Concurrency Theory

Winter 2025/26

Lecture 4: Strong Bisimulation

Thomas Noll, Peter Thiemann
Programming Languages Group
University of Freiburg

https://proglang.github.io/teaching/25ws/ct.html

Thomas Noll, Peter Thiemann

Winter 2025/26

Outline of Lecture 4

The Wish List for Behavioural Equivalences

(1) Less distinctive than isomorphism: an equivalence should distinguish less processes than LTS isomorphism does, i.e., $\equiv$ should be coarser than LTS isomorphism:

$$LTS(P) \equiv_{iso} LTS(Q) \Rightarrow P \equiv Q.$$

(2) More distinctive than trace equivalence: an equivalence should distinguish more processes than trace equivalence does, i.e., $\equiv$ should be finer than trace equivalence:

$$P \equiv Q \Rightarrow Tr(P) = Tr(Q).$$

(3) Congruence property: the equivalence must be substitutive with respect to all CCS operators (in the following).

(4) Deadlock preservation: equivalent processes should have the same deadlock behaviour, i.e., they can either both deadlock, or both cannot (in the following).

(5) Optional: the coarsest possible equivalence: there should be no less discriminating equivalence satisfying all these requirements.

Trace Equivalence

Definition (Trace language)

For every $P \in Prc$, let $Tr(P) := \{w \in Act^* \mid \text{ex. } P' \in Prc \text{ such that } P \xrightarrow{w} P'\}$ be the trace language of P (where $\xrightarrow{w} := \xrightarrow{\alpha_1} \circ \ldots \circ \xrightarrow{\alpha_n}$ for $w = \alpha_1 \ldots \alpha_n$).

$P, Q \in Prc$ are called trace equivalent if $Tr(P) = Tr(Q)$.

Trace Equivalence

Definition (Trace language)

For every $P \in Prc$, let $Tr(P) := \{w \in Act^* \mid \text{ex. } P' \in Prc \text{ such that } P \xrightarrow{w} P'\}$ be the trace language of P (where $\xrightarrow{w} := \xrightarrow{\alpha_1} \circ \ldots \circ \xrightarrow{\alpha_n}$ for $w = \alpha_1 \ldots \alpha_n$).

$P, Q \in Prc$ are called trace equivalent if $Tr(P) = Tr(Q)$.

- Trace equivalence is a possible behavioural equivalence, is a congruence, but does not preserve deadlocks.

Trace Equivalence

Definition (Trace language)

For every $P \in Prc$, let $Tr(P) := \{w \in Act^* \mid \text{ex. } P' \in Prc \text{ such that } P \xrightarrow{w} P'\}$ be the trace language of P (where $\xrightarrow{w} := \xrightarrow{\alpha_1} \circ \ldots \circ \xrightarrow{\alpha_n}$ for $w = \alpha_1 \ldots \alpha_n$).

$P, Q \in Prc$ are called trace equivalent if $Tr(P) = Tr(Q)$.

- Trace equivalence is a possible behavioural equivalence, is a congruence, but does not preserve deadlocks.
- Main problem:

$$Tr(\alpha.(P + Q)) = Tr(\alpha.P + \alpha.Q),$$

whereas their deadlock behaviour in a context can differ.

Trace Equivalence

Definition (Trace language)

For every $P \in Prc$, let $Tr(P) := \{w \in Act^* \mid \text{ex. } P' \in Prc \text{ such that } P \xrightarrow{w} P'\}$ be the trace language of P (where $\xrightarrow{w} := \xrightarrow{\alpha_1} \circ \ldots \circ \xrightarrow{\alpha_n}$ for $w = \alpha_1 \ldots \alpha_n$).

$P, Q \in Prc$ are called trace equivalent if $Tr(P) = Tr(Q)$.

- Trace equivalence is a possible behavioural equivalence, is a congruence, but does not preserve deadlocks.
- Main problem:

  $$Tr(\alpha.(P + Q)) = Tr(\alpha.P + \alpha.Q),$$

  whereas their deadlock behaviour in a context can differ.
- Solution: consider finer behavioural equivalences $\equiv$ such that

  $$\alpha.(P + Q) \not\equiv \alpha.P + \alpha.Q.$$

Definition (Trace language)

For every $P \in Prc$, let $Tr(P) := \{w \in Act^* \mid \text{ex. } P' \in Prc \text{ such that } P \xrightarrow{w} P'\}$ be the trace language of P (where $\xrightarrow{w} := \xrightarrow{\alpha_1} \circ \ldots \circ \xrightarrow{\alpha_n}$ for $w = \alpha_1 \ldots \alpha_n$).

