2. Bisimulation 101

Lecture on Models of Concurrent Systems

(Summer 2022)

Stephan Mennicke

Apr 12-20, 2022

What is up next?

Week 1: A Primer in Programming Language Semantics
Week 2: Bisimilarity and Interaction
Week 3: Algebraic Properties of Bisimilarity (SOS needed!)
Week 4: Bisimilarity for Processes with Internal Actions
Week 5: Towards True-Concurrency Semantics
Week 6: ...
Week 7: ...
Week 8: Mobility: The π-Calculus
Week 9: ...
Week 10: ...
Week 11: Advanced Topics: Expressiveness
Week 12: Advances Topics: Expressiveness
Week 13: Advanced Topics: Data-Manipulating Systems
Week 14: Advanced Topics: Data-Manipulating Systems

Labeled Transition System: A Unifying Model

Definition 2.1: A **labeled transition system** (LTS) is a triple $(Pr, Act, \rightarrow)$ where Pr is a non-empty set of **states/processes** (also called the **domain** of the LTS), Act is the set of **labels** (or **actions**), and $\rightarrow \subseteq Pr \times Act \times Pr$ is the **transition relation**.

- Write $P \xrightarrow{\alpha} Q$ for $(P, \alpha, Q) \in \rightarrow$ and call Q the (α-)derivative of P;
  or P performs α and becomes Q.
- If $s = \alpha_1 \alpha_2 \dots \alpha_{k-1}$ for $\alpha_i \in Act$ $(1 \leq i < k)$, then $P \xrightarrow{s} P'$ if there are $P_0, P_1, \dots, P_{k-1}, P_k$ such that $P_{j-1} \xrightarrow{\alpha_j} P_j$ $(0 < j \leq k)$, $P = P_0$, and $P' = P_k$.
- Write $P \xrightarrow{\alpha}$ if there is a P' with $P \xrightarrow{\alpha} P'$ and $P \not\xrightarrow{\alpha}$ if there is no such P'.
- The same notion carries over to sequences of actions $s \in Act^*$.

LTSs and Processes: Further Notation and Classes

If $\mathcal{L} = (Pr, Act, \rightarrow)$ is an LTS, then every process $P \in Pr$ describes an LTS by its own, namely $(\mathbf{P}, Act, \rightarrow_P)$ with $\mathbf{P}$ being the smallest subset of Pr, such that

- $P \in \mathbf{P}$ and
- if $Q \in \mathbf{P}$ and $Q \xrightarrow{\alpha} Q'$, then $Q' \in \mathbf{P}$ and $Q \xrightarrow{\alpha}_P Q'$.

Definition 2.2: An LTS $(Pr, Act, \rightarrow)$ is

1. **image-finite** if for each $P \in Pr$ and $\alpha \in Act$, $\{P' \in Pr \mid P \xrightarrow{\alpha} P'\}$ is finite;
2. **finite-state** if Pr is finite;
3. **finite** if it is finite-state and $\rightarrow$ is acyclic;
4. **deterministic** if for each $P \in Pr$ and $\alpha \in Act$, $P \xrightarrow{\alpha} P'$ and $P \xrightarrow{\alpha} P''$ implies $P' = P''$.

Notions carry over to processes $P \in Pr$.

Equivalence of Processes (1/2)

We call a binary relation on the states of an LTS a **process relation**.

We subsequently assume a single "global" LTS $\mathcal{L} = (Pr, Act, \rightarrow)$.

Every LTS has a natural graph representation, called the **process graph**, interpreting Pr as the set of nodes, Act is the set of edge labels, and $\rightarrow$ is the labeled edge relation.

Definition 2.3: Let $G_i = (V_i, \Sigma, E_i)$ be two edge-labeled directed graphs (V_i and Σ are disjoint sets, $E_i \subseteq V_i \times \Sigma \times V_i$). A bijective function $f : V_1 \rightarrow V_2$ is called an **isomorphism between G_1 and G_2** if $(v, a, w) \in E_1$ if, and only if, $(f(v), a, f(w)) \in E_2$.
Two processes P and Q are **equivalent up to isomorphisms**, denoted by $P \cong Q$, if, and only if, there is an isomorphism between the process graphs of P and Q.

Note, functions $f : A \rightarrow B$ are ultimately relations since $f = \{(x, y) \mid f(x) = y\}$.

