniquely define the input?**
- **Irreversibility:** typical of functions that are *not invertible*.
- Example 1: conjunctions/disjunctions are irreversible.
- Example 2: negation is reversible.

Reversible Computing

- What does (ir)reversibility mean in computing?
- Well established concept in mathematics, physics, chemistry, biology: inverse relation/function/operation/formula/law/reaction ...
- Much more recent in informatics: seminal papers by Landauer in 1961 and Bennett in 1973 on IBM Journal of Research and Development.

Reversible Computing

- What does (ir)reversibility mean in computing?
- Well established concept in mathematics, physics, chemistry, biology: inverse relation/function/operation/formula/law/reaction ...
- Much more recent in informatics: seminal papers by Landauer in 1961 and Bennett in 1973 on IBM Journal of Research and Development.
- **Landauer principle** states that any manipulation of information that is *irreversible* – i.e., causes information loss – such as:
 - erasure/overwriting of bits
 - merging of computation pathsmust be accompanied by a corresponding *entropy increase*.

Reversible Computing

- What does (ir)reversibility mean in computing?
- Well established concept in mathematics, physics, chemistry, biology: inverse relation/function/operation/formula/law/reaction ...
- Much more recent in informatics: seminal papers by Landauer in 1961 and Bennett in 1973 on IBM Journal of Research and Development.
- **Landauer principle** states that any manipulation of information that is *irreversible* – i.e., causes information loss – such as:
 - erasure/overwriting of bits
 - merging of computation pathsmust be accompanied by a corresponding *entropy increase*.
- Minimal *heat generation* due to *extra work* for standardizing signals and making them independent of their history, so that it becomes *impossible to determine the input from the output*.

- Due to Landauer principle, the **logical irreversibility** of a function implies the **physical irreversibility** of computing that function and the consequent dissipative effects.
- Experimentally verified by Bérut et al in 2012 and revisited in terms of its physical foundations by Frank in 2018.

- Due to Landauer principle, the **logical irreversibility** of a function implies the **physical irreversibility** of computing that function and the consequent dissipative effects.
- Experimentally verified by Bérut et al in 2012 and revisited in terms of its physical foundations by Frank in 2018.
- Every reversible computation, where no information is lost instead, may be potentially carried out **without dissipating further heat**.
- Lower energy consumption could therefore be achieved by resorting to **reversible computing**.

- Due to Landauer principle, the **logical irreversibility** of a function implies the **physical irreversibility** of computing that function and the consequent dissipative effects.
- Experimentally verified by Bérut et al in 2012 and revisited in terms of its physical foundations by Frank in 2018.
- Every reversible computation, where no information is lost instead, may be potentially carried out **without dissipating further heat**.
- Lower energy consumption could therefore be achieved by resorting to **reversible computing**.
- There are many other applications of reversible computing:
 - Biochemical reaction modeling (nature).
 - Parallel discrete-event simulation (speedup).
 - Fault-tolerant computing systems (rollback).
 - Robotics and control theory (backtrack).
 - Concurrent program debugging (reproducibility).
 - Distributed algorithms (deadlock, consensus).

- Two directions of computation characterize every reversible system:
 - **Forward**: coincides with the normal way of computing.
 - **Backward**: the effects of the forward one are undone (when needed).

- Two directions of computation characterize every reversible system:
 - **Forward**: coincides with the normal way of computing.
 - **Backward**: the effects of the forward one are undone (when needed).
- **How to proceed backward? Same path as the forward direction?**
Paths with **causally independent actions** should be deemed equivalent.
- **Is the last executed action uniquely identifiable?**
Not necessarily, especially in the case of a concurrent system.

- Two directions of computation characterize every reversible system:
 - **Forward**: coincides with the normal way of computing.
 - **Backward**: the effects of the forward one are undone (when needed).
- **How to proceed backward? Same path as the forward direction?**
Paths with **causally independent actions** should be deemed equivalent.
- **Is the last executed action uniquely identifiable?**
Not necessarily, especially in the case of a concurrent system.
- Different notions of reversibility developed in different settings:
 - **Causal reversibility** is the capability of going back to a past state *consistently with the computational history*: an action can be undone iff all of its consequences have been undone already [DanosKrivine04].

- Two directions of computation characterize every reversible system:
 - **Forward**: coincides with the normal way of computing.
 - **Backward**: the effects of the forward one are undone (when needed).
- **How to proceed backward? Same path as the forward direction?**
Paths with **causally independent actions** should be deemed equivalent.
- **Is the last executed action uniquely identifiable?**
Not necessarily, especially in the case of a concurrent system.
- Different notions of reversibility developed in different settings:
 - **Causal reversibility** is the capability of going back to a past state *consistently with the computational history*: an action can be undone iff all of its consequences have been undone already [DanosKrivine04].
 - **Time reversibility** refers to the conditions under which the stochastic behavior remains the same when the *direction of time* is reversed (quantitative models, efficient performance evaluation) [Kelly79].

- Two directions of computation characterize every reversible system:
 - **Forward**: coincides with the normal way of computing.
 - **Backward**: the effects of the forward one are undone (when needed).
- **How to proceed backward? Same path as the forward direction?**
Paths with **causally independent actions** should be deemed equivalent.
- **Is the last executed action uniquely identifiable?**
Not necessarily, especially in the case of a concurrent system.
- Different notions of reversibility developed in different settings:
 - **Causal reversibility** is the capability of going back to a past state *consistently with the computational history*: an action can be undone iff all of its consequences have been undone already [DanosKrivine04].
 - **Time reversibility** refers to the conditions under which the stochastic behavior remains the same when the *direction of time* is reversed (quantitative models, efficient performance evaluation) [Kelly79].
 - Only recently the relationships between the two have been investigated (the former implies the latter over models based on Markov chains when certain constraints are met).

Reversibility in Process Algebra

- There are no inverse process algebraic operators!

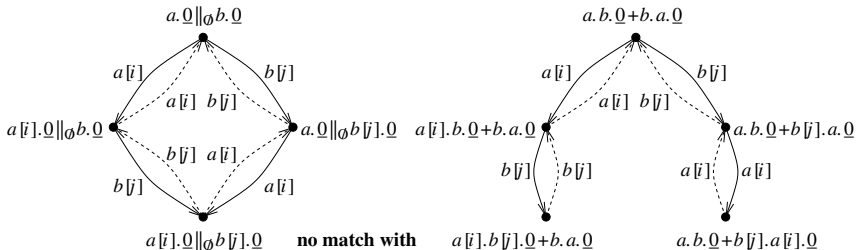
Reversibility in Process Algebra

- There are no inverse process algebraic operators!
- The **dynamic approach** of [DanosKrivine04] yielding **RCCS** uses explicit **stack-based memories** attached to processes to record all executed actions and all discarded subprocesses.
- A single transition relation is defined, while actions are divided into forward and backward resulting in forward and backward transitions.

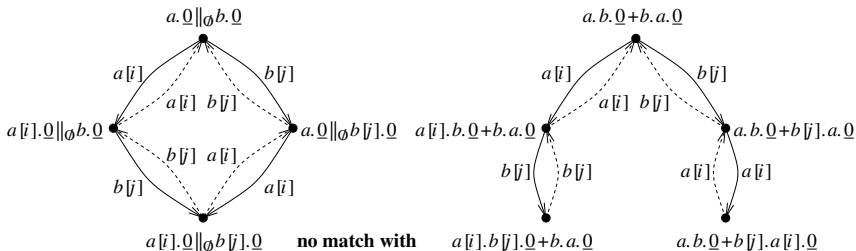
Reversibility in Process Algebra

- There are no inverse process algebraic operators!
- The **dynamic approach** of [DanosKrivine04] yielding **RCCS** uses explicit **stack-based memories** attached to processes to record all executed actions and all discarded subprocesses.
- A single transition relation is defined, while actions are divided into forward and backward resulting in forward and backward transitions.
- The **static approach** of [PhillipsUlidowski07] yielding **CCSK** is a method to reverse calculi by **retaining within process syntax**:
 - all executed actions, which are suitably decorated;
 - all dynamic operators, which are therefore treated as static.
- A forward transition relation and a backward transition relation are separately defined, labeled with **communication keys** so as to know who synchronized with whom when building backward transitions.

- In [PU07] **forward-reverse bisimilarity** has been introduced too.
- **Truly concurrent** as it does not satisfy the **expansion law** of parallel composition into a choice among all possible action sequencings ($a \neq b$):



- In [PU07] **forward-reverse bisimilarity** has been introduced too.
- **Truly concurrent** as it does not satisfy the **expansion law** of parallel composition into a choice among all possible action sequencings ($a \neq b$):



- In [DeNicolaMontanariVaandrager90] **back-and-forth bisimilarity**.
- **Interleaving view** restoration because this bisimilarity is defined on computations instead of states to **preserve both causality and history** (one transition relation, viewed as bidirectional, outgoing/incoming).

- What are the properties of bisimilarity over reversible processes?

- What are the properties of bisimilarity over reversible processes?
- Minimal process calculus tailored for reversible processes to *comparatively* study congruence, logics, and axioms for:
 - Forward-reverse bisimilarity.
 - Forward bisimilarity.
 - Reverse bisimilarity.

- What are the properties of bisimilarity over reversible processes?
- Minimal process calculus tailored for reversible processes to *comparatively* study congruence, logics, and axioms for:
 - Forward-reverse bisimilarity.
 - Forward bisimilarity.
 - Reverse bisimilarity.
- Two different kinds of bisimilarities:
 - Strong bisimilarities (all actions are treated in the same way).
 - Weak bisimilarities (abstracting from unobservable actions).

- What are the properties of bisimilarity over reversible processes?
- Minimal process calculus tailored for reversible processes to *comparatively* study congruence, logics, and axioms for:
 - Forward-reverse bisimilarity.
 - Forward bisimilarity.
 - Reverse bisimilarity.
- Two different kinds of bisimilarities:
 - Strong bisimilarities (all actions are treated in the same way).
 - Weak bisimilarities (abstracting from unobservable actions).
- Considering only sequential processes (i.e., no parallel composition) to be neutral with respect to interleaving view vs. true concurrency.
- Adding parallel composition to uniformly investigate expansion laws (relate sequential *specifications* to concurrent *implementations*).

- What are the properties of bisimilarity over reversible processes?
- Minimal process calculus tailored for reversible processes to *comparatively* study congruence, logics, and axioms for:
 - Forward-reverse bisimilarity.
 - Forward bisimilarity.
 - Reverse bisimilarity.
- Two different kinds of bisimilarities:
 - Strong bisimilarities (all actions are treated in the same way).
 - Weak bisimilarities (abstracting from unobservable actions).
- Considering only sequential processes (i.e., no parallel composition) to be neutral with respect to interleaving view vs. true concurrency.
- Adding parallel composition to uniformly investigate expansion laws (relate sequential *specifications* to concurrent *implementations*).
- Characterizations via other behavioral equivalences.

- What are the properties of bisimilarity over reversible processes?
- Minimal process calculus tailored for reversible processes to *comparatively* study congruence, logics, and axioms for:
 - Forward-reverse bisimilarity.
 - Forward bisimilarity.
 - Reverse bisimilarity.
- Two different kinds of bisimilarities:
 - Strong bisimilarities (all actions are treated in the same way).
 - Weak bisimilarities (abstracting from unobservable actions).
- Considering only sequential processes (i.e., no parallel composition) to be neutral with respect to interleaving view vs. true concurrency.
- Adding parallel composition to uniformly investigate expansion laws (relate sequential *specifications* to concurrent *implementations*).
- Characterizations via other behavioral equivalences.
- Can we avoid external memories and communication keys?

PRPC – Proved Reversible Process Calculus

- Countable set \mathcal{A} of actions including the unobservable action τ , renaming $\rho : \mathcal{A} \rightarrow \mathcal{A}$ s.t. $\rho(\tau) = \tau$, synchronization set $L \subseteq \mathcal{A} \setminus \{\tau\}$.
- Usually only the **future behavior** of processes is described.

PRPC – Proved Reversible Process Calculus

- Countable set \mathcal{A} of actions including the unobservable action τ , renaming $\rho : \mathcal{A} \rightarrow \mathcal{A}$ s.t. $\rho(\tau) = \tau$, synchronization set $L \subseteq \mathcal{A} \setminus \{\tau\}$.
- Usually only the **future behavior** of processes is described.
- We store the **past behavior** in the syntax like in [PU07]:

$$P ::= \underline{0} \mid a.P \mid a^\dagger.P \mid P \sqcup \rho^\top \mid P + P \mid P \parallel_L P$$

- $a^\dagger.P$ executed action a , its forward continuation is inside P , and can undo a after all executed actions within P have been undone.

PRPC – Proved Reversible Process Calculus

- Countable set \mathcal{A} of actions including the unobservable action τ , renaming $\rho : \mathcal{A} \rightarrow \mathcal{A}$ s.t. $\rho(\tau) = \tau$, synchronization set $L \subseteq \mathcal{A} \setminus \{\tau\}$.
- Usually only the **future behavior** of processes is described.
- We store the **past behavior** in the syntax like in [PU07]:

$$P ::= \underline{0} \mid a.P \mid a^\dagger.P \mid P \sqcup \rho^\top \mid P + P \mid P \parallel_L P$$

- $a^\dagger.P$ executed action a , its forward continuation is inside P , and can undo a after all executed actions within P have been undone.
- Single transition relation like in [DMV90] labeled just with actions.
- Therefore there is no need of communication keys [PU07], thus allowing for uniform action decorations like in [BoudolCastellani94].
- No need to distinguish between forward and backward actions or resort to stack-based memories [DK04].

- **Initial processes:** standard as all of their actions are unexecuted.