$P, Q \in Prc$ are called trace equivalent if $Tr(P) = Tr(Q)$.

- Trace equivalence is a possible behavioural equivalence, is a congruence, but does not preserve deadlocks.
- Main problem:
$$Tr(\alpha.(P+Q)) = Tr(\alpha.P + \alpha.Q),$$
whereas their deadlock behaviour in a context can differ.
- Solution: consider finer behavioural equivalences $\equiv$ such that
$$\alpha.(P+Q) \not\equiv \alpha.P + \alpha.Q.$$
- Our (serious) attempt today: Milner's strong bisimulation.

Robin Milner (1934–2010)

Outline of Lecture 4

Rationale

Observation

In order for a behavioural equivalence to be deadlock sensitive, it has to take the branching structure of processes into account.

Rationale

Observation

In order for a behavioural equivalence to be deadlock sensitive, it has to take the branching structure of processes into account.

This is achieved by an equivalence that is defined according to the following scheme:

Bisimulation scheme

$P, Q \in Prc$ are equivalent iff, for every action α, every α-successor of P is equivalent to some α-successor of Q, and vice versa.

Rationale

Observation

In order for a behavioural equivalence to be deadlock sensitive, it has to take the branching structure of processes into account.

This is achieved by an equivalence that is defined according to the following scheme:

Bisimulation scheme

$P, Q \in Prc$ are equivalent iff, for every action α, every α-successor of P is equivalent to some α-successor of Q, and vice versa.

Three variants will be considered in this course:

(1) Strong bisimulation: ignore the special role of τ-actions
(2) Weak bisimulation: treat τ-actions as invisible
(3) Simulation relations: unidirectional versions of bisimulation

Strong Bisimulation I

Definition 4.1 (Strong bisimulation) (Park 1981, Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation if for every $(P, Q) \in \rho$ and $\alpha \in Act$:

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \, \rho \, Q'$, and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \, \rho \, Q'$.

Strong Bisimulation I

Definition 4.1 (Strong bisimulation) (Park 1981, Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation if for every $(P, Q) \in \rho$ and $\alpha \in Act$:

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \; \rho \; Q'$, and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \; \rho \; Q'$.

Note: strong bisimulations are not necessarily equivalences (e.g., $\rho = \emptyset$).

Strong Bisimulation I

Definition 4.1 (Strong bisimulation) (Park 1981, Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation if for every $(P, Q) \in \rho$ and $\alpha \in Act$:

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \rho Q'$, and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \rho Q'$.

Note: strong bisimulations are not necessarily equivalences (e.g., $\rho = \emptyset$).

Definition 4.2 (Strong bisimilarity)

Processes $P, Q \in Prc$ are strongly bisimilar ($P \sim Q$), iff there is a strong bisimulation ρ with $P \rho Q$.

Strong Bisimulation I

Definition 4.1 (Strong bisimulation) (Park 1981, Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation if for every $(P, Q) \in \rho$ and $\alpha \in Act$:

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$, and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \,\rho\, Q'$.

Note: strong bisimulations are not necessarily equivalences (e.g., $\rho = \emptyset$).

Definition 4.2 (Strong bisimilarity)

Processes $P, Q \in Prc$ are strongly bisimilar ($P \sim Q$), iff there is a strong bisimulation ρ with $P \,\rho\, Q$.

$$\sim \;=\; \bigcup \{\rho \subseteq Prc \times Prc \mid \rho \text{ is a strong bisimulation}\}.$$

Strong Bisimulation I

Definition 4.1 (Strong bisimulation) (Park 1981, Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation if for every $(P, Q) \in \rho$ and $\alpha \in Act$:

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$, and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \,\rho\, Q'$.

Note: strong bisimulations are not necessarily equivalences (e.g., $\rho = \emptyset$).

Definition 4.2 (Strong bisimilarity)

Processes $P, Q \in Prc$ are strongly bisimilar ($P \sim Q$), iff there is a strong bisimulation ρ with $P \,\rho\, Q$.

$$\sim = \bigcup \{\rho \subseteq Prc \times Prc \mid \rho \text{ is a strong bisimulation}\}.$$

Relation $\sim$ is called strong bisimilarity.