Equivalence of Processes (1.5/2)

Comparison LTS to NFAs:

- set of states
- alphabet
- initial state
- final states
- transition function

Definition 2.4: For process P, a **trace of** P is a sequence of actions $w \in Act^*$, such that $P \xrightarrow{w}$. The set of all traces of process P is denoted by $\mathsf{Tr}(P)$. Two processes P and Q are **trace-equivalent**, denoted by $P \sim_{\mathsf{Tr}} Q$, if, and only if, $\mathsf{Tr}(P) = \mathsf{Tr}(Q)$.

Equivalence of Processes (2/2)

Definition 2.5: A process relation $\mathcal{R}$ is called a **bisimulation** if, and only if, $(P, Q) \in \mathcal{R}$ implies for all $\alpha \in Act$,

- for all $P' \in Pr$ with $P \xrightarrow{\alpha} P'$, there is a $Q' \in Pr$ with $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{R}$, and
- for all $Q' \in Pr$ with $Q \xrightarrow{\alpha} Q'$, there is a $P' \in Pr$ with $P \xrightarrow{\alpha} P'$ and $(P', Q') \in \mathcal{R}$.

If there is a bisimulation $\mathcal{R}$ with $(P, Q) \in \mathcal{R}$ we say that P **is bisimilar to** Q, denoted $P \leftrightarrow Q$. $\leftrightarrow$ is called **bisimilarity**.

Hence, $\leftrightarrow$ is the union of all bisimulations.

Theorem 2.6: (I) Bisimilarity is reflexive, symmetric, and transitive. (II) Bisimilarity is itself a bisimulation. (III) Bisimilarity is the largest bisimulation.

“Alternative” Notions/Definitions

A process relation $\mathcal{R}$ is a **noitalumisid** if, and only if, $(P, Q) \in \mathcal{R}$ implies for all $\alpha \in Act$,

- for all P' with $P \xrightarrow{\alpha} P'$, and for all Q' with $Q \xrightarrow{\alpha} Q'$, it holds that $(P', Q') \in \mathcal{R}$, and
- for all Q' with $Q \xrightarrow{\alpha} Q'$, and for all P' with $P \xrightarrow{\alpha} P'$, it holds that $(P', Q') \in \mathcal{R}$.

Discuss noitalumisid in light of bisimulation.

A process relation $\mathcal{S}$ is a **simulation** if, and only if, $(P, Q) \in \mathcal{S}$ implies for all $\alpha \in Act$,

- if $P \xrightarrow{\alpha} P'$, then there is a $Q' \in Pr$ with $Q \xrightarrow{\alpha} Q'$ and $(P', Q') \in \mathcal{S}$.

We say that Q simulates P if there is a simulation $\mathcal{S}$ with $(P, Q) \in \mathcal{S}$, denoted $P \preceq Q$.
P and Q are **simulation equivalent** if, and only if, $P \preceq Q$ and $Q \preceq P$.

Compare similarity ($\preceq$) to bisimilarity ($\leftrightarroweq$). Are two simulation equivalent processes also bisimilar?

Final Remarks

Bisimilarity ($\leftrightarrow$) is the largest bisimulation (Theorem 2.6):

Bisimilarity is the largest process relation, such that for each $P \leftrightarrow Q$ and label $\alpha \in Act$,

1. if $P \xrightarrow{\alpha} P'$, then there is a $Q' \in Pr$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \leftrightarrow Q'$, and
2. if $Q \xrightarrow{\alpha} Q'$, then there is a $P' \in Pr$ such that $P \xrightarrow{\alpha} P'$ and $P' \leftrightarrow Q'$.

To show that $P \leftrightarrow Q$ (i. e., $(P, Q) \in \leftrightarrow$), it is sufficient to give a bisimulation $\mathcal{R}$ such that $(P, Q) \in \mathcal{R}$.

An inductive definition of process equality:

$P = Q$ if, for all α:

1. for all P' with $P \xrightarrow{\alpha} P'$, there is a Q' such that $Q \xrightarrow{\alpha} Q'$ and $P' = Q'$, and
2. for all Q' with $Q \xrightarrow{\alpha} Q'$, there is a P' such that $P \xrightarrow{\alpha} P'$ and $P' = Q'$.

What about Interaction? Testing!