- **Initial processes:** standard as all of their actions are unexecuted.
- Set **P** of **well-formed processes** with unexecuted and executed actions:

$$\begin{aligned}
 & wf(\underline{0}) \\
 & wf(a . P') \quad \text{iff} \quad initial(P') \\
 & wf(a^\dagger . P') \quad \text{iff} \quad wf(P') \\
 & wf(P' \sqcup \rho^\neg) \quad \text{iff} \quad wf(P') \\
 & wf(P_1 + P_2) \quad \text{iff} \quad (wf(P_1) \wedge initial(P_2)) \vee \\
 & \quad \quad \quad (initial(P_1) \wedge wf(P_2)) \\
 & wf(P_1 \parallel_L P_2) \quad \text{iff} \quad wf(P_1) \wedge wf(P_2)
 \end{aligned}$$

- **Initial processes:** standard as all of their actions are unexecuted.
- Set P of **well-formed processes** with unexecuted and executed actions:

$$\begin{aligned}
 & wf(\underline{0}) \\
 & wf(a . P') \quad \text{iff} \quad initial(P') \\
 & wf(a^\dagger . P') \quad \text{iff} \quad wf(P') \\
 & wf(P' \sqcup \rho^\neg) \quad \text{iff} \quad wf(P') \\
 & wf(P_1 + P_2) \quad \text{iff} \quad (wf(P_1) \wedge initial(P_2)) \vee \\
 & \quad \quad \quad (initial(P_1) \wedge wf(P_2)) \\
 & wf(P_1 \parallel_L P_2) \quad \text{iff} \quad wf(P_1) \wedge wf(P_2)
 \end{aligned}$$

- $\underline{0}$ is both initial and well-formed.
- Any initial process is well-formed too.
- P also contains processes that are not initial: $a^\dagger . b . \underline{0}$.

- **Initial processes**: standard as all of their actions are unexecuted.
- Set P of **well-formed processes** with unexecuted and executed actions:

$$\begin{aligned}
 & wf(\underline{0}) \\
 & wf(a . P') \quad \text{iff} \quad initial(P') \\
 & wf(a^\dagger . P') \quad \text{iff} \quad wf(P') \\
 & wf(P' \sqcup \rho^\neg) \quad \text{iff} \quad wf(P') \\
 & wf(P_1 + P_2) \quad \text{iff} \quad (wf(P_1) \wedge initial(P_2)) \vee \\
 & \quad \quad \quad (initial(P_1) \wedge wf(P_2)) \\
 & wf(P_1 \parallel_L P_2) \quad \text{iff} \quad wf(P_1) \wedge wf(P_2)
 \end{aligned}$$

- $\underline{0}$ is both initial and well-formed.
- Any initial process is well-formed too.
- P also contains processes that are not initial: $a^\dagger . b . \underline{0}$.
- **Past actions can never follow future actions**: $b . a^\dagger . \underline{0} \notin P$.
- **Alternative processes cannot be both non-initial**: $a^\dagger . \underline{0} + b^\dagger . \underline{0} \notin P$.

- Proved labeled transition system $(P, \Theta, \longrightarrow)$ with $\longrightarrow \subseteq P \times \Theta \times P$.
- Labeling every transition with a **proof term** [BoudolCastellani88] will enable the uniform derivation of expansion laws.
- Action preceded by the operators in the scope of which it occurs:

$$\theta ::= a \mid .a\theta \mid \sqsubset_{\rho}\theta \mid \dot{+}\theta \mid \dot{-}\theta \mid \ll_L\theta \mid \ll_L\theta \mid \langle\theta, \theta\rangle_L$$

- Set $\mathbb{P} \subsetneq P$ of **reachable processes** from an initial one: $a^{\dagger}.\underline{0} \parallel_{\{a\}} \underline{0} \notin \mathbb{P}$.

- Proved labeled transition system $(P, \Theta, \longrightarrow)$ with $\longrightarrow \subseteq P \times \Theta \times P$.
- Labeling every transition with a **proof term** [BoudolCastellani88] will enable the uniform derivation of expansion laws.
- Action preceded by the operators in the scope of which it occurs:

$$\theta ::= a \mid ._a\theta \mid \sqsubset_p\theta \mid \dot{+}\theta \mid \dot{-}\theta \mid \parallel_L\theta \mid \llbracket_L\theta \mid \langle\theta, \theta\rangle_L$$

- Set $\mathbb{P} \subsetneq P$ of **reachable processes** from an initial one: $a^\dagger.\underline{0} \parallel_{\{a\}} \underline{0} \notin \mathbb{P}$.
- Single transition relation viewed as symmetric to meet **loop property**: *executed actions can be undone and undone actions can be redone.*
- Like in [DMV90] a transition $P \xrightarrow{\theta} P'$ goes:
 - forward if it is viewed as an outgoing transition of P , in which case action $act(\theta)$ is done;
 - backward if it is viewed as an incoming transition of P' , in which case action $act(\theta)$ is undone.

- Proved labeled transition system $(P, \Theta, \longrightarrow)$ with $\longrightarrow \subseteq P \times \Theta \times P$.
- Labeling every transition with a **proof term** [BoudolCastellani88] will enable the uniform derivation of expansion laws.

- Action preceded by the operators in the scope of which it occurs:

$$\theta ::= a \mid .a\theta \mid \sqsubset_p\theta \mid \dot{+}\theta \mid \dot{-}\theta \mid \parallel_L\theta \mid \ll_L\theta \mid \langle\theta, \theta\rangle_L$$

- Set $\mathbb{P} \subsetneq P$ of **reachable processes** from an initial one: $a^\dagger.\underline{0} \parallel_{\{a\}} \underline{0} \notin \mathbb{P}$.
- Single transition relation viewed as symmetric to meet **loop property**: *executed actions can be undone and undone actions can be redone*.
- Like in [DMV90] a transition $P \xrightarrow{\theta} P'$ goes:
 - forward if it is viewed as an outgoing transition of P , in which case action $act(\theta)$ is done;
 - backward if it is viewed as an incoming transition of P' , in which case action $act(\theta)$ is undone.
- Since all information needed to enable reversibility is in the syntax, **action prefix and choice are made static** by the semantics [PU07].

- Operational semantic rules for action prefix (traditionally dynamic):

$$\frac{\text{initial}(P)}{a.P \xrightarrow{a} a^\dagger.P} \qquad \frac{P \xrightarrow{\theta} P'}{a^\dagger.P \xrightarrow{\cdot a^\theta} a^\dagger.P'}$$

- The prefix related to the executed action is *not discarded*.
- It becomes a \dagger -decorated part of the target process, necessary to offer again that action after rolling back.
- Additional rule for performing unexecuted actions that are preceded by already executed actions (direct consequence of making prefix static).
- This second rule propagates actions executed by initial subprocesses.
- Can we view $a^\dagger._$ as the inverse operator of $a._$?

- Semantic rules for alternative composition (traditionally dynamic):

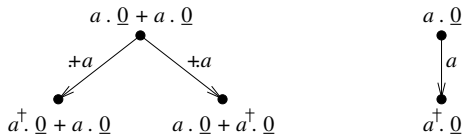
$$\frac{P_1 \xrightarrow{\theta} P'_1 \quad \text{initial}(P_2)}{P_1 + P_2 \xrightarrow{+\theta} P'_1 + P_2}$$

$$\frac{P_2 \xrightarrow{\theta} P'_2 \quad \text{initial}(P_1)}{P_1 + P_2 \xrightarrow{+\theta} P_1 + P'_2}$$

- The subprocess not involved in the executed action is *not discarded* but cannot proceed further (only the non-initial subprocess can).
- It becomes part of the target process, which is necessary for offering again the original choice after undoing all the executed actions.
- If both subprocesses are initial, both rules apply (nondet. choice).
- If not, should operator $+$ become something like $+\dagger$?
Not needed due to action decorations within either subprocess.

- The proved labeled transition system for a *sequential* process is a *tree*, whose branching points correspond to occurrences of $+$:
 - Every non-final process has at least one outgoing transition (non-final means that not all actions are decorated along one path).
 - Every non-initial process has exactly one incoming transition due to decorations associated with executed actions.

- The proved labeled transition system for a *sequential* process is a *tree*, whose branching points correspond to occurrences of $+$:
 - Every non-final process has at least one outgoing transition (non-final means that not all actions are decorated along one path).
 - Every non-initial process has exactly one incoming transition due to decorations associated with executed actions.
- Proved labeled transition systems of $a.\underline{0} + a.\underline{0}$ and $a.\underline{0}$:



- Single a -transition on the left in a forward-only process calculus.
- These two distinct processes should be considered equivalent though.

- Semantic rule for renaming (traditionally static):

$$\frac{P \xrightarrow{\theta} P'}{P \sqsubset_{\rho} \top \xrightarrow{\sqsubset_{\rho}^{\theta}} P' \sqsubset_{\rho} \top}$$

- Semantic rule for renaming (traditionally static):

$$\frac{P \xrightarrow{\theta} P'}{P \sqsubset_{\rho} \top \xrightarrow{\sqsubset_{\rho}^{\theta}} P' \sqsubset_{\rho} \top}$$

- Semantic rules for parallel composition (traditionally static):

$$\frac{P_1 \xrightarrow{\theta} P'_1 \quad \text{act}(\theta) \notin L}{P_1 \parallel_L P_2 \xrightarrow{\sqcup_L^{\theta}} P'_1 \parallel_L P_2} \quad \frac{P_2 \xrightarrow{\theta} P'_2 \quad \text{act}(\theta) \notin L}{P_1 \parallel_L P_2 \xrightarrow{\sqcup_L^{\theta}} P_1 \parallel_L P'_2}$$

$$\frac{P_1 \xrightarrow{\theta_1} P'_1 \quad P_2 \xrightarrow{\theta_2} P'_2 \quad \text{act}(\theta_1) = \text{act}(\theta_2) \in L}{P_1 \parallel_L P_2 \xrightarrow{\langle \theta_1, \theta_2 \rangle_L} P'_1 \parallel_L P'_2}$$

- Variant of the rule for synchronization that is sensitive to causality thanks to the presence of proof terms on transition labels:

$$\frac{P_1 \xrightarrow{\theta_1} P'_1 \quad P_2 \xrightarrow{\theta_2} P'_2 \quad \text{act}(\theta_1) = \text{act}(\theta_2) \in L}{P_1 \parallel_L P_2 \xrightarrow{\langle \theta_1, \theta_2 \rangle_L} \text{enr}(P'_1 \parallel_L P'_2, \langle \theta_1, \theta_2 \rangle_L)}$$

- The \dagger -decoration of every action participating in the synchronization is enriched with a proof term of the form $\langle \theta_1, \theta_2 \rangle_L$.

- Variant of the rule for synchronization that is sensitive to causality thanks to the presence of proof terms on transition labels:

$$\frac{P_1 \xrightarrow{\theta_1} P'_1 \quad P_2 \xrightarrow{\theta_2} P'_2 \quad \text{act}(\theta_1) = \text{act}(\theta_2) \in L}{P_1 \parallel_L P_2 \xrightarrow{\langle \theta_1, \theta_2 \rangle_L} \text{enr}(P'_1 \parallel_L P'_2, \langle \theta_1, \theta_2 \rangle_L)}$$

- The \dagger -decoration of every action participating in the synchronization is enriched with a proof term of the form $\langle \theta_1, \theta_2 \rangle_L$.
- The LTS of the synchronization of autoconcurrency and autocausation $(a . \underline{0} \parallel_{\emptyset} a . \underline{0}) \parallel_{\{a\}} a . a . \underline{0}$ is different from the one of $a . \underline{0} \parallel_{\emptyset} a . \underline{0}$:

- Variant of the rule for synchronization that is sensitive to causality thanks to the presence of proof terms on transition labels:

$$\frac{P_1 \xrightarrow{\theta_1} P'_1 \quad P_2 \xrightarrow{\theta_2} P'_2 \quad \text{act}(\theta_1) = \text{act}(\theta_2) \in L}{P_1 \parallel_L P_2 \xrightarrow{\langle \theta_1, \theta_2 \rangle_L} \text{enr}(P'_1 \parallel_L P'_2, \langle \theta_1, \theta_2 \rangle_L)}$$

- The \dagger -decoration of every action participating in the synchronization is enriched with a proof term of the form $\langle \theta_1, \theta_2 \rangle_L$.
- The LTS of the synchronization of autoconcurrency and autocausation $(a . \underline{0} \parallel_{\emptyset} a . \underline{0}) \parallel_{\{a\}} a . a . \underline{0}$ is different from the one of $a . \underline{0} \parallel_{\emptyset} a . \underline{0}$:

$$\begin{aligned} & (a . \underline{0} \parallel_{\emptyset} a . \underline{0}) \parallel_{\{a\}} a . a . \underline{0} \xrightarrow{\langle \llbracket_{\emptyset} a, a \rrbracket_{\{a\}} \rangle} \\ & (a^{\dagger} \langle \llbracket_{\emptyset} a, a \rrbracket_{\{a\}} \rangle . \underline{0} \parallel_{\emptyset} a . \underline{0}) \parallel_{\{a\}} a^{\dagger} \langle \llbracket_{\emptyset} a, a \rrbracket_{\{a\}} \rangle . a . \underline{0} \xrightarrow{\langle \llbracket_{\emptyset} a, \cdot a \rrbracket_{\{a\}} \rangle} \\ & (a^{\dagger} \langle \llbracket_{\emptyset} a, a \rrbracket_{\{a\}} \rangle . \underline{0} \parallel_{\emptyset} a^{\dagger} \langle \llbracket_{\emptyset} a, \cdot a \rrbracket_{\{a\}} \rangle . \underline{0}) \parallel_{\{a\}} a^{\dagger} \langle \llbracket_{\emptyset} a, a \rrbracket_{\{a\}} \rangle . a^{\dagger} \langle \llbracket_{\emptyset} a, \cdot a \rrbracket_{\{a\}} \rangle . \underline{0} \end{aligned}$$

- Variant of the rule for synchronization that is sensitive to causality thanks to the presence of proof terms on transition labels:

$$\frac{P_1 \xrightarrow{\theta_1} P'_1 \quad P_2 \xrightarrow{\theta_2} P'_2 \quad \text{act}(\theta_1) = \text{act}(\theta_2) \in L}{P_1 \parallel_L P_2 \xrightarrow{\langle \theta_1, \theta_2 \rangle_L} \text{enr}(P'_1 \parallel_L P'_2, \langle \theta_1, \theta_2 \rangle_L)}$$