Strong Bisimulation II

$$\begin{array}{ccc} P & \xrightarrow{\alpha} & P' \\ \rho & & \\ Q & & \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} P & \xrightarrow{\alpha} & P' \\ \rho & & \rho \\ Q & \xrightarrow{\alpha} & Q' \end{array}$$

Strong Bisimulation II

$$\begin{array}{ccc} P & \xrightarrow{\alpha} & P' \\ \rho & & \\ Q & & \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} P & \xrightarrow{\alpha} & P' \\ \rho & & \rho \\ Q & \xrightarrow{\alpha} & Q' \end{array}$$

and

$$\begin{array}{ccc} P & & \\ \rho & & \\ Q & \xrightarrow{\alpha} & Q' \end{array} \quad \text{can be completed to} \quad \begin{array}{ccc} P & \xrightarrow{\alpha} & P' \\ \rho & & \rho \\ Q & \xrightarrow{\alpha} & Q' \end{array}$$

Examples

Definition 4.3 (Strong bisimulation — recall) (Park 1981, Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation if for every $(P, Q) \in \rho$ and $\alpha \in Act$:

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$, and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \,\rho\, Q'$.

Example 4.4 (A first example)

Claim: $P \sim Q$ where

$$\begin{array}{rclcrcl} P & = & a.P_1 + a.P_2 & \quad & Q & = & a.Q_1 \\ P_1 & = & b.P_2 & & Q_1 & = & b.Q_1 \\ P_2 & = & b.P_2 & & & & \end{array}$$

Examples

Definition 4.3 (Strong bisimulation — recall) (Park 1981, Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation if for every $(P, Q) \in \rho$ and $\alpha \in Act$:

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \;\rho\; Q'$, and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \;\rho\; Q'$.

Example 4.4 (A first example)

Claim: $P \sim Q$ where
$$\begin{array}{rclcrcl} P & = & a.P_1 + a.P_2 & \quad & Q & = & a.Q_1 \\ P_1 & = & b.P_2 & & Q_1 & = & b.Q_1 \\ P_2 & = & b.P_2 & & & & \end{array}$$

Proof: $\rho = \{(P, Q), (P_1, Q_1), (P_2, Q_1)\}$ is a strong bisimulation

Examples

Definition 4.3 (Strong bisimulation — recall) (Park 1981, Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation if for every $(P, Q) \in \rho$ and $\alpha \in Act$:

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$, and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \,\rho\, Q'$.

Example 4.4 (A first example)

Claim: $P \sim Q$ where

$$\begin{array}{rcl@{\qquad}rcl} P & = & a.P_1 + a.P_2 & Q & = & a.Q_1 \\ P_1 & = & b.P_2 & Q_1 & = & b.Q_1 \\ P_2 & = & b.P_2 & & & \end{array}$$

Proof: $\rho = \{(P, Q), (P_1, Q_1), (P_2, Q_1)\}$ is a strong bisimulation

Example 4.5 (Relating a finite to an infinite-state process)

Claim: $P_0 \sim Q$ where $P_i = a.P_{i+1}$ for $i \in \mathbb{N}$ and $Q = a.Q$.

Examples

Definition 4.3 (Strong bisimulation — recall) (Park 1981, Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation if for every $(P, Q) \in \rho$ and $\alpha \in Act$:

(1) if $P \xrightarrow{\alpha} P'$, then there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \; \rho \; Q'$, and

(2) if $Q \xrightarrow{\alpha} Q'$, then there exists $P' \in Prc$ such that $P \xrightarrow{\alpha} P'$ and $P' \; \rho \; Q'$.

Example 4.4 (A first example)

Claim: $P \sim Q$ where

$$\begin{array}{rcl} P & = & a.P_1 + a.P_2 \\ P_1 & = & b.P_2 \\ P_2 & = & b.P_2 \end{array} \qquad \begin{array}{rcl} Q & = & a.Q_1 \\ Q_1 & = & b.Q_1 \end{array}$$

Proof: $\rho = \{(P, Q), (P_1, Q_1), (P_2, Q_1)\}$ is a strong bisimulation

Example 4.5 (Relating a finite to an infinite-state process)

Claim: $P_0 \sim Q$ where $P_i = a.P_{i+1}$ for $i \in \mathbb{N}$ and $Q = a.Q$.

Proof: $\rho = \{(P_i, Q) \mid i \in \mathbb{N}\}$ is a strong bisimulation.