- As before, we consider a single LTS $(Pr, Act, \rightarrow)$.
- Additionally, we'll assume image-finiteness for the transition system.
- For tests T and processes P we have a look at observations

$$\mathcal{O}(T, P) \subseteq \{\top, \bot\}$$

- Testing scenario: very simple, only success and failure (absence of success)
- Recall, $P \stackrel{a}{\nrightarrow}$ means there is no P' with $P \xrightarrow{a} P'$

Testing: Syntax and Semantics

A test T is an expression of the following grammar:

$$T \quad ::= \quad \mathsf{SUCC} \mid \mathsf{FAIL} \mid a.T \mid \tilde{a}.T \mid T \wedge T \mid T \vee T \mid \forall T \mid \exists T$$

For an arbitrary process P and test T, define the observations admitted by P through T as:

$$\begin{array}{rcl}
\mathcal{O}(\mathsf{SUCC}, P) & = & \{\top\} \\
\mathcal{O}(\mathsf{FAIL}, P) & = & \{\bot\} \\
\mathcal{O}(a.T, P) & = & \begin{cases} \{\bot\} & \text{if } P \stackrel{a}{\not\rightarrow} \\ \bigcup\{\mathcal{O}(T, P') \mid P \stackrel{a}{\rightarrow} P'\} & \text{otherwise.} \end{cases} \\
\mathcal{O}(\tilde{a}.T, P) & = & \begin{cases} \{\top\} & \text{if } P \stackrel{a}{\not\rightarrow} \\ \bigcup\{\mathcal{O}(T, P') \mid P \stackrel{a}{\rightarrow} P'\} & \text{otherwise.} \end{cases} \\
\mathcal{O}(T_1 \wedge T_2, P) & = & \mathcal{O}(T_1, P) \wedge^\star \mathcal{O}(T_2, P) \\
\mathcal{O}(T_1 \vee T_2, P) & = & \mathcal{O}(T_1, P) \vee^\star \mathcal{O}(T_2, P)
\end{array}$$

Testing: Syntax and Semantics

$$T \quad ::= \quad \mathsf{SUCC} \mid \mathsf{FAIL} \mid a.T \mid \tilde{a}.T \mid T \wedge T \mid T \vee T \mid \forall T \mid \exists T$$

$$\begin{aligned}
\mathcal{O}(\mathsf{SUCC}, P) &= \{\top\} \\
\mathcal{O}(\mathsf{FAIL}, P) &= \{\bot\} \\
\mathcal{O}(a.T, P) &= \begin{cases} \{\bot\} & \text{if } P \stackrel{a}{\not\to} \\ \bigcup\{\mathcal{O}(T, P') \mid P \xrightarrow{a} P'\} & \text{otherwise.} \end{cases} \\
\mathcal{O}(\tilde{a}.T, P) &= \begin{cases} \{\top\} & \text{if } P \stackrel{a}{\not\to} \\ \bigcup\{\mathcal{O}(T, P') \mid P \xrightarrow{a} P'\} & \text{otherwise.} \end{cases} \\
\mathcal{O}(T_1 \wedge T_2, P) &= \mathcal{O}(T_1, P) \wedge^\star \mathcal{O}(T_2, P) \\
\mathcal{O}(T_1 \vee T_2, P) &= \mathcal{O}(T_1, P) \vee^\star \mathcal{O}(T_2, P) \\
\mathcal{O}(\forall T, P) &= \begin{cases} \{\bot\} & \text{if } \bot \in \mathcal{O}(T, P) \\ \{\top\} & \text{otherwise} \end{cases} \\
\mathcal{O}(\exists T, P) &= \begin{cases} \{\top\} & \text{if } \top \in \mathcal{O}(T, P) \\ \{\bot\} & \text{otherwise} \end{cases}
\end{aligned}$$

Properties of Tests and Observation (1/)

Theorem 2.7: Every test T has an inverse test $\overline{T}$, such that for all processes P,

1. $\bot \in \mathcal{O}(T, P)$ if, and only if, $\top \in \mathcal{O}(\overline{T}, P)$ and
2. $\top \in \mathcal{O}(T, P)$ if, and only if, $\bot \in \mathcal{O}(\overline{T}, P)$.

Proof (of 1): Define $\overline{T}$ by

$$\begin{array}{rcl@{\qquad}rcl}
\overline{\mathsf{SUCC}} & = & \mathsf{FAIL} & \overline{\mathsf{FAIL}} & = & \mathsf{SUCC} \\
\overline{a.T'} & = & \tilde{a}.\overline{T'} & \overline{\tilde{a}.T'} & = & a.\overline{T'} \\
\overline{T_1 \wedge T_2} & = & \overline{T_1} \vee \overline{T_2} & \overline{T_1 \vee T_2} & = & \overline{T_1} \wedge \overline{T_2} \\
\overline{\exists T'} & = & \forall \overline{T'} & \overline{\forall T'} & = & \exists \overline{T'}
\end{array}$$

Proof by induction on the structure of T. Let P be a process.