- The \dagger -decoration of every action participating in the synchronization is enriched with a proof term of the form $\langle \theta_1, \theta_2 \rangle_L$.
- The LTS of the synchronization of autoconcurrency and autocausation $(a . \underline{0} \parallel_{\emptyset} a . \underline{0}) \parallel_{\{a\}} a . a . \underline{0}$ is different from the one of $a . \underline{0} \parallel_{\emptyset} a . \underline{0}$:

- $(a . \underline{0} \parallel_{\emptyset} a . \underline{0}) \parallel_{\{a\}} a . a . \underline{0} \xrightarrow{\langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}}} (a^{\dagger} \langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}} . \underline{0} \parallel_{\emptyset} a . \underline{0}) \parallel_{\{a\}} a^{\dagger} \langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}} . a . \underline{0} \xrightarrow{\langle \mathbb{I}_{\emptyset} a, . a a \rangle_{\{a\}}} (a^{\dagger} \langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}} . \underline{0} \parallel_{\emptyset} a^{\dagger} \langle \mathbb{I}_{\emptyset} a, . a a \rangle_{\{a\}} . \underline{0}) \parallel_{\{a\}} a^{\dagger} \langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}} . a^{\dagger} \langle \mathbb{I}_{\emptyset} a, . a a \rangle_{\{a\}} . \underline{0}$
- $(a . \underline{0} \parallel_{\emptyset} a . \underline{0}) \parallel_{\{a\}} a . a . \underline{0} \xrightarrow{\langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}}} (a . \underline{0} \parallel_{\emptyset} a^{\dagger} \langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}} . \underline{0}) \parallel_{\{a\}} a^{\dagger} \langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}} . a . \underline{0} \xrightarrow{\langle \mathbb{I}_{\emptyset} a, . a a \rangle_{\{a\}}} (a^{\dagger} \langle \mathbb{I}_{\emptyset} a, . a a \rangle_{\{a\}} . \underline{0} \parallel_{\emptyset} a^{\dagger} \langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}} . \underline{0}) \parallel_{\{a\}} a^{\dagger} \langle \mathbb{I}_{\emptyset} a, a \rangle_{\{a\}} . a^{\dagger} \langle \mathbb{I}_{\emptyset} a, . a a \rangle_{\{a\}} . \underline{0}$

Strong Bisimilarities for PRPC

- Bisimulation game: *outgoing* transitions for forward direction and *incoming* transitions for backward direction [DMV90].

Strong Bisimilarities for PRPC

- Bisimulation game: *outgoing* transitions for forward direction and *incoming* transitions for backward direction [DMV90].
- A symmetric relation \mathcal{B} over \mathbb{P} is a:
 - **Forward bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xrightarrow{\theta_2} P'_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.

Strong Bisimilarities for PRPC

- Bisimulation game: *outgoing* transitions for forward direction and *incoming* transitions for backward direction [DMV90].
- A symmetric relation \mathcal{B} over \mathbb{P} is a:
 - **Forward bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xrightarrow{\theta_2} P'_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - **Reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xrightarrow{\theta_2} P_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.

Strong Bisimilarities for PRPC

- Bisimulation game: *outgoing* transitions for forward direction and *incoming* transitions for backward direction [DMV90].
- A symmetric relation \mathcal{B} over \mathbb{P} is a:
 - **Forward bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xrightarrow{\theta_2} P'_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - **Reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xrightarrow{\theta_2} P_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - **Forward-reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xrightarrow{\theta_2} P'_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xrightarrow{\theta_2} P_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.

Strong Bisimilarities for PRPC

- Bisimulation game: *outgoing* transitions for forward direction and *incoming* transitions for backward direction [DMV90].
- A symmetric relation \mathcal{B} over \mathbb{P} is a:
 - **Forward bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xrightarrow{\theta_2} P'_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - **Reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xrightarrow{\theta_2} P_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - **Forward-reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xrightarrow{\theta_2} P'_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xrightarrow{\theta_2} P_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
- Largest such relations: \sim_{FB} , \sim_{RB} , \sim_{FRB} .
- In order for $P_1, P_2 \in \mathbb{P}$ to be identified by $\sim_{\text{FB}}/\sim_{\text{RB}}$ their **forward/backward ready sets** must coincide.

Discriminating Power

- $\sim_{\text{FRB}} \subsetneq \sim_{\text{FB}} \cap \sim_{\text{RB}}$:
 - The inclusion is strict because the two processes $a^\dagger.\underline{0}$ and $a^\dagger.\underline{0} + c.\underline{0}$ are identified by \sim_{FB} and \sim_{RB} , but distinguished by \sim_{FRB} .
 - \sim_{FB} and \sim_{RB} are incomparable because $a^\dagger.\underline{0} \sim_{\text{FB}} \underline{0}$ but $a^\dagger.\underline{0} \not\sim_{\text{RB}} \underline{0}$ while $a.\underline{0} \sim_{\text{RB}} \underline{0}$ but $a.\underline{0} \not\sim_{\text{FB}} \underline{0}$.

Discriminating Power

- $\sim_{\text{FRB}} \subsetneq \sim_{\text{FB}} \cap \sim_{\text{RB}}$:
 - The inclusion is strict because the two processes $a^\dagger.\underline{0}$ and $a^\dagger.\underline{0} + c.\underline{0}$ are identified by \sim_{FB} and \sim_{RB} , but distinguished by \sim_{FRB} .
 - \sim_{FB} and \sim_{RB} are incomparable because $a^\dagger.\underline{0} \sim_{\text{FB}} \underline{0}$ but $a^\dagger.\underline{0} \not\sim_{\text{RB}} \underline{0}$ while $a.\underline{0} \sim_{\text{RB}} \underline{0}$ but $a.\underline{0} \not\sim_{\text{FB}} \underline{0}$.
- **First comparative remark** (\sim_{FB} vs. \sim_{RB}):
 - $\sim_{\text{FRB}} = \sim_{\text{FB}}$ over initial processes, with \sim_{RB} strictly coarser.
 - $\sim_{\text{FRB}} \neq \sim_{\text{RB}}$ over final processes because, after going backward, discarded subprocesses come into play again for \sim_{FRB} .

Discriminating Power

- $\sim_{\text{FRB}} \subsetneq \sim_{\text{FB}} \cap \sim_{\text{RB}}$:
 - The inclusion is strict because the two processes $a^\dagger.\underline{0}$ and $a^\dagger.\underline{0} + c.\underline{0}$ are identified by \sim_{FB} and \sim_{RB} , but distinguished by \sim_{FRB} .
 - \sim_{FB} and \sim_{RB} are incomparable because $a^\dagger.\underline{0} \sim_{\text{FB}} \underline{0}$ but $a^\dagger.\underline{0} \not\sim_{\text{RB}} \underline{0}$ while $a.\underline{0} \sim_{\text{RB}} \underline{0}$ but $a.\underline{0} \not\sim_{\text{FB}} \underline{0}$.
- **First comparative remark** (\sim_{FB} vs. \sim_{RB}):
 - $\sim_{\text{FRB}} = \sim_{\text{FB}}$ over initial processes, with \sim_{RB} strictly coarser.
 - $\sim_{\text{FRB}} \neq \sim_{\text{RB}}$ over final processes because, after going backward, discarded subprocesses come into play again for \sim_{FRB} .
- $a.\underline{0} + a.\underline{0}$ and $a.\underline{0}$ are identified by all three bisimilarities as witnessed by any bisimulation containing the pairs $(a.\underline{0} + a.\underline{0}, a.\underline{0})$, $(a^\dagger.\underline{0} + a.\underline{0}, a^\dagger.\underline{0})$, $(a.\underline{0} + a^\dagger.\underline{0}, a^\dagger.\underline{0})$.

Compositionality Properties

- \sim_{FB} equates processes with different past: $a_1^\dagger . \underline{0} \sim_{\text{FB}} a_2^\dagger . \underline{0} \sim_{\text{FB}} \underline{0}$.
- \sim_{RB} equates processes with different future: $a_1 . \underline{0} \sim_{\text{RB}} a_2 . \underline{0} \sim_{\text{RB}} \underline{0}$.

Compositionality Properties

- \sim_{FB} equates processes with different past: $a_1^\dagger.\underline{0} \sim_{\text{FB}} a_2^\dagger.\underline{0} \sim_{\text{FB}} \underline{0}$.
- \sim_{RB} equates processes with different future: $a_1.\underline{0} \sim_{\text{RB}} a_2.\underline{0} \sim_{\text{RB}} \underline{0}$.
- **Second comparative remark** (\sim_{FB} vs. \sim_{RB}):
 - $a^\dagger.b.\underline{0} \sim_{\text{FB}} b.\underline{0}$ but $a^\dagger.b.\underline{0} + c.\underline{0} \not\sim_{\text{FB}} b.\underline{0} + c.\underline{0}$.
 - $a^\dagger.b.\underline{0} \not\sim_{\text{RB}} b.\underline{0}$ hence no such compositionality violation for \sim_{RB} .

Compositionality Properties

- \sim_{FB} equates processes with different past: $a_1^\dagger.\underline{0} \sim_{\text{FB}} a_2^\dagger.\underline{0} \sim_{\text{FB}} \underline{0}$.
- \sim_{RB} equates processes with different future: $a_1.\underline{0} \sim_{\text{RB}} a_2.\underline{0} \sim_{\text{RB}} \underline{0}$.
- **Second comparative remark** (\sim_{FB} vs. \sim_{RB}):
 - $a^\dagger.b.\underline{0} \sim_{\text{FB}} b.\underline{0}$ but $a^\dagger.b.\underline{0} + c.\underline{0} \not\sim_{\text{FB}} b.\underline{0} + c.\underline{0}$.
 - $a^\dagger.b.\underline{0} \not\sim_{\text{RB}} b.\underline{0}$ hence no such compositionality violation for \sim_{RB} .
- \sim_{RB} and \sim_{FRB} never identify an initial process with a non-initial one, hence \sim_{FB} has to be made sensitive to the *presence of the past*.

Compositionality Properties

- \sim_{FB} equates processes with different past: $a_1^\dagger . \underline{0} \sim_{\text{FB}} a_2^\dagger . \underline{0} \sim_{\text{FB}} \underline{0}$.
- \sim_{RB} equates processes with different future: $a_1 . \underline{0} \sim_{\text{RB}} a_2 . \underline{0} \sim_{\text{RB}} \underline{0}$.
- **Second comparative remark** (\sim_{FB} vs. \sim_{RB}):
 - $a^\dagger . b . \underline{0} \sim_{\text{FB}} b . \underline{0}$ but $a^\dagger . b . \underline{0} + c . \underline{0} \not\sim_{\text{FB}} b . \underline{0} + c . \underline{0}$.
 - $a^\dagger . b . \underline{0} \not\sim_{\text{RB}} b . \underline{0}$ hence no such compositionality violation for \sim_{RB} .
- \sim_{RB} and \sim_{FRB} never identify an initial process with a non-initial one, hence \sim_{FB} has to be made sensitive to the *presence of the past*.
- A symmetric relation \mathcal{B} over \mathbb{P} is a **past-sensitive forward bisimulation** iff it is a forward bisimulation in which $\text{initial}(P_1) \iff \text{initial}(P_2)$ for all $(P_1, P_2) \in \mathcal{B}$.
- Largest such relation: $\sim_{\text{FB:ps}}$.
- $a_1^\dagger . \underline{0} \sim_{\text{FB:ps}} a_2^\dagger . \underline{0}$, but $a^\dagger . \underline{0} \not\sim_{\text{FB:ps}} \underline{0}$ and $a^\dagger . b . \underline{0} \not\sim_{\text{FB:ps}} b . \underline{0}$.

- Let $P_1, P_2 \in \mathbb{P}$ be such that $P_1 \sim P_2$ and take arbitrary a, ρ, L, P .

- Let $P_1, P_2 \in \mathbb{P}$ be such that $P_1 \sim P_2$ and take arbitrary a, ρ, L, P .
- All strong bisimilarities are congruences w.r.t. action prefix:
 - $a.P_1 \sim a.P_2$ provided that $\text{initial}(P_1) \wedge \text{initial}(P_2)$.
 - $a^\dagger.P_1 \sim a^\dagger.P_2$.

- Let $P_1, P_2 \in \mathbb{P}$ be such that $P_1 \sim P_2$ and take arbitrary a, ρ, L, P .
- All strong bisimilarities are congruences w.r.t. action prefix:
 - $a.P_1 \sim a.P_2$ provided that $\text{initial}(P_1) \wedge \text{initial}(P_2)$.
 - $a^\dagger.P_1 \sim a^\dagger.P_2$.
- All strong bisimilarities are congruences w.r.t. renaming:
 - $P_1 \sqsubset \rho^\top \sim P_2 \sqsubset \rho^\top$.

- Let $P_1, P_2 \in \mathbb{P}$ be such that $P_1 \sim P_2$ and take arbitrary a, ρ, L, P .
- All strong bisimilarities are **congruences w.r.t. action prefix**:
 - $a.P_1 \sim a.P_2$ provided that $\text{initial}(P_1) \wedge \text{initial}(P_2)$.
 - $a^\dagger.P_1 \sim a^\dagger.P_2$.
- All strong bisimilarities are **congruences w.r.t. renaming**:
 - $P_1 \sqsubset \rho^\top \sim P_2 \sqsubset \rho^\top$.
- All strong bisimilarities are **congruences w.r.t. parallel composition**:
 - $P_1 \parallel_L P \sim P_2 \parallel_L P$ and $P \parallel_L P_1 \sim P \parallel_L P_2$
provided that $P_1 \parallel_L P, P_2 \parallel_L P, P \parallel_L P_1, P \parallel_L P_2 \in \mathbb{P}$.