A Counterexample

Example 4.6 (Vending machines; cf. Example 3.13)

Show $CTM \not\sim CTM'$ where
$$\begin{aligned} CTM &= coin.\left(\overline{coffee}.CTM + \overline{tea}.CTM\right) \\ CTM' &= coin.\overline{coffee}.CTM' + coin.\overline{tea}.CTM'. \end{aligned}$$

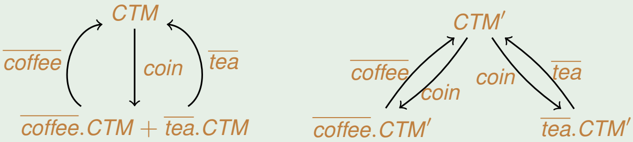
A Counterexample

Example 4.6 (Vending machines; cf. Example 3.13)

Show $CTM \not\sim CTM'$ where

$$\begin{aligned} CTM &= coin.\left(\overline{coffee}.CTM + \overline{tea}.CTM\right) \\ CTM' &= coin.\overline{coffee}.CTM' + coin.\overline{tea}.CTM'. \end{aligned}$$

Corresponding LTSs:



A Counterexample

Example 4.6 (Vending machines; cf. Example 3.13)

Show $CTM \not\sim CTM'$ where
$$\begin{aligned} CTM &= coin.\left(\overline{coffee}.CTM + \overline{tea}.CTM\right) \\ CTM' &= coin.\overline{coffee}.CTM' + coin.\overline{tea}.CTM'. \end{aligned}$$

Corresponding LTSs:

$$\begin{tikzcd}
CTM \arrow[d, "coin"] \\
\overline{coffee}.CTM + \overline{tea}.CTM \arrow[u, bend left=60, "\overline{coffee}"] \arrow[u, bend right=60, "\overline{tea}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
 & CTM' \arrow[dl, bend left, "coin"] \arrow[dr, bend right, "coin"'] & \\
\overline{coffee}.CTM' \arrow[ur, bend left, "\overline{coffee}"] & & \overline{tea}.CTM' \arrow[ul, bend right, "\overline{tea}"']
\end{tikzcd}$$

Assumption: there exists bisimulation ρ such that $CTM \ \rho \ CTM'$.

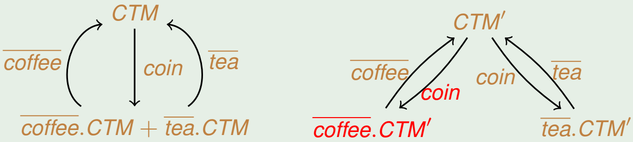
A Counterexample

Example 4.6 (Vending machines; cf. Example 3.13)

Show $CTM \not\sim CTM'$ where

$$\begin{aligned} CTM &= coin.\left(\overline{coffee}.CTM + \overline{tea}.CTM\right) \\ CTM' &= coin.\overline{coffee}.CTM' + coin.\overline{tea}.CTM'. \end{aligned}$$

Corresponding LTSs:



Assumption: there exists bisimulation ρ such that $CTM \; \rho \; CTM'$.

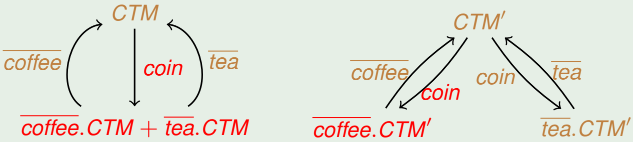
- First CTM' chooses the left *coin*-transition.

A Counterexample

Example 4.6 (Vending machines; cf. Example 3.13)

Show $CTM \not\sim CTM'$ where
$$\begin{aligned} CTM &= coin.\left(\overline{coffee}.CTM + \overline{tea}.CTM\right) \\ CTM' &= coin.\overline{coffee}.CTM' + coin.\overline{tea}.CTM'. \end{aligned}$$

Corresponding LTSs:



Assumption: there exists bisimulation ρ such that $CTM \ \rho \ CTM'$.

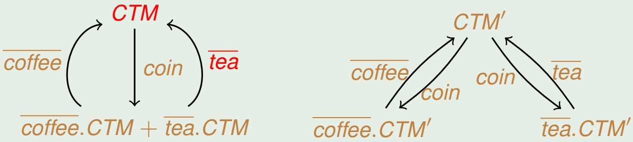
- First CTM' chooses the left $coin$-transition.
- The only possible reaction by CTM is its $coin$-transition; thus $(\overline{coffee}.CTM + \overline{tea}.CTM) \ \rho \ \overline{coffee}.CTM'$ must hold.