Base: $T = \mathsf{FAIL}$. Then $\mathcal{O}(T, P) = \{\bot\}$ and $\mathcal{O}(\overline{T}, P) = \mathcal{O}(\mathsf{SUCC}, P) = \{\top\}$.

Properties of Tests and Observations (2/)

$$\begin{aligned}
\overline{\mathsf{SUCC}} &= \mathsf{FAIL} & \overline{\mathsf{FAIL}} &= \mathsf{SUCC}\\
\overline{a.T'} &= \tilde{a}.\overline{T'} & \overline{\tilde{a}.T'} &= a.\overline{T'}\\
\overline{T_1 \wedge T_2} &= \overline{T_1} \vee \overline{T_2} & \overline{T_1 \vee T_2} &= \overline{T_1} \wedge \overline{T_2}\\
\overline{\exists T'} &= \forall \overline{T'} & \overline{\forall T'} &= \exists \overline{T'}
\end{aligned}$$

Step: By case distinction.

- $T = T_1 \wedge T_2$: $\bot \in \mathcal{O}(T, P)$ iff $\bot \in \mathcal{O}(T_1, P)$ or $\bot \in \mathcal{O}(T_2, P)$ iff(IH) $\top \in \mathcal{O}(\overline{T_1}, P)$ or $\top \in \mathcal{O}(\overline{T_2}, P)$ iff $\top \in \mathcal{O}(\overline{T_1} \vee \overline{T_2}, P)$ iff $\top \in \mathcal{O}(\overline{T}, P)$
- $T = T_1 \vee T_2$: $\bot \in \mathcal{O}(T, P)$ iff $\bot \in \mathcal{O}(T_1, P)$ and $\bot \in \mathcal{O}(T_2, P)$ iff(IH) $\top \in \mathcal{O}(\overline{T_1}, P)$ and $\top \in \mathcal{O}(\overline{T_2}, P)$ iff $\top \in \mathcal{O}(\overline{T_1} \wedge \overline{T_2}, P)$ iff $\top \in \mathcal{O}(\overline{T}, P)$
- $T = \exists T'$: $\bot \in \mathcal{O}(T, P)$ iff $\mathcal{O}(T', P) = \{\bot\}$ iff(IH) $\mathcal{O}(\overline{T'}, P) = \{\top\}$ iff $\top \in \mathcal{O}(\forall \overline{T'}, P)$ iff $\top \in \mathcal{O}(\overline{T}, P)$.
- $T = \forall T'$: $\bot \in \mathcal{O}(T, P)$ iff $\bot \in \mathcal{O}(T', P)$ iff(IH) $\top \in \mathcal{O}(\overline{T'}, P)$ iff $\top \in \mathcal{O}(\exists \overline{T'}, P)$ iff $\top \in \mathcal{O}(\overline{T}, P)$.

Properties of Tests and Observations (3/)

$$
\begin{array}{rclcrcl}
\overline{\mathsf{SUCC}} & = & \mathsf{FAIL} & \quad & \overline{\mathsf{FAIL}} & = & \mathsf{SUCC} \\
\overline{a.T'} & = & \tilde{a}.\overline{T'} & & \overline{\tilde{a}.T'} & = & a.\overline{T'} \\
\overline{T_1 \wedge T_2} & = & \overline{T_1} \vee \overline{T_2} & & \overline{T_1 \vee T_2} & = & \overline{T_1} \wedge \overline{T_2} \\
\overline{\exists T'} & = & \forall \overline{T'} & & \overline{\forall T'} & = & \exists \overline{T'}
\end{array}
$$

Step (cont'd): By case distinction.

- $T = a.T'$: $\bot \in \mathcal{O}(T, P)$ iff (a) $P \stackrel{a}{\not\to}$ or (b) $\bot \in \mathcal{O}(T', P')$ for some P' with $P \xrightarrow{a} P'$. In case (a), $\mathcal{O}(\tilde{a}.\overline{T'}, P) = \{\top\}$. In case (b), $\top \in \mathcal{O}(\overline{T'}, P')$ by IH. Hence, $\top \in \mathcal{O}(\tilde{a}.\overline{T'}, P)$ by the arguments for (a) and (b).
- $T = \tilde{a}.T'$: $\bot \in \mathcal{O}(T, P)$ iff $P \xrightarrow{a} P'$ (for some P') and $\bot \in \mathcal{O}(T', P')$ iff $\top \in \mathcal{O}(\overline{T'}, P')$ iff $\top \in \mathcal{O}(a.\overline{T'}, P)$ iff $\top \in \mathcal{O}(\overline{T}, P)$. $\square$

Properties of Tests and Observation (4/4)

Definition 2.8: $P \sim_T Q$ if, and only if, $\mathcal{O}(T, P) = \mathcal{O}(T, Q)$ for all tests T.