- Let $P_1, P_2 \in \mathbb{P}$ be such that $P_1 \sim P_2$ and take arbitrary a, ρ, L, P .
- All strong bisimilarities are **congruences w.r.t. action prefix**:
 - $a.P_1 \sim a.P_2$ provided that $\text{initial}(P_1) \wedge \text{initial}(P_2)$.
 - $a^\dagger.P_1 \sim a^\dagger.P_2$.
- All strong bisimilarities are **congruences w.r.t. renaming**:
 - $P_1 \sqcup \rho^\neg \sim P_2 \sqcup \rho^\neg$.
- All strong bisimilarities are **congruences w.r.t. parallel composition**:
 - $P_1 \parallel_L P \sim P_2 \parallel_L P$ and $P \parallel_L P_1 \sim P \parallel_L P_2$
provided that $P_1 \parallel_L P, P_2 \parallel_L P, P \parallel_L P_1, P \parallel_L P_2 \in \mathbb{P}$.
- $\sim_{\text{FB:ps}}, \sim_{\text{RB}}, \sim_{\text{FRB}}$ are **congruences w.r.t. alternative composition**:
 - $P_1 + P \sim P_2 + P$ and $P + P_1 \sim P + P_2$
provided that $\text{initial}(P) \vee (\text{initial}(P_1) \wedge \text{initial}(P_2))$.

- Let $P_1, P_2 \in \mathbb{P}$ be such that $P_1 \sim P_2$ and take arbitrary a, ρ, L, P .
- All strong bisimilarities are **congruences w.r.t. action prefix**:
 - $a.P_1 \sim a.P_2$ provided that $\text{initial}(P_1) \wedge \text{initial}(P_2)$.
 - $a^\dagger.P_1 \sim a^\dagger.P_2$.
- All strong bisimilarities are **congruences w.r.t. renaming**:
 - $P_1 \sqcup \rho^\neg \sim P_2 \sqcup \rho^\neg$.
- All strong bisimilarities are **congruences w.r.t. parallel composition**:
 - $P_1 \parallel_L P \sim P_2 \parallel_L P$ and $P \parallel_L P_1 \sim P \parallel_L P_2$
provided that $P_1 \parallel_L P, P_2 \parallel_L P, P \parallel_L P_1, P \parallel_L P_2 \in \mathbb{P}$.
- $\sim_{\text{FB:ps}}, \sim_{\text{RB}}, \sim_{\text{FRB}}$ are **congruences w.r.t. alternative composition**:
 - $P_1 + P \sim P_2 + P$ and $P + P_1 \sim P + P_2$
provided that $\text{initial}(P) \vee (\text{initial}(P_1) \wedge \text{initial}(P_2))$.
- $\sim_{\text{FB:ps}}$ is the **coarsest congruence** w.r.t. $+$ contained in \sim_{FB} :
 - $P_1 \sim_{\text{FB:ps}} P_2$ iff $P_1 + P \sim_{\text{FB}} P_2 + P$
for all $P \in \mathbb{P}$ s.t. $\text{initial}(P) \vee (\text{initial}(P_1) \wedge \text{initial}(P_2))$.

Modal Logic Characterizations

- Properties preserved by each equivalence; diagnostic information via distinguishing formulas explaining why two processes are not bisimilar.
- Hennessy-Milner logic extended with a backward modality (and init) from which suitable fragments are taken.
- Syntax:

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

Modal Logic Characterizations

- Properties preserved by each equivalence; diagnostic information via distinguishing formulas explaining why two processes are not bisimilar.
- Hennessy-Milner logic extended with a backward modality (and init) from which suitable fragments are taken.
- Syntax:

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

- Semantics:

$$P \models \text{true} \quad \text{for all } P \in \mathbb{P}$$

$$P \models \text{init} \quad \text{iff } \text{initial}(P)$$

$$P \models \neg\phi \quad \text{iff } P \not\models \phi$$

$$P \models \phi_1 \wedge \phi_2 \quad \text{iff } P \models \phi_1 \text{ and } P \models \phi_2$$

$$P \models \langle a \rangle \phi \quad \text{iff there exists } P \xrightarrow{\theta} P' \text{ s.t. } \text{act}(\theta) = a \text{ and } P' \models \phi$$

$$P \models \langle a^\dagger \rangle \phi \quad \text{iff there exists } P' \xrightarrow{\theta} P \text{ s.t. } \text{act}(\theta) = a \text{ and } P' \models \phi$$

- Fragments characterizing the four strong bisimilarities:

	true	init	\neg	\wedge	$\langle a \rangle$	$\langle a^\dagger \rangle$
\mathcal{L}_{FB}	✓		✓	✓	✓	
$\mathcal{L}_{\text{FB:ps}}$	✓	✓	✓	✓	✓	
\mathcal{L}_{RB}	✓					✓
\mathcal{L}_{FRB}	✓		✓	✓	✓	✓

- $\mathcal{L}_{\text{FB}} / \mathcal{L}_{\text{FB:ps}} / \mathcal{L}_{\text{RB}} / \mathcal{L}_{\text{FRB}}$ characterizes $\sim_{\text{FB}} / \sim_{\text{FB:ps}} / \sim_{\text{RB}} / \sim_{\text{FRB}}$:
 $P_1 \sim_B P_2$ iff $\forall \phi \in \mathcal{L}_B. P_1 \models \phi \iff P_2 \models \phi$

- Fragments characterizing the four strong bisimilarities:

	true	init	\neg	\wedge	$\langle a \rangle$	$\langle a^\dagger \rangle$
\mathcal{L}_{FB}	✓		✓	✓	✓	
$\mathcal{L}_{\text{FB:ps}}$	✓	✓	✓	✓	✓	
\mathcal{L}_{RB}	✓					✓
\mathcal{L}_{FRB}	✓		✓	✓	✓	✓

- $\mathcal{L}_{\text{FB}} / \mathcal{L}_{\text{FB:ps}} / \mathcal{L}_{\text{RB}} / \mathcal{L}_{\text{FRB}}$ characterizes $\sim_{\text{FB}} / \sim_{\text{FB:ps}} / \sim_{\text{RB}} / \sim_{\text{FRB}}$:
 $P_1 \sim_B P_2$ iff $\forall \phi \in \mathcal{L}_B. P_1 \models \phi \iff P_2 \models \phi$
- \sim_{RB} boils down to reverse trace equivalence!
- Obvious over sequential processes because each of them has at most one incoming transition due to executed actions being decorated.

Equational Characterizations

- Fundamental equational laws; exploitable as bisimilarity-preserving rewriting rules for manipulating processes.
- Deduction system \vdash has axioms and inference rules corresponding to $\sim_{\text{FB:ps}}$, \sim_{RB} , \sim_{FRB} being equivalence relations and congruences:

Equational Characterizations

- Fundamental equational laws; exploitable as bisimilarity-preserving rewriting rules for manipulating processes.
- Deduction system \vdash has axioms and inference rules corresponding to $\sim_{\text{FB:ps}}$, \sim_{RB} , \sim_{FRB} being equivalence relations and congruences:
 - Reflexivity $P = P$, symmetry $\frac{P_1 = P_2}{P_2 = P_1}$, transitivity $\frac{P_1 = P_2 \quad P_2 = P_3}{P_1 = P_3}$.

Equational Characterizations

- Fundamental equational laws; exploitable as bisimilarity-preserving rewriting rules for manipulating processes.
- Deduction system \vdash has axioms and inference rules corresponding to $\sim_{\text{FB:ps}}$, \sim_{RB} , \sim_{FRB} being equivalence relations and congruences:

- Reflexivity $P = P$, symmetry $\frac{P_1 = P_2}{P_2 = P_1}$, transitivity $\frac{P_1 = P_2 \quad P_2 = P_3}{P_1 = P_3}$.
- \cdot -Substitutivity: $\frac{P_1 = P_2 \quad \text{initial}(P_1) \wedge \text{initial}(P_2)}{a \cdot P_1 = a \cdot P_2}$, $\frac{P_1 = P_2}{a^\dagger \cdot P_1 = a^\dagger \cdot P_2}$.
- \sqcap -substitutivity: $\frac{P_1 = P_2}{P_1 \sqcap \rho^\top = P_2 \sqcap \rho^\top}$.
- $+$ -Substitutivity: $\frac{P_1 = P_2 \quad \text{initial}(P) \vee (\text{initial}(P_1) \wedge \text{initial}(P_2))}{P_1 + P = P_2 + P \quad P + P_1 = P + P_2}$.
- \parallel -substitutivity: $\frac{P_1 = P_2 \quad P_1 \parallel_L P, P_2 \parallel_L P, P \parallel_L P_1, P \parallel_L P_2 \in \mathbb{P}}{P_1 \parallel_L P = P_2 \parallel_L P \quad P \parallel_L P_1 = P \parallel_L P_2}$.

Equational Characterizations

- Fundamental equational laws; exploitable as bisimilarity-preserving rewriting rules for manipulating processes.
- Deduction system \vdash has axioms and inference rules corresponding to $\sim_{\text{FB:ps}}$, \sim_{RB} , \sim_{FRB} being equivalence relations and congruences:

- Reflexivity $P = P$, symmetry $\frac{P_1 = P_2}{P_2 = P_1}$, transitivity $\frac{P_1 = P_2 \quad P_2 = P_3}{P_1 = P_3}$.

- .-Substitutivity: $\frac{P_1 = P_2 \quad \text{initial}(P_1) \wedge \text{initial}(P_2)}{a.P_1 = a.P_2}$, $\frac{P_1 = P_2}{a^\dagger.P_1 = a^\dagger.P_2}$.

- \sqcap -substitutivity: $\frac{P_1 = P_2}{P_1 \sqcap \rho^\top = P_2 \sqcap \rho^\top}$.

- + -Substitutivity: $\frac{P_1 = P_2 \quad \text{initial}(P) \vee (\text{initial}(P_1) \wedge \text{initial}(P_2))}{P_1 + P = P_2 + P \quad P + P_1 = P + P_2}$.

- \parallel -substitutivity: $\frac{P_1 = P_2 \quad P_1 \parallel_L P, P_2 \parallel_L P, P \parallel_L P_1, P \parallel_L P_2 \in \mathbb{P}}{P_1 \parallel_L P = P_2 \parallel_L P \quad P \parallel_L P_1 = P \parallel_L P_2}$.

- \vdash is **sound and complete** w.r.t. \sim when $\vdash P_1 = P_2$ iff $P_1 \sim P_2$.

- Operator-specific axioms for **renaming-free sequential processes**:

(A ₁)		$(P + Q) + R = P + (Q + R)$	where at least two are initial
(A ₂)		$P + Q = Q + P$	where $initial(P) \vee initial(Q)$
(A ₃)		$P + \underline{0} = P$	
(A ₄)	$[\sim_{FB:ps}]$	$a^\dagger.P = b^\dagger.P$	if $initial(P)$
(A ₅)	$[\sim_{FB:ps}]$	$a^\dagger.P = P$	if $\neg initial(P)$
(A ₆)	$[\sim_{FB:ps}]$	$P + Q = P$	if $\neg initial(P)$, where $initial(Q)$
(A ₇)	$[\sim_{RB}]$	$a.P = P$	where $initial(P)$
(A ₈)	$[\sim_{RB}]$	$P + Q = P$	if $initial(Q)$
(A ₉)	$[\sim_{FB:ps}]$	$P + P = P$	where $initial(P)$
(A ₁₀)	$[\sim_{FRB}]$	$P + Q = P$	if $initial(Q) \wedge to_initial(P) = Q$

- A₈ subsumes A₃ (with $Q = \underline{0}$) and A₉ (with $Q = P$).
- A₉ and A₆ apply in two different cases (P initial or not).
- A₁₀ originally developed in [LanesePhillips21].
- $\vdash_{4,5,6,9}^{1,2,3} / \vdash_{7,8}^{1,2} / \vdash_{10}^{1,2,3}$ **sound and complete** for $\sim_{FB:ps} / \sim_{RB} / \sim_{FRB}$.
- Third comparative remark**: explicit vs. implicit idempotency.

- Axioms for **renaming**:

(A ₁₁)	$\underline{0} \sqsubseteq \rho^\top = \underline{0}$	
(A ₁₂)	$(a.P) \sqsubseteq \rho^\top = \rho(a).(P \sqsubseteq \rho^\top)$	where $initial(P)$
(A ₁₃)	$(a^\dagger.P) \sqsubseteq \rho^\top = \rho(a)^\dagger.(P \sqsubseteq \rho^\top)$	
(A ₁₄)	$(P + Q) \sqsubseteq \rho^\top = (P \sqsubseteq \rho^\top) + (Q \sqsubseteq \rho^\top)$	where $initial(P) \vee initial(Q)$

- They progressively remove all occurrences of renaming.
- $\sim_{\text{FB:ps}}$ needs all of them.
- \sim_{RB} only needs A₁₁ and A₁₃.
- \sim_{FRB} needs all of them.
- We will see later on **expansion laws for parallel composition**.

Weak Bisimilarities for PRPC

- Abstracting from possibly empty sequences \Longrightarrow of τ -transitions:

$$\xRightarrow{\hat{\theta}} = \Longrightarrow \text{ if } \text{act}(\theta) = \tau, \quad \xRightarrow{\hat{\theta}} = \Longrightarrow \xrightarrow{\theta} \Longrightarrow \text{ if } \text{act}(\theta) \neq \tau.$$

Weak Bisimilarities for PRPC

- Abstracting from possibly empty sequences \Longrightarrow of τ -transitions:
 $\xRightarrow{\hat{\theta}} = \Longrightarrow$ if $act(\theta) = \tau$, $\xRightarrow{\hat{\theta}} = \Longrightarrow \xrightarrow{\theta} \Longrightarrow$ if $act(\theta) \neq \tau$.
- A symmetric relation \mathcal{B} over \mathbb{P} is a:
 - **Weak forward bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xRightarrow{\hat{\theta}_2} P'_2 . act(\theta_1) = act(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.

Weak Bisimilarities for PRPC

- Abstracting from possibly empty sequences \Longrightarrow of τ -transitions:
 $\xRightarrow{\hat{\theta}} = \Longrightarrow$ if $act(\theta) = \tau$, $\xRightarrow{\hat{\theta}} = \Longrightarrow \xrightarrow{\theta} \Longrightarrow$ if $act(\theta) \neq \tau$.
- A symmetric relation \mathcal{B} over \mathbb{P} is a:
 - **Weak forward bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xRightarrow{\hat{\theta}_2} P'_2 . act(\theta_1) = act(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - **Weak reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xRightarrow{\hat{\theta}_2} P_2 . act(\theta_1) = act(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.