A Counterexample

Example 4.6 (Vending machines; cf. Example 3.13)

Show $CTM \not\sim CTM'$ where
$$\begin{aligned} CTM &= coin.\left(\overline{coffee}.CTM + \overline{tea}.CTM\right) \\ CTM' &= coin.\overline{coffee}.CTM' + coin.\overline{tea}.CTM'. \end{aligned}$$

Corresponding LTSs:



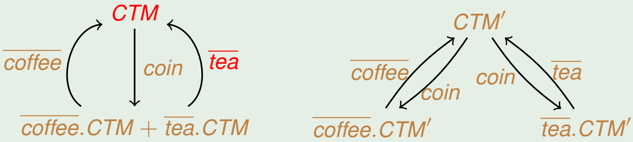
Assumption: there exists bisimulation ρ such that $CTM \; \rho \; CTM'$.

- First CTM' chooses the left $coin$-transition.
- The only possible reaction by CTM is its $coin$-transition; thus $(\overline{coffee}.CTM + \overline{tea}.CTM) \; \rho \; \overline{coffee}.CTM'$ must hold.
- CTM proceeds by selecting the $\overline{tea}$-transition.

Example 4.6 (Vending machines; cf. Example 3.13)

Show $CTM \not\sim CTM'$ where
$$\begin{aligned} CTM &= coin.\left(\overline{coffee}.CTM + \overline{tea}.CTM\right) \\ CTM' &= coin.\overline{coffee}.CTM' + coin.\overline{tea}.CTM'. \end{aligned}$$

Corresponding LTSs:



Assumption: there exists bisimulation ρ such that $CTM \; \rho \; CTM'$.

- First CTM' chooses the left $coin$-transition.
- The only possible reaction by CTM is its $coin$-transition; thus $(\overline{coffee}.CTM + \overline{tea}.CTM) \; \rho \; \overline{coffee}.CTM'$ must hold.
- CTM proceeds by selecting the $\overline{tea}$-transition.
- But CTM' cannot react to this step. ↯

(Verify using CAAL)

Outline of Lecture 4

Properties of Strong Bisimilarity

Lemma 4.7 (Properties of $\sim$)

(1) $\sim$ *is an equivalence relation (i.e., reflexive, symmetric, and transitive).*

Properties of Strong Bisimilarity

Lemma 4.7 (Properties of $\sim$)

(1) $\sim$ *is an* *equivalence relation* *(i.e., reflexive, symmetric, and transitive).*

(2) $\sim$ *is the* *coarsest* *strong bisimulation.*

Properties of Strong Bisimilarity

Lemma 4.7 (Properties of $\sim$)

(1) $\sim$ *is an equivalence relation (i.e., reflexive, symmetric, and transitive).*

(2) $\sim$ *is the coarsest strong bisimulation.*

Proof.

(1) $\sim$ is an equivalence relation:

- Reflexivity:

$$\mathrm{id}_{Prc} := \{(P, P) \mid P \in Prc\}$$

  is obviously a strong bisimulation.
  Since $\mathrm{id}_{Prc} \subseteq \sim$ by Definition 4.2, $\sim$ is reflexive.

Properties of Strong Bisimilarity

Lemma 4.7 (Properties of $\sim$)

(1) *$\sim$ is an equivalence relation (i.e., reflexive, symmetric, and transitive).*

(2) *$\sim$ is the coarsest strong bisimulation.*

Proof.

(1) $\sim$ is an equivalence relation:

- Symmetry: (**Caveat:** not every strong bisimulation is symmetric; cf. Example 4.4.)
  But if ρ is a strong bisimulation, then so is its inverse

$$\rho^{-1} := \{(Q, P) \mid P\rho Q\}$$

  (due to symmetry in Definition 4.3). Therefore, $\sim$ is symmetric by Definition 4.2.

Properties of Strong Bisimilarity

Lemma 4.7 (Properties of $\sim$)

(1) $\sim$ *is an equivalence relation (i.e., reflexive, symmetric, and transitive).*

(2) $\sim$ *is the coarsest strong bisimulation.*

Proof.

(1) $\sim$ is an equivalence relation:

- Transitivity: (**Caveat:** not every strong bisimulation is transitive.)
  But if ρ and σ are strong bisimulations, then so is their composition $\rho \circ \sigma := \{(P, R) \mid \exists Q : P\rho Q, Q\sigma R\}$.
  Proof: $P\ (\rho \circ \sigma)\ R$ and $P \xrightarrow{\alpha} P'$
  $\Rightarrow \exists Q : P\rho Q, Q\sigma R$ and $P \xrightarrow{\alpha} P'$ (def. $\circ$)
  $\Rightarrow \exists Q, Q' : Q\sigma R, Q \xrightarrow{\alpha} Q'$ and $P' \rho Q'$ (ρ strong bisimulation)
  $\Rightarrow \exists Q', R' : P' \rho Q', R \