Theorem 2.9: If $P \not\sim_T Q$, then there is a test case T, such that $\mathcal{O}(T, P) = \{\bot\}$ and $\mathcal{O}(T, Q) = \{\top\}$.

Proof: Since $P \not\sim_T Q$, there is at least one test case T_0 with $\mathcal{O}(T_0, P) \neq \mathcal{O}(T_0, Q)$. Transform T_0 into the required T by the following procedure:

1. If $\mathcal{O}(T_0, Q) = \{\top\}$, set $T = \forall T_0$. If $\mathcal{O}(T_0, Q) = \{\bot\}$, set $\mathcal{O}(\forall \overline{T_0})$.
2. Otherwise, if $\mathcal{O}(T_0, P) = \{\bot\}$, set $T = \exists T_0$ and if $\mathcal{O}(T_0, P) = \{\top\}$, set $T = \exists \overline{T_0}$. □

Theorem 2.10: $\leftrightarrow = \sim_T$ on image-finite processes.

Intermezzo: What is a Good Equivalence on Processes?

Completed Traces and Failure Equivalence

Definition 2.11: For process P, a trace $w \in \mathsf{Tr}(P)$ is a **completed trace of** P if for some process P', $P \xrightarrow{w} P'$ and for all $\alpha \in Act$, $P' \stackrel{\alpha}{\not\rightarrow}$. Denote by $\mathsf{CTr}(P)$ the set of all completed traces of P. Process P is **completed trace equivalent to** process Q if, and only if, $P \sim_{\mathsf{Tr}} Q$ and $\mathsf{CTr}(P) = \mathsf{CTr}(Q)$.

Definition 2.12: For process P, $\langle w, X \rangle$ is a **failure pair of** P if $w \in \mathsf{Tr}(P)$ and for some P' with $P \xrightarrow{w} P'$, $P' \stackrel{\alpha}{\not\rightarrow}$ for all $\alpha \in X$. Denote by $\mathsf{F}(P)$ the set of all failure pairs of P. Process P is **failure equivalent** to process Q if, and only if, $\mathsf{F}(P) = \mathsf{F}(Q)$.

Testing Revisited

$$T \quad ::= \quad \mathsf{SUCC} \;\big|\; \mathsf{FAIL} \;\big|\; a.T \;\big|\; \tilde{a}.T \;\big|\; T \wedge T \;\big|\; T \vee T \;\big|\; \forall T \;\big|\; \exists T$$

What if a test is a process itself (i. e., $T \in Pr$)? A special action $\checkmark \in Act$ would signal success of a test.

A **testing configuration** is an expression of the following grammar:

$$E \quad ::= \quad \langle T, P \rangle \;\big|\; \top$$

where $T, P \in Pr$. Define the **testing transition relation** $\Longrightarrow$ as the smallest relation satisfying the following two rules:

$$\text{(ACT)}\ \frac{T \xrightarrow{a} T' \qquad P \xrightarrow{a} P'}{\langle T, P \rangle \Longrightarrow \langle T', P' \rangle} \qquad\qquad \text{(SUCC)}\ \frac{T \xrightarrow{\checkmark} T'}{\langle T, P \rangle \Longrightarrow \top}$$

Testing Preorder and Equivalence

$$\text{(ACT)}\ \frac{T \xrightarrow{a} T' \qquad P \xrightarrow{a} P'}{\langle T, P\rangle \Longrightarrow \langle T', P'\rangle} \qquad \text{(SUCC)}\ \frac{T \xrightarrow{\checkmark} T'}{\langle T, P\rangle \Longrightarrow \top}$$

A (finite or infinite) sequence of testing configurations $E_0 E_1 \dots$ is called a **testing sequence** of process P and test T if $E_0 = \langle T, P\rangle$ and for all $i > 0$, $E_{i-1} \Longrightarrow E_i$.

1. $\top \in \mathcal{O}(T, P)$ if there is a testing sequence emanating from $\langle T, P\rangle$ on which $\top$ occurs;
2. $\bot \in \mathcal{O}(T, P)$ if there is a testing sequence emanating from $\langle T, P\rangle$ on which no $\top$ occurs.