Weak Bisimilarities for PRPC

- Abstracting from possibly empty sequences \Longrightarrow of τ -transitions:

$$\xRightarrow{\hat{\theta}} = \Longrightarrow \text{ if } \text{act}(\theta) = \tau, \quad \xRightarrow{\hat{\theta}} = \Longrightarrow \xrightarrow{\theta} \Longrightarrow \text{ if } \text{act}(\theta) \neq \tau.$$

- A symmetric relation \mathcal{B} over \mathbb{P} is a:

- **Weak forward bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:

- $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xRightarrow{\hat{\theta}_2} P'_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}.$

- **Weak reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:

- $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xRightarrow{\hat{\theta}_2} P_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}.$

- **Weak forward-reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:

- $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xRightarrow{\hat{\theta}_2} P'_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}.$

- $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xRightarrow{\hat{\theta}_2} P_2 . \text{act}(\theta_1) = \text{act}(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}.$

Weak Bisimilarities for PRPC

- Abstracting from possibly empty sequences \Longrightarrow of τ -transitions:
 $\xRightarrow{\hat{\theta}} = \Longrightarrow$ if $act(\theta) = \tau$, $\xRightarrow{\hat{\theta}} = \Longrightarrow \xrightarrow{\theta} \Longrightarrow$ if $act(\theta) \neq \tau$.
- A symmetric relation \mathcal{B} over \mathbb{P} is a:
 - **Weak forward bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xRightarrow{\hat{\theta}_2} P'_2 . act(\theta_1) = act(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - **Weak reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xRightarrow{\hat{\theta}_2} P_2 . act(\theta_1) = act(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - **Weak forward-reverse bisimulation** iff, whenever $(P_1, P_2) \in \mathcal{B}$, then:
 - $\forall P_1 \xrightarrow{\theta_1} P'_1 . \exists P_2 \xRightarrow{\hat{\theta}_2} P'_2 . act(\theta_1) = act(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
 - $\forall P'_1 \xrightarrow{\theta_1} P_1 . \exists P'_2 \xRightarrow{\hat{\theta}_2} P_2 . act(\theta_1) = act(\theta_2) \wedge (P'_1, P'_2) \in \mathcal{B}$.
- Largest such relations: \approx_{FB} , \approx_{RB} , \approx_{FRB} .
- Alternative definitions with $\xRightarrow{\hat{\theta}_1}$ in place of $\xrightarrow{\theta_1}$.
- In order for $P_1, P_2 \in \mathbb{P}$ to be identified by $\approx_{\text{FB}}/\approx_{\text{RB}}$ their **weak forward/backward ready sets** have to coincide.

Discriminating Power

- Each weak bisimilarity is strictly coarser than its strong counterpart.
- $\approx_{\text{FRB}} \subsetneq \approx_{\text{FB}} \cap \approx_{\text{RB}}$ with \approx_{FB} and \approx_{RB} being incomparable.

Discriminating Power

- Each weak bisimilarity is strictly coarser than its strong counterpart.
- $\approx_{\text{FRB}} \subsetneq \approx_{\text{FB}} \cap \approx_{\text{RB}}$ with \approx_{FB} and \approx_{RB} being incomparable.
- $\approx_{\text{FRB}} \neq \approx_{\text{FB}}$ over initial processes:
 - $\tau.a.\underline{0} + a.\underline{0} + b.\underline{0}$ and $\tau.a.\underline{0} + b.\underline{0}$ are identified by \approx_{FB} but told apart by \approx_{FRB} :
 - Doing a on the left is matched by doing τ and then a on the right.
 - Undoing a on the right cannot be matched on the left.

Discriminating Power

- Each weak bisimilarity is strictly coarser than its strong counterpart.
- $\approx_{\text{FRB}} \subsetneq \approx_{\text{FB}} \cap \approx_{\text{RB}}$ with \approx_{FB} and \approx_{RB} being incomparable.
- $\approx_{\text{FRB}} \neq \approx_{\text{FB}}$ over initial processes:
 - $\tau.a.\underline{0} + a.\underline{0} + b.\underline{0}$ and $\tau.a.\underline{0} + b.\underline{0}$ are identified by \approx_{FB} but told apart by \approx_{FRB} :
 - Doing a on the left is matched by doing τ and then a on the right.
 - Undoing a on the right cannot be matched on the left.
 - $c.(\tau.a.\underline{0} + a.\underline{0} + b.\underline{0})$ and $c.(\tau.a.\underline{0} + b.\underline{0})$ is an analogous counterexample with non-initial τ -actions:
 - Doing c on one side is matched by doing c on the other side.
 - Doing a on the left is matched by doing τ and then a on the right.
 - Undoing a on the right cannot be matched on the left.

Compositionality Properties

- Neither \approx_{FB} nor \approx_{FRB} is compositional:
 - $a^\dagger.b.\underline{0} \approx_{\text{FB}} b.\underline{0}$ but $a^\dagger.b.\underline{0} + c.\underline{0} \not\approx_{\text{FB}} b.\underline{0} + c.\underline{0}$ (same as \sim_{FB}).
 - $\tau.a.\underline{0} \approx_{\text{FB}} a.\underline{0}$ but $\tau.a.\underline{0} + b.\underline{0} \not\approx_{\text{FB}} a.\underline{0} + b.\underline{0}$.
 - $\tau.a.\underline{0} \approx_{\text{FRB}} a.\underline{0}$ but $\tau.a.\underline{0} + b.\underline{0} \not\approx_{\text{FRB}} a.\underline{0} + b.\underline{0}$.
- The weak congruence construction à la Milner does not work here, past sensitivity is the solution again.

Compositionality Properties

- Neither \approx_{FB} nor \approx_{FRB} is compositional:
 - $a^\dagger.b.\underline{0} \approx_{\text{FB}} b.\underline{0}$ but $a^\dagger.b.\underline{0} + c.\underline{0} \not\approx_{\text{FB}} b.\underline{0} + c.\underline{0}$ (same as \sim_{FB}).
 - $\tau.a.\underline{0} \approx_{\text{FB}} a.\underline{0}$ but $\tau.a.\underline{0} + b.\underline{0} \not\approx_{\text{FB}} a.\underline{0} + b.\underline{0}$.
 - $\tau.a.\underline{0} \approx_{\text{FRB}} a.\underline{0}$ but $\tau.a.\underline{0} + b.\underline{0} \not\approx_{\text{FRB}} a.\underline{0} + b.\underline{0}$.
- The weak congruence construction à la Milner does not work here, past sensitivity is the solution again.
- A symmetric relation \mathcal{B} over \mathbb{P} is a **weak past-sensitive forward bisim.** iff it is a weak forward bisim. in which $\text{initial}(P_1) \iff \text{initial}(P_2)$ for all $(P_1, P_2) \in \mathcal{B}$.
- A symm. rel. \mathcal{B} over \mathbb{P} is a **weak past-sensitive forward-reverse bisim.** iff it is a weak forward-reverse bisim. s.t. $\text{initial}(P_1) \iff \text{initial}(P_2)$ for all $(P_1, P_2) \in \mathcal{B}$.
- Largest such relations: $\approx_{\text{FB:ps}}$, $\approx_{\text{FRB:ps}}$.
- $\sim_{\text{FRB}} \subsetneq \approx_{\text{FRB:ps}}$ as the former satisfies the initiality condition.

- Let $P_1, P_2 \in \mathbb{P}$ be such that $P_1 \approx P_2$ and take arbitrary a, ρ, L, P .
- All weak bisimilarities are **congruences w.r.t. action prefix**:
 - $a.P_1 \approx a.P_2$ provided that $\text{initial}(P_1) \wedge \text{initial}(P_2)$.
 - $a^\dagger.P_1 \approx a^\dagger.P_2$.
- All weak bisimilarities are **congruences w.r.t. renaming**:
 - $P_1 \sqsubset \rho^\neg \approx P_2 \sqsubset \rho^\neg$.
- All weak bisimilarities are **congruences w.r.t. parallel composition**:
 - $P_1 \parallel_L P \approx P_2 \parallel_L P$ and $P \parallel_L P_1 \approx P \parallel_L P_2$
provided that $P_1 \parallel_L P, P_2 \parallel_L P, P \parallel_L P_1, P \parallel_L P_2 \in \mathbb{P}$.

- Let $P_1, P_2 \in \mathbb{P}$ be such that $P_1 \approx P_2$ and take arbitrary a, ρ, L, P .
- All weak bisimilarities are **congruences w.r.t. action prefix**:
 - $a.P_1 \approx a.P_2$ provided that $\text{initial}(P_1) \wedge \text{initial}(P_2)$.
 - $a^\dagger.P_1 \approx a^\dagger.P_2$.
- All weak bisimilarities are **congruences w.r.t. renaming**:
 - $P_1 \downarrow \rho^\top \approx P_2 \downarrow \rho^\top$.
- All weak bisimilarities are **congruences w.r.t. parallel composition**:
 - $P_1 \parallel_L P \approx P_2 \parallel_L P$ and $P \parallel_L P_1 \approx P \parallel_L P_2$
provided that $P_1 \parallel_L P, P_2 \parallel_L P, P \parallel_L P_1, P \parallel_L P_2 \in \mathbb{P}$.
- $\approx_{\text{FB:ps}}, \approx_{\text{RB}}, \approx_{\text{FRB:ps}}$ are **congruences w.r.t. alternative composition**:
 - $P_1 + P \approx P_2 + P$ and $P + P_1 \approx P + P_2$
provided that $\text{initial}(P) \vee (\text{initial}(P_1) \wedge \text{initial}(P_2))$.
- $\approx_{\text{FB:ps}}$ is the **coarsest congruence** w.r.t. $+$ contained in \approx_{FB} :
 - $P_1 \approx_{\text{FB:ps}} P_2$ iff $P_1 + P \approx_{\text{FB}} P_2 + P$
for all $P \in \mathbb{P}$ s.t. $\text{initial}(P) \vee (\text{initial}(P_1) \wedge \text{initial}(P_2))$.
- $\approx_{\text{FRB:ps}}$ is the **coarsest congruence** w.r.t. $+$ contained in \approx_{FRB} :
 - $P_1 \approx_{\text{FRB:ps}} P_2$ iff $P_1 + P \approx_{\text{FRB}} P_2 + P$
for all $P \in \mathbb{P}$ s.t. $\text{initial}(P) \vee (\text{initial}(P_1) \wedge \text{initial}(P_2))$.

Modal Logic Characterizations

- Modal logic with weak forward/backward modalities ($a \in \mathcal{A} \setminus \{\tau\}$):

$\phi ::= \text{true} \mid \text{init} \mid \neg\phi \mid \phi \wedge \phi \mid \langle\!\langle \tau \rangle\!\rangle\phi \mid \langle\!\langle a \rangle\!\rangle\phi \mid \langle\!\langle \tau^\dagger \rangle\!\rangle\phi \mid \langle\!\langle a^\dagger \rangle\!\rangle\phi$

Modal Logic Characterizations

- Modal logic with weak forward/backward modalities ($a \in \mathcal{A} \setminus \{\tau\}$):

$$\phi ::= \text{true} \mid \text{init} \mid \neg\phi \mid \phi \wedge \phi \mid \langle\!\langle\tau\rangle\!\rangle\phi \mid \langle\!\langle a\rangle\!\rangle\phi \mid \langle\!\langle\tau^\dagger\rangle\!\rangle\phi \mid \langle\!\langle a^\dagger\rangle\!\rangle\phi$$

- Semantics:

$$P \models \text{true} \quad \text{for all } P \in \mathbb{P}$$

$$P \models \text{init} \quad \text{iff } \text{initial}(P)$$

$$P \models \neg\phi \quad \text{iff } P \not\models \phi$$

$$P \models \phi_1 \wedge \phi_2 \quad \text{iff } P \models \phi_1 \text{ and } P \models \phi_2$$

$$P \models \langle\!\langle\tau\rangle\!\rangle\phi \quad \text{iff there exists } P \Longrightarrow P' \text{ such that } P' \models \phi$$

$$P \models \langle\!\langle a\rangle\!\rangle\phi \quad \text{iff there exists } P \xrightarrow{\theta} P' \text{ s.t. } \text{act}(\theta) = a \text{ and } P' \models \phi$$

$$P \models \langle\!\langle\tau^\dagger\rangle\!\rangle\phi \quad \text{iff there exists } P' \Longrightarrow P \text{ such that } P' \models \phi$$

$$P \models \langle\!\langle a^\dagger\rangle\!\rangle\phi \quad \text{iff there exists } P' \xrightarrow{\theta} P \text{ s.t. } \text{act}(\theta) = a \text{ and } P' \models \phi$$

- Fragments characterizing the five weak bisimilarities:

	true	init	\neg	\wedge	$\langle\langle\tau\rangle\rangle$	$\langle\langle a \rangle\rangle$	$\langle\langle\tau^\dagger\rangle\rangle$	$\langle\langle a^\dagger\rangle\rangle$
$\mathcal{L}_{\text{FB}}^\tau$	✓		✓	✓	✓	✓		
$\mathcal{L}_{\text{FB:ps}}^\tau$	✓	✓	✓	✓	✓	✓		
$\mathcal{L}_{\text{RB}}^\tau$	✓						✓	✓
$\mathcal{L}_{\text{FRB}}^\tau$	✓		✓	✓	✓	✓	✓	✓
$\mathcal{L}_{\text{FRB:ps}}^\tau$	✓	✓	✓	✓	✓	✓	✓	✓

- $\mathcal{L}_{\text{FB}}^\tau / \mathcal{L}_{\text{FB:ps}}^\tau / \mathcal{L}_{\text{RB}}^\tau / \mathcal{L}_{\text{FRB}}^\tau / \mathcal{L}_{\text{FRB:ps}}^\tau$ characterizes

$\approx_{\text{FB}} / \approx_{\text{FB:ps}} / \approx_{\text{RB}} / \approx_{\text{FRB}} / \approx_{\text{FRB:ps}}$:

$$P_1 \approx_B P_2 \text{ iff } \forall \phi \in \mathcal{L}_B^\tau. P_1 \models \phi \iff P_2 \models \phi$$

Equational Characterizations

- Additional operator-specific axioms called τ -laws:

(A_1^τ)	$[\approx_{\text{FB:ps}}]$	$a . \tau . P = a . P$	where $\text{initial}(P)$
(A_2^τ)	$[\approx_{\text{FB:ps}}]$	$P + \tau . P = \tau . P$	where $\text{initial}(P)$
(A_3^τ)	$[\approx_{\text{FB:ps}}]$	$a . (P + \tau . Q) + a . Q = a . (P + \tau . Q)$	where P, Q initial
(A_4^τ)	$[\approx_{\text{FB:ps}}]$	$a^\dagger . \tau . P = a^\dagger . P$	where $\text{initial}(P)$
(A_5^τ)	$[\approx_{\text{RB}}]$	$\tau^\dagger . P = P$	
(A_6^τ)	$[\approx_{\text{FRB:ps}}]$	$a . (\tau . (P + Q) + P) = a . (P + Q)$	where P, Q initial
(A_7^τ)	$[\approx_{\text{FRB:ps}}]$	$a^\dagger . (\tau . (P + Q) + P') = a^\dagger . (P' + Q)$	if $\text{to_initial}(P') = P$, where P, Q initial
(A_8^τ)	$[\approx_{\text{FRB:ps}}]$	$a^\dagger . (\tau^\dagger . (P' + Q) + P) = a^\dagger . (P' + Q)$	if $\text{to_initial}(P') = P$, where $\text{initial}(P)$

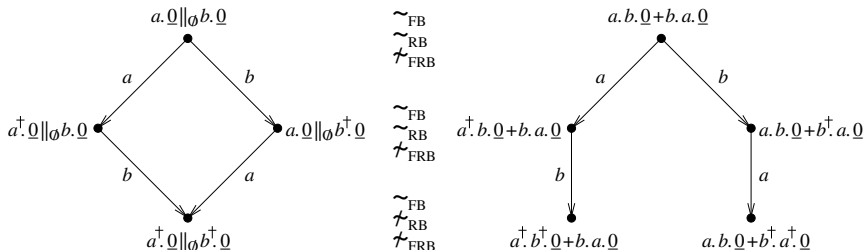
- $A_1^\tau, A_2^\tau, A_3^\tau$ are Milner τ -laws, A_4^τ needed for completeness.
- A_5^τ is a variant of $\tau . P = P$ (not valid for weak bisim. congruence).
- A_6^τ is Van Glabbeek-Weijland τ -law, A_7^τ and A_8^τ needed for complet.
- $\vdash_{1,2,3,4,5,6,9}^{1,2,3,4,5,6,9} / \vdash_5^{1,2,7,8} / \vdash_{6,7,8}^{1,2,3,10}$ is sound and complete for
 $\approx_{\text{FB:ps}} / \approx_{\text{RB}} / \approx_{\text{FRB:ps}}$ over renaming-free sequential processes.
- \approx_{FRB} is branching bisimilarity over initial sequential processes!

Expansion Laws for Parallel Composition

- In forward-only process calculi $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ and $a.b.\underline{0} + b.a.\underline{0}$ are deemed equivalent: *the latter is the expansion of the former.*

Expansion Laws for Parallel Composition

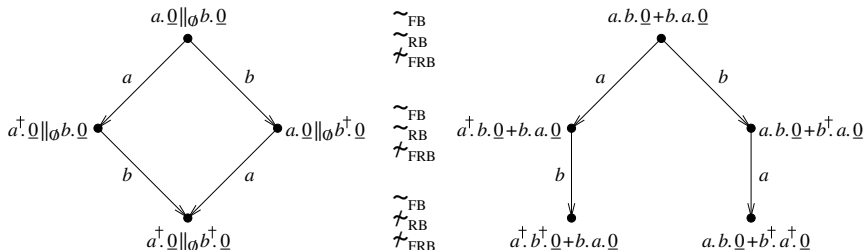
- In forward-only process calculi $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ and $a.b.\underline{0} + b.a.\underline{0}$ are deemed equivalent: *the latter is the expansion of the former*.
- In our reversible setting we obtain instead ($a \neq b$):



- \sim_{FB} is interleaving, while \sim_{RB} and \sim_{FRB} are truly concurrent.

Expansion Laws for Parallel Composition

- In forward-only process calculi $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ and $a.b.\underline{0} + b.a.\underline{0}$ are deemed equivalent: *the latter is the expansion of the former*.
- In our reversible setting we obtain instead ($a \neq b$):



- \sim_{FB} is interleaving, while \sim_{RB} and \sim_{FRB} are truly concurrent.
- What are the expansion laws for the six bisimulation congruences $\sim_{\text{FB:ps}}$, \sim_{RB} , \sim_{FRB} , $\approx_{\text{FB:ps}}$, \approx_{RB} , $\approx_{\text{FRB:ps}}$?

- Expansion laws for forward-only calculi in the **interleaving** setting are used to **identify** $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ and $a.b.\underline{0} + b.a.\underline{0}$.

- Expansion laws for forward-only calculi in the **interleaving** setting are used to **identify** $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ and $a.b.\underline{0} + b.a.\underline{0}$.
- Also used in **truly concurrent** semantics to **distinguish** those processes by adding suitable **discriminating information within action prefixes**:

- Expansion laws for forward-only calculi in the **interleaving** setting are used to **identify** $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ and $a.b.\underline{0} + b.a.\underline{0}$.
- Also used in **truly concurrent** semantics to **distinguish** those processes by adding suitable **discriminating information within action prefixes**:
 - **Causal bisimilarity** [DarondeauDegano90] (corresponding to history-preserving bisimilarity [RabinovichTrakhtenbrot88]): every action is enriched with the set of its causing actions each of which is expressed as a numeric backward pointer, hence we get $\langle a, \emptyset \rangle . \langle b, \emptyset \rangle . \underline{0} + \langle b, \emptyset \rangle . \langle a, \emptyset \rangle . \underline{0}$ and $\langle a, \emptyset \rangle . \langle b, \{1\} \rangle . \underline{0} + \langle b, \emptyset \rangle . \langle a, \{1\} \rangle . \underline{0}$.

- Expansion laws for forward-only calculi in the **interleaving** setting are used to **identify** $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ and $a.b.\underline{0} + b.a.\underline{0}$.
- Also used in **truly concurrent** semantics to **distinguish** those processes by adding suitable **discriminating information within action prefixes**:
 - **Causal bisimilarity** [DarondeauDegano90] (corresponding to history-preserving bisimilarity [RabinovichTrakhtenbrot88]): every action is enriched with the set of its causing actions each of which is expressed as a numeric backward pointer, hence we get $\langle a, \emptyset \rangle . \langle b, \emptyset \rangle . \underline{0} + \langle b, \emptyset \rangle . \langle a, \emptyset \rangle . \underline{0}$ and $\langle a, \emptyset \rangle . \langle b, \{1\} \rangle . \underline{0} + \langle b, \emptyset \rangle . \langle a, \{1\} \rangle . \underline{0}$.
 - **Location bisimilarity** [BoudolCastellaniHennessyKiehn94]: every action is enriched with the name of the location in which it is executed, hence we get $\langle a, l_a \rangle . \langle b, l_b \rangle . \underline{0} + \langle b, l_b \rangle . \langle a, l_a \rangle . \underline{0}$ and $\langle a, l_a \rangle . \langle b, l_a l_b \rangle . \underline{0} + \langle b, l_b \rangle . \langle a, l_b l_a \rangle . \underline{0}$.

- Expansion laws for forward-only calculi in the **interleaving** setting are used to **identify** $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ and $a.b.\underline{0} + b.a.\underline{0}$.
- Also used in **truly concurrent** semantics to **distinguish** those processes by adding suitable **discriminating information within action prefixes**:
 - **Causal bisimilarity** [DarondeauDegano90] (corresponding to history-preserving bisimilarity [RabinovichTrakhtenbrot88]): every action is enriched with the set of its causing actions each of which is expressed as a numeric backward pointer, hence we get $\langle a, \emptyset \rangle . \langle b, \emptyset \rangle . \underline{0} + \langle b, \emptyset \rangle . \langle a, \emptyset \rangle . \underline{0}$ and $\langle a, \emptyset \rangle . \langle b, \{1\} \rangle . \underline{0} + \langle b, \emptyset \rangle . \langle a, \{1\} \rangle . \underline{0}$.
 - **Location bisimilarity** [BoudolCastellaniHennessyKiehn94]: every action is enriched with the name of the location in which it is executed, hence we get $\langle a, l_a \rangle . \langle b, l_b \rangle . \underline{0} + \langle b, l_b \rangle . \langle a, l_a \rangle . \underline{0}$ and $\langle a, l_a \rangle . \langle b, l_a l_b \rangle . \underline{0} + \langle b, l_b \rangle . \langle a, l_b l_a \rangle . \underline{0}$.
 - **Pomset bisimilarity** [BoudolCastellani88]: a prefix may contain a combination of actions that are causally related or independent, hence the former process becomes $a.b.\underline{0} + b.a.\underline{0} + (a \parallel b).\underline{0}$.

- How to uniformly derive the six expansion laws?
- Proved trees approach of [DeganoPriami92].
- Enabled by the use of proof terms in the operational semantics!

- How to uniformly derive the six expansion laws?
- Proved trees approach of [DeganoPriami92].
- Enabled by the use of proof terms in the operational semantics!
- Interleaving: proof terms are reduced to the actions they contain, hence we are done for $\sim_{\text{FB:ps}}$ and $\approx_{\text{FB:ps}}$.

- How to uniformly derive the six expansion laws?
- Proved trees approach of [DeganoPriami92].
- Enabled by the use of proof terms in the operational semantics!
- Interleaving: proof terms are reduced to the actions they contain, hence we are done for $\sim_{\text{FB:ps}}$ and $\approx_{\text{FB:ps}}$.
- True concurrency: they are transformed into actions extended with suitable discriminating information (then encode processes accordingly).
- Information already available in the operational semantics for causal bisimilarity, location bisimilarity, pomset bisimilarity.
- Unfortunately not available in our proved operational semantics for \sim_{RB} , \sim_{FRB} , \approx_{RB} , $\approx_{\text{FRB:ps}}$!

- The equivalence of interest drives an **observation function** that maps proof terms to the required observations.
- Observation function ℓ applied to proof terms labeling transitions, so that $\ell(\theta_1)$ and $\ell(\theta_2)$ are considered in the bisimulation game.
- Action preservation: $\ell(\theta_1) = \ell(\theta_2)$ implies $act(\theta_1) = act(\theta_2)$.
- ℓ may depend on other possible parameters that are present in the proved labeled transition system.

- The equivalence of interest drives an **observation function** that maps proof terms to the required observations.
- Observation function ℓ applied to proof terms labeling transitions, so that $\ell(\theta_1)$ and $\ell(\theta_2)$ are considered in the bisimulation game.
- Action preservation: $\ell(\theta_1) = \ell(\theta_2)$ implies $act(\theta_1) = act(\theta_2)$.
- ℓ may depend on other possible parameters that are present in the proved labeled transition system.
- $\sim_{\text{FB:ps}:\ell_F}$, $\sim_{\text{RB}:\ell_R}$, $\sim_{\text{FRB}:\ell_{FR}}$, $\sim_{\text{FB:ps}:\ell_{F,w}}$, $\sim_{\text{RB}:\ell_{R,w}}$, $\sim_{\text{FRB:ps}:\ell_{FR,w}}$ are the six resulting equivalences.
- When do they coincide with the six congruences?
- What is the discriminating information that is needed by reverse and forward-reverse semantics?

- As already anticipated $\sim_{\text{FB:ps}:\ell_F} = \sim_{\text{FB:ps}}$ and $\approx_{\text{FB:ps}:\ell_{F,w}} = \approx_{\text{FB:ps}}$ when $\ell_F(\theta) = \ell_{F,w}(\theta) = \text{act}(\theta)$.
- Expansion law for $\sim_{\text{FB:ps}}$ and $\approx_{\text{FB:ps}}$:

$$(A_{15}) \quad P_1 \parallel_L P_2 = [a^\dagger.] \left(\sum_{i \in I_1, a_{1,i} \notin L} a_{1,i} \cdot (P_{1,i} \parallel_L P'_2) + \sum_{i \in I_2, a_{2,i} \notin L} a_{2,i} \cdot (P'_1 \parallel_L P_{2,i}) + \sum_{i \in I_1, a_{1,i} \in L} \sum_{j \in I_2, a_{2,j} = a_{1,i}} a_{1,i} \cdot (P_{1,i} \parallel_L P_{2,j}) \right)$$

- $P_k = [a_k^\dagger.]P'_k$ with $P'_k = \sum_{i \in I_k} a_{k,i} \cdot P_{k,i}$ for $k \in \{1, 2\}$, called **F-nf**.
- $[a^\dagger.]$ is present iff $[a_1^\dagger.]$ or $[a_2^\dagger.]$ is present (they are optional).

- $\sim_{\text{RB}:\ell_{\text{R}}} = \sim_{\text{RB}}$ and $\sim_{\text{FRB}:\ell_{\text{FR}}} = \sim_{\text{FRB}}$ when $\ell_{\text{R}}(\theta)_{P'} = \ell_{\text{FR}}(\theta)_{P'} = \langle \text{act}(\theta), \text{brs}(P') \rangle \triangleq \ell_{\text{brs}}(\theta)_{P'}$ for every proved transition $P \xrightarrow{\theta} P'$.
- $\text{brs}(P')$ is the **backward ready set** of P' , the set of actions labeling the incoming transitions of P' .