$\mathcal{O}(T, P)$ are non-empty subsets of $\{\top, \bot\}$ (as a lattice $\bot \sqsubseteq \top$).

Lifting for observations: $\{\bot\} \sqsubseteq \{\top, \bot\} \sqsubseteq \{\top\}$.

Definition 2.13: $P \lesssim Q$ if, and only if, for all processes T, $\mathcal{O}(T, P) \sqsubseteq \mathcal{O}(T, Q)$. P and Q are **observational testing equivalent**, denoted $P \simeq Q$, if, and only if, $P \lesssim Q$ and $Q \lesssim P$.

Theorem 2.14: $\simeq$ coincides with failure equivalence.

May/Must Testing (1/2)

Another lifting for observations: $\{\bot\} \sqsubseteq_{\mathsf{may}} \{\top, \bot\} \equiv_{\mathsf{may}} \{\top\}$

Definition 2.15: $P \lesssim_{\mathsf{may}} Q$ if, and only if, for all processes T, $\mathcal{O}(T, P) \sqsubseteq_{\mathsf{may}} \mathcal{O}(T, P)$. P and Q are **may-testing equivalent**, denoted $P \simeq_{\mathsf{may}} Q$, if, and only if, $P \lesssim_{\mathsf{may}} Q$ and $Q \lesssim_{\mathsf{may}} P$.

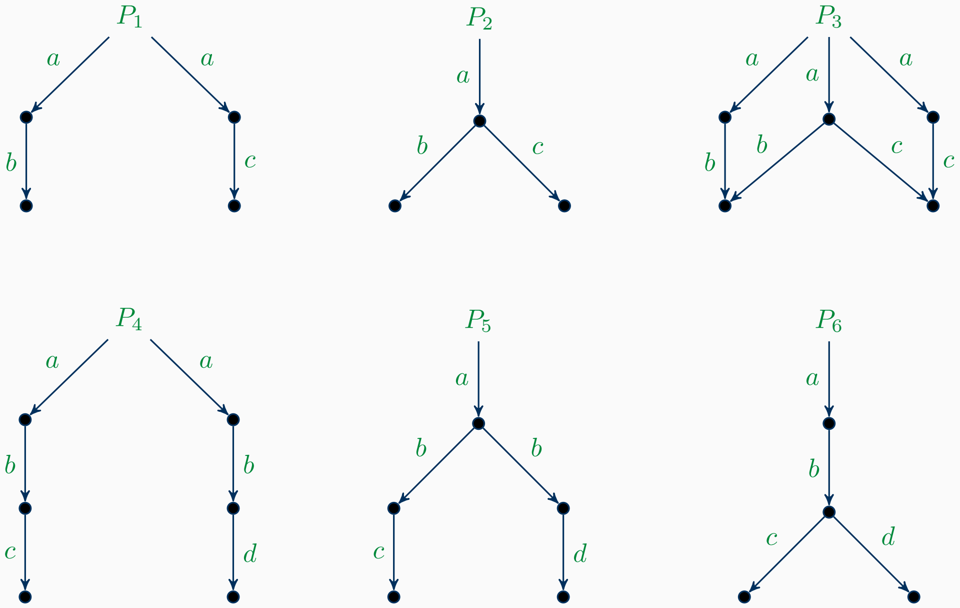
Theorem 2.16: $\simeq_{\mathsf{may}} = \sim_{\mathsf{Tr}}$

May/Must Testing (2/2)

Another lifting for observations: $\{\bot\} \equiv_{\text{must}} \{\top, \bot\} \sqsubseteq_{\text{must}} \{\top\}$

Definition 2.17: $P \lesssim_{\text{must}} Q$ if, and only if, for all processes T, $\mathcal{O}(T, P) \sqsubseteq_{\text{must}} \mathcal{O}(T, P)$. P and Q are **must-testing equivalent**, denoted $P \simeq_{\text{must}} Q$, if, and only if, $P \lesssim_{\text{must}} Q$ and $Q \lesssim_{\text{must}} P$.

Theorem 2.18: (I) $\simeq = \simeq_{\text{may}} \cap \simeq_{\text{must}}$ (II) $\simeq_{\text{must}} \subseteq \simeq_{\text{may}}$ (III) $\simeq_{\text{must}} = \simeq$


P_1
a a b c
P_2
a b c
P_3
a a a b b c c
P_4
a a b b c d
P_5
a b b c d
P_6
a b c d