- $\sim_{\text{RB}:\ell_{\text{R}}} = \sim_{\text{RB}}$ and $\sim_{\text{FRB}:\ell_{\text{FR}}} = \sim_{\text{FRB}}$ when $\ell_{\text{R}}(\theta)_{P'} = \ell_{\text{FR}}(\theta)_{P'} = \langle \text{act}(\theta), \text{brs}(P') \rangle \triangleq \ell_{\text{brs}}(\theta)_{P'}$ for every proved transition $P \xrightarrow{\theta} P'$.
- $\text{brs}(P')$ is the **backward ready set** of P' , the set of actions labeling the incoming transitions of P' .
- Thus $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ is encoded as:
 $\langle a, \{a\} \rangle . \langle b, \{a, b\} \rangle . \underline{0} + \langle b, \{b\} \rangle . \langle a, \{a, b\} \rangle . \underline{0}$
 while $a.b.\underline{0} + b.a.\underline{0}$ is encoded as:
 $\langle a, \{a\} \rangle . \langle b, \{b\} \rangle . \underline{0} + \langle b, \{b\} \rangle . \langle a, \{a\} \rangle . \underline{0}$

- $\sim_{\text{RB}:\ell_{\text{R}}} = \sim_{\text{RB}}$ and $\sim_{\text{FRB}:\ell_{\text{FR}}} = \sim_{\text{FRB}}$ when $\ell_{\text{R}}(\theta)_{P'} = \ell_{\text{FR}}(\theta)_{P'} = \langle \text{act}(\theta), \text{brs}(P') \rangle \triangleq \ell_{\text{brs}}(\theta)_{P'}$ for every proved transition $P \xrightarrow{\theta} P'$.
- $\text{brs}(P')$ is the **backward ready set** of P' , the set of actions labeling the incoming transitions of P' .
- Thus $a.\underline{0} \parallel_{\emptyset} b.\underline{0}$ is encoded as:

$$\langle a, \{a\} \rangle . \langle b, \{a, b\} \rangle . \underline{0} + \langle b, \{b\} \rangle . \langle a, \{a, b\} \rangle . \underline{0}$$
while $a.b.\underline{0} + b.a.\underline{0}$ is encoded as:

$$\langle a, \{a\} \rangle . \langle b, \{b\} \rangle . \underline{0} + \langle b, \{b\} \rangle . \langle a, \{a\} \rangle . \underline{0}$$
- The encoding of $a^{\dagger}.\underline{0} \parallel_{\emptyset} b^{\dagger}.\underline{0}$ (a case not addressed in [DP92]) *cannot be*:

$$\langle a^{\dagger}, \{a\} \rangle . \langle b^{\dagger}, \{a, b\} \rangle . \underline{0} + \langle b^{\dagger}, \{b\} \rangle . \langle a^{\dagger}, \{a, b\} \rangle . \underline{0}$$
- It is $\langle a^{\dagger}, \{a\} \rangle . \langle b^{\dagger}, \{a, b\} \rangle . \underline{0} + \langle b, \{b\} \rangle . \langle a, \{a, b\} \rangle . \underline{0}$
or $\langle a, \{a\} \rangle . \langle b, \{a, b\} \rangle . \underline{0} + \langle b^{\dagger}, \{b\} \rangle . \langle a^{\dagger}, \{a, b\} \rangle . \underline{0}$
depending on whether trace $a b$ or trace $b a$ has been executed (initial subprocesses are needed by the forward-reverse semantics).

- Encoding to \mathbb{P}_{brs} : set of sequential processes in which every action prefix is a pair composed of an action and an action set.
- Let \widetilde{P} be the ℓ_{brs} -encoding of P .
- Let \widehat{P} be the $\ell_{\text{brs},w}$ -encoding of P .

- Encoding to \mathbb{P}_{brs} : set of sequential processes in which every action prefix is a pair composed of an action and an action set.
- Let \widetilde{P} be the ℓ_{brs} -encoding of P .
- Let \widehat{P} be the $\ell_{\text{brs},w}$ -encoding of P .
- Expansion laws for \sim_{RB} and \approx_{RB} :

$$\begin{array}{l}
 (\text{A}_{16}) \quad \widetilde{P_1 \parallel_L P_2} = el_{\text{brs},R}^{\varepsilon}(\widetilde{P_1}, \widetilde{P_2}, L)_{P_1 \parallel_L P_2} \\
 (\text{A}_{17}) \quad \widehat{P_1 \parallel_L P_2} = el_{\text{brs},R}^{\varepsilon}(\widehat{P_1}, \widehat{P_2}, L)_{P_1 \parallel_L P_2}
 \end{array}$$

- $P_k = \underline{0}$ or $P_k = a^{\dagger}.P'_k$ for $k \in \{1, 2\}$, called **R-nf**.

- Encoding to \mathbb{P}_{brs} : set of sequential processes in which every action prefix is a pair composed of an action and an action set.
- Let \widetilde{P} be the ℓ_{brs} -encoding of P .
- Let \widehat{P} be the $\ell_{\text{brs},w}$ -encoding of P .
- Expansion laws for \sim_{RB} and \approx_{RB} :

$$\begin{array}{l} \text{(A}_{16}\text{)} \quad \widetilde{P_1 \parallel_L P_2} = el_{\text{brs},R}^\varepsilon(\widetilde{P_1}, \widetilde{P_2}, L)_{P_1 \parallel_L P_2} \\ \text{(A}_{17}\text{)} \quad \widehat{P_1 \parallel_L P_2} = el_{\text{brs},R}^\varepsilon(\widehat{P_1}, \widehat{P_2}, L)_{P_1 \parallel_L P_2} \end{array}$$

- $P_k = \underline{0}$ or $P_k = a^\dagger . P'_k$ for $k \in \{1, 2\}$, called **R-nf**.
- Expansion laws for \sim_{FRB} and $\approx_{\text{FRB};\text{ps}}$:

$$\begin{array}{l} \text{(A}_{18}\text{)} \quad \widetilde{P_1 \parallel_L P_2} = el_{\text{brs}}^\varepsilon(\widetilde{P_1}, \widetilde{P_2}, L)_{P_1 \parallel_L P_2} \\ \text{(A}_{19}\text{)} \quad \widehat{P_1 \parallel_L P_2} = el_{\text{brs}}^\varepsilon(\widehat{P_1}, \widehat{P_2}, L)_{P_1 \parallel_L P_2} \end{array}$$

- $P_k = [a^\dagger . P'_k +] \sum_{i \in I_k} a_{k,i} . P_{k,i}$ for $k \in \{1, 2\}$, called **FR-nf**.

True Concurrency

- How close is \sim_{FRB} to hereditary history-preserving bisimilarity?

True Concurrency

- How close is \sim_{FRB} to hereditary history-preserving bisimilarity?
- A labeled **configuration structure** is a tuple $C = (\mathcal{E}, \mathcal{C}, \ell)$ where:
 - \mathcal{E} is a set of events.
 - $\mathcal{C} \subseteq \mathcal{P}_{\text{fin}}(\mathcal{E})$ is a set of configurations.
 - $\ell : \bigcup_{X \in \mathcal{C}} X \rightarrow \mathcal{A}$ is the labeling function.

True Concurrency

- How close is \sim_{FRB} to hereditary history-preserving bisimilarity?
- A labeled **configuration structure** is a tuple $C = (\mathcal{E}, \mathcal{C}, \ell)$ where:
 - \mathcal{E} is a set of events.
 - $\mathcal{C} \subseteq \mathcal{P}_{\text{fin}}(\mathcal{E})$ is a set of configurations.
 - $\ell : \bigcup_{X \in \mathcal{C}} X \rightarrow \mathcal{A}$ is the labeling function.
- A configuration structure C is **stable** iff it is:
 - Rooted: $\emptyset \in \mathcal{C}$.
 - Connected: $\forall X \in \mathcal{C} \setminus \{\emptyset\}. \exists e \in X. X \setminus \{e\} \in \mathcal{C}$.
 - Closed under bounded unions and intersections:
$$\forall X, Y, Z \in \mathcal{C}. X \cup Y \subseteq Z \implies X \cup Y \in \mathcal{C} \wedge X \cap Y \in \mathcal{C}.$$

True Concurrency

- How close is \sim_{FRB} to hereditary history-preserving bisimilarity?
- A labeled **configuration structure** is a tuple $C = (\mathcal{E}, \mathcal{C}, \ell)$ where:
 - \mathcal{E} is a set of events.
 - $\mathcal{C} \subseteq \mathcal{P}_{\text{fin}}(\mathcal{E})$ is a set of configurations.
 - $\ell : \bigcup_{X \in \mathcal{C}} X \rightarrow \mathcal{A}$ is the labeling function.
- A configuration structure C is **stable** iff it is:
 - Rooted: $\emptyset \in \mathcal{C}$.
 - Connected: $\forall X \in \mathcal{C} \setminus \{\emptyset\}. \exists e \in X. X \setminus \{e\} \in \mathcal{C}$.
 - Closed under bounded unions and intersections:
$$\forall X, Y, Z \in \mathcal{C}. X \cup Y \subseteq Z \implies X \cup Y \in \mathcal{C} \wedge X \cap Y \in \mathcal{C}.$$
- The **causality relation** over $X \in \mathcal{C}$ is defined by letting $e_1 <_X e_2$ for $e_1, e_2 \in X$ s.t. $e_1 \neq e_2$ iff $\forall Y \in \mathcal{C}. Y \subseteq X \wedge e_2 \in Y \implies e_1 \in Y$.
- The **concurrency relation** over X is $co_X = (X \times X) \setminus (\leq_X \cup \geq_X)$.

True Concurrency

- How close is \sim_{FRB} to hereditary history-preserving bisimilarity?
- A labeled **configuration structure** is a tuple $C = (\mathcal{E}, \mathcal{C}, \ell)$ where:
 - \mathcal{E} is a set of events.
 - $\mathcal{C} \subseteq \mathcal{P}_{\text{fin}}(\mathcal{E})$ is a set of configurations.
 - $\ell : \bigcup_{X \in \mathcal{C}} X \rightarrow \mathcal{A}$ is the labeling function.
- A configuration structure C is **stable** iff it is:
 - Rooted: $\emptyset \in \mathcal{C}$.
 - Connected: $\forall X \in \mathcal{C} \setminus \{\emptyset\}. \exists e \in X. X \setminus \{e\} \in \mathcal{C}$.
 - Closed under bounded unions and intersections:
$$\forall X, Y, Z \in \mathcal{C}. X \cup Y \subseteq Z \implies X \cup Y \in \mathcal{C} \wedge X \cap Y \in \mathcal{C}.$$
- The **causality relation** over $X \in \mathcal{C}$ is defined by letting $e_1 <_X e_2$ for $e_1, e_2 \in X$ s.t. $e_1 \neq e_2$ iff $\forall Y \in \mathcal{C}. Y \subseteq X \wedge e_2 \in Y \implies e_1 \in Y$.
- The **concurrency relation** over X is $co_X = (X \times X) \setminus (\leq_X \cup \geq_X)$.
- $X \xrightarrow{a}_C X'$ for $X, X' \in \mathcal{C}$ iff $X \subseteq X' \wedge X' \setminus X = \{e\} \wedge \ell(e) = a$.

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

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

- Two stable configuration structures $C_i = (\mathcal{E}_i, \mathcal{C}_i, l_i)$, $i \in \{1, 2\}$, are **hereditary history-preserving bisimilar**, written $C_1 \sim_{\text{HHPB}} C_2$, iff there exists a hereditary history-preserving bisimulation between C_1 and C_2 , i.e., a relation $\mathcal{B} \subseteq \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{P}(\mathcal{E}_1 \times \mathcal{E}_2)$ such that:
 - $(\emptyset, \emptyset, \emptyset) \in \mathcal{B}$.
 - Whenever $(X_1, X_2, f) \in \mathcal{B}$, then:
 - f is a *bijection* from X_1 to X_2 that preserves *labeling*, i.e., $l_1(e) = l_2(f(e))$ for all $e \in X_1$, and *causality*, i.e., $e \leq_{X_1} e' \iff f(e) \leq_{X_2} f(e')$ for all $e, e' \in X_1$.

- Two stable configuration structures $C_i = (\mathcal{E}_i, \mathcal{C}_i, l_i)$, $i \in \{1, 2\}$, are **hereditary history-preserving bisimilar**, written $C_1 \sim_{\text{HHPB}} C_2$, iff there exists a hereditary history-preserving bisimulation between C_1 and C_2 , i.e., a relation $\mathcal{B} \subseteq \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{P}(\mathcal{E}_1 \times \mathcal{E}_2)$ such that:
 - $(\emptyset, \emptyset, \emptyset) \in \mathcal{B}$.
 - Whenever $(X_1, X_2, f) \in \mathcal{B}$, then:
 - f is a *bijection* from X_1 to X_2 that preserves *labeling*, i.e., $l_1(e) = l_2(f(e))$ for all $e \in X_1$, and *causality*, i.e., $e \leq_{X_1} e' \iff f(e) \leq_{X_2} f(e')$ for all $e, e' \in X_1$.
 - For each $X_1 \xrightarrow{a}_{C_1} X'_1$ there exist $X_2 \xrightarrow{a}_{C_2} X'_2$ and f' such that $(X'_1, X'_2, f') \in \mathcal{B}$ and $f' \upharpoonright X_1 = f$, and vice versa.

- Two stable configuration structures $C_i = (\mathcal{E}_i, \mathcal{C}_i, l_i)$, $i \in \{1, 2\}$, are **hereditary history-preserving bisimilar**, written $C_1 \sim_{\text{HHPB}} C_2$, iff there exists a hereditary history-preserving bisimulation between C_1 and C_2 , i.e., a relation $\mathcal{B} \subseteq \mathcal{C}_1 \times \mathcal{C}_2 \times \mathcal{P}(\mathcal{E}_1 \times \mathcal{E}_2)$ such that:
 - $(\emptyset, \emptyset, \emptyset) \in \mathcal{B}$.
 - Whenever $(X_1, X_2, f) \in \mathcal{B}$, then:
 - f is a *bijection* from X_1 to X_2 that preserves *labeling*, i.e., $l_1(e) = l_2(f(e))$ for all $e \in X_1$, and *causality*, i.e., $e \leq_{X_1} e' \iff f(e) \leq_{X_2} f(e')$ for all $e, e' \in X_1$.
 - For each $X_1 \xrightarrow{a}_{c_1} X'_1$ there exist $X_2 \xrightarrow{a}_{c_2} X'_2$ and f' such that $(X'_1, X'_2, f') \in \mathcal{B}$ and $f' \upharpoonright X_1 = f$, and vice versa.
 - For each $X'_1 \xrightarrow{a}_{c_1} X_1$ there exist $X'_2 \xrightarrow{a}_{c_2} X_2$ and f' such that $(X'_1, X'_2, f') \in \mathcal{B}$ and $f \upharpoonright X'_1 = f'$, and vice versa.

- \sim_{HHPB} [Bednarczyk91] is the finest truly concurrent equivalence preserved under action refinement that is capable of respecting causality, branching, and their interplay while abstracting from choices between identical alternatives [VanGlabbeekGoltz01].
- \sim_{FRB} coincides with \sim_{HHPB} in the absence of autoconcurrency at the same causality level [PhillipsUlidowski12].
- Cross fertilization for their equational and logical characterizations.

- \sim_{HHPB} [Bednarczyk91] is the finest truly concurrent equivalence preserved under action refinement that is capable of respecting causality, branching, and their interplay while abstracting from choices between identical alternatives [VanGlabbeekGoltz01].
- \sim_{FRB} coincides with \sim_{HHPB} in the absence of autoconcurrency at the same causality level [PhillipsUlidowski12].
- Cross fertilization for their equational and logical characterizations.
- Autoconcurrency is $a.\underline{0} \parallel_{\emptyset} a.\underline{0}$, while $a.a.\underline{0}$ is autocaustion.
- $a.\underline{0} \parallel_{\emptyset} a.\underline{0} \sim_{\text{FRB}} a.a.\underline{0} + a.a.\underline{0} \sim_{\text{FRB}} a.a.\underline{0}$.
- Their ℓ_{brs} -encodings are basically the same:

$$\begin{aligned} &\langle a, \{a\} \rangle . \langle a, \{a, a\} \rangle . \underline{0} + \langle a, \{a\} \rangle . \langle a, \{a, a\} \rangle . \underline{0} \\ &\langle a, \{a\} \rangle . \langle a, \{a\} \rangle . \underline{0} + \langle a, \{a\} \rangle . \langle a, \{a\} \rangle . \underline{0} \\ &\langle a, \{a\} \rangle . \langle a, \{a\} \rangle . \underline{0} \end{aligned}$$

- Denotational semantics $\llbracket _ \rrbracket$ for \mathbb{P} based on configuration structures in which events are proof terms.
- $\llbracket a . \underline{0} \parallel_{\emptyset} a . \underline{0} \rrbracket \not\sim_{\text{HHPB}} \llbracket a . a . \underline{0} \rrbracket$ as $\llbracket_{\emptyset} a$ and $\llbracket_{\emptyset} a$ are independent while a and $._a a$ are causally related, hence no bijection exists between the former and the latter that preserves causality.

- Denotational semantics $\llbracket _ \rrbracket$ for \mathbb{P} based on configuration structures in which events are proof terms.
- $\llbracket a . \underline{0} \parallel_{\emptyset} a . \underline{0} \rrbracket \not\sim_{\text{HHPB}} \llbracket a . a . \underline{0} \rrbracket$ as $\llbracket_{\emptyset} a$ and $\llbracket_{\emptyset} a$ are independent while a and $.a$ are causally related, hence no bijection exists between the former and the latter that preserves causality.
- \sim_{FRB} plus **backward ready multiset equality** distinguish them.
- $\sim_{\text{FRB:brm}} = \sim_{\text{HHPB}}$ in the presence of autoconcurrency if for each set of conflicting events all those events are caused by the same event.
- $\sim_{\text{FRB:brm}}$ *counts* the incoming a -transitions of related configurations, no bijection between identically labeled events [AubertCristescu20].

- Denotational semantics $\llbracket _ \rrbracket$ for \mathbb{P} based on configuration structures in which events are proof terms.
- $\llbracket a . \underline{0} \parallel_{\emptyset} a . \underline{0} \rrbracket \not\sim_{\text{HHPB}} \llbracket a . a . \underline{0} \rrbracket$ as $\llbracket_{\emptyset} a$ and $\llbracket_{\emptyset} a$ are independent while a and $.a$ are causally related, hence no bijection exists between the former and the latter that preserves causality.
- \sim_{FRB} plus **backward ready multiset equality** distinguish them.
- $\sim_{\text{FRB:brm}} = \sim_{\text{HHPB}}$ in the presence of autoconcurrency if for each set of conflicting events all those events are caused by the same event.
- $\sim_{\text{FRB:brm}}$ counts the incoming a -transitions of related configurations, no bijection between identically labeled events [AubertCristescu20].
- $\sim_{\text{FRB:brm}}$ over \mathbb{P} is an operational representation of \sim_{HHPB} .
- The ℓ_{brm} -encoding of $a . \underline{0} \parallel_{\emptyset} a . \underline{0}$:

$$\langle a, \{a\} \rangle . \langle a, \{a, a\} \rangle . \underline{0} + \langle a, \{a\} \rangle . \langle a, \{a, a\} \rangle . \underline{0}$$
differs from its ℓ_{brs} -encoding:

$$\langle a, \{a\} \rangle . \langle a, \{a, a\} \rangle . \underline{0} + \langle a, \{a\} \rangle . \langle a, \{a, a\} \rangle . \underline{0}$$

Concluding Remarks and Future Work

- Reversibility as a bridge between different worlds (branching-time vs. linear-time, interleaving vs. true concurrency) that retrospectively enlightens concurrency theory:
 - \sim_{FB} is the usual bisimilarity.
 - \sim_{RB} boils down to reverse trace equivalence over \mathbb{P}_{seq} .
 - \approx_{FRB} is branching bisimilarity over \mathbb{P}_{seq} .
 - $\sim_{\text{FRB:brm}}$ is almost hereditary history-preserving bisimilarity over \mathbb{P} .
 - Expansion laws addressing interleaving semantics or true concurrency.

Concluding Remarks and Future Work

- Reversibility as a bridge between different worlds (branching-time vs. linear-time, interleaving vs. true concurrency) that retrospectively enlightens concurrency theory:
 - \sim_{FB} is the usual bisimilarity.
 - \sim_{RB} boils down to reverse trace equivalence over \mathbb{P}_{seq} .
 - \approx_{FRB} is branching bisimilarity over \mathbb{P}_{seq} .
 - $\sim_{\text{FRB:brm}}$ is almost hereditary history-preserving bisimilarity over \mathbb{P} .
 - Expansion laws addressing interleaving semantics or true concurrency.
- Applying noninterference analysis to reversible systems (branching bisimilarity) and extending causal reversibility by construction [PU07]:
 - Probabilistic processes (alternation with nondeterminism).
 - Deterministically timed processes (time additivity/determinism).
 - Stochastically timed processes (ordinary/exact/strict lumpability, causal reversibility implies time reversibility).

Concluding Remarks and Future Work

- Reversibility as a bridge between different worlds (branching-time vs. linear-time, interleaving vs. true concurrency) that retrospectively enlightens concurrency theory:
 - \sim_{FB} is the usual bisimilarity.
 - \sim_{RB} boils down to reverse trace equivalence over \mathbb{P}_{seq} .
 - \approx_{FRB} is branching bisimilarity over \mathbb{P}_{seq} .
 - $\sim_{\text{FRB:brm}}$ is almost hereditary history-preserving bisimilarity over \mathbb{P} .
 - Expansion laws addressing interleaving semantics or true concurrency.
- Applying noninterference analysis to reversible systems (branching bisimilarity) and extending causal reversibility by construction [PU07]:
 - Probabilistic processes (alternation with nondeterminism).
 - Deterministically timed processes (time additivity/determinism).
 - Stochastically timed processes (ordinary/exact/strict lumpability, causal reversibility implies time reversibility).
- When does time reversibility imply causal reversibility?
- What changes when admitting irreversible actions or recursion?
- Underpinning reversible concurrent programming languages?
- Unitary transformations in quantum computing are reversible!

Inspiring References

- [1] R. Landauer,
"Irreversibility and Heat Generation in the Computing Process",
IBM Journal of Research and Development 5:183–191, 1961.
- [2] C.H. Bennett,
"Logical Reversibility of Computation",
IBM Journal of Research and Development 17:525–532, 1973.
- [3] R. De Nicola, U. Montanari, F. Vaandrager,
"Back and Forth Bisimulations",
Proc. of CONCUR 1990.
- [4] G. Boudol, I. Castellani,
"Flow Models of Distributed Computations: Three Equivalent Semantics for CCS",
Information and Computation 114:247–314, 1994.
- [5] V. Danos, J. Krivine,
"Reversible Communicating Systems",
Proc. of CONCUR 2004.
- [6] I. Phillips, I. Ulidowski,
"Reversing Algebraic Process Calculi",
Journal of Logic and Algebraic Programming 73:70–96, 2007.
- [7] I. Lanese, I. Phillips, I. Ulidowski,
"An Axiomatic Theory for Reversible Computation",
ACM Trans. on Computational Logic 25(2):11:1–11:40, 2024.
- [8] F.P. Kelly,
"Reversibility and Stochastic Networks",
John Wiley & Sons, 1979.
- [9] A. Marin, S. Rossi,
"On the Relations between Markov Chain Lumpability and Reversibility",
Acta Informatica 54:447–485, 2017.

- [10] P. Degano, C. Priami,
“*Proved Trees*”,
Proc. of ICALP 1992.
- [11] G. Boudol, I. Castellani,
“*A Non-Interleaving Semantics for CCS Based on Proved Transitions*”,
Fundamenta Informaticae 11:433–452, 1988.
- [12] R.J. van Glabbeek, U. Goltz,
“*Refinement of Actions and Equivalence Notions for Concurrent Systems*”,
Acta Informatica 37:229–327, 2001.
- [13] Ph. Darondeau, P. Degano,
“*Causal Trees: Interleaving + Causality*”,
Proc. of the LITP Spring School on Theoretical Computer Science, 1990.
- [14] G. Boudol, I. Castellani, M. Hennessy, A. Kiehn,
“*A Theory of Processes with Localities*”,
Formal Aspects of Computing 6:165–200, 1994.
- [15] G. Boudol, I. Castellani,
“*Concurrency and Atomicity*”,
Theoretical Computer Science 59:25–84, 1988.
- [16] A.M. Rabinovich, B.A. Trakhtenbrot,
“*Behavior Structures and Nets*”,
Acta Informatica 11:357–404, 1988.
- [17] M.A. Bednarczyk,
“*Hereditary History Preserving Bisimulations or What Is the Power of the Future Perfect in Program Logics*”,
Technical Report, Polish Academy of Sciences, Gdansk, 1991.
- [18] I. Phillips, I. Ulidowski,
“*A Hierarchy of Reverse Bisimulations on Stable Configuration Structures*”,
Mathematical Structures in Computer Science 22:333–372, 2012.
- [19] C. Aubert, I. Cristescu,
“*How Reversibility Can Solve Traditional Questions: The Example of Hereditary History-Preserving Bisimulation*”,
Proc. of CONCUR 2020.

Our Contributions

- [1] M. Bernardo, S. Rossi,
"Reverse Bisimilarity vs. Forward Bisimilarity",
Proc. of FOSSACS 2023.
- [2] M. Bernardo, A. Esposito,
"On the Weak Continuation of Reverse Bisimilarity vs. Forward Bisimilarity",
Proc. of ICTCS 2023.
- [3] M. Bernardo, A. Esposito,
"Modal Logic Characterizations of Forward, Reverse, and Forward-Reverse Bisimilarities",
Proc. of GANDALF 2023.
- [4] M. Bernardo, A. Esposito, C.A. Mezzina,
"Expansion Laws for Forward-Reverse, Forward, and Reverse Bisimilarities via Proved Encodings",
Proc. of EXPRESS/SOS 2024.
- [5] M. Bernardo, A. Esposito, C.A. Mezzina,
"Alternative Characterizations of Hereditary History-Preserving Bisimilarity via Backward Ready Multisets",
Proc. of FOSSACS 2025.

- [6] A. Esposito, A. Aldini, M. Bernardo,
"Branching Bisimulation Semantics Enables Noninterference Analysis of Reversible Systems",
Proc. of FORTE 2023.
- [7] A. Esposito, A. Aldini, M. Bernardo, S. Rossi,
"Noninterference Analysis of Reversible Systems: An Approach Based on Branching Bisimilarity",
Logical Methods in Computer Science 21(1):6:1–6:28, 2025.
- [8] A. Esposito, A. Aldini, M. Bernardo,
"Noninterference Analysis of Reversible Probabilistic Systems",
Proc. of FORTE 2024.
- [9] A. Esposito, A. Aldini, M. Bernardo,
"Noninterference Analysis of Stochastically Timed Reversible Systems",
Proc. of FORTE 2025.
- [10] A. Esposito, A. Aldini, M. Bernardo,
"Noninterference Analysis of Deterministically Timed Reversible Systems",
Proc. of QEST+FORMATS 2025.
- [11] A. Esposito,
"A Process Algebraic Theory of Reversible Concurrent Systems with Applications to Noninterference Analysis",
Ph.D. Thesis, University of Urbino, 2025.

- [12] M. Bernardo, C.A. Mezzina,
"Towards Bridging Time and Causal Reversibility",
Proc. of FORTE 2020.
- [13] M. Bernardo, C.A. Mezzina,
"Bridging Causal Reversibility and Time Reversibility: A Stochastic Process Algebraic Approach",
Logical Methods in Computer Science 19(2):6:1–6:27, 2023.
- [14] M. Bernardo, I. Lanese, A. Marin, C.A. Mezzina, S. Rossi, C. Sacerdoti Coen,
"Causal Reversibility Implies Time Reversibility",
Proc. of QEST 2023.
- [15] M. Bernardo, C.A. Mezzina,
"Causal Reversibility for Timed Process Calculi with Lazy/Eager Durationless Actions and Time Additivity",
Proc. of FORMATS 2023.
- [16] M. Bernardo, C.A. Mezzina,
"Reversibility in Process Calculi with Nondeterminism and Probabilities",
Proc. of ICTAC 2024.
- [17] M. Bernardo, C.A. Mezzina, A. Esposito,
"Causal Reversibility in Nondeterministic Process Calculi Extended with Time or Probabilities",
Theoretical Computer Science 1063:115646, 2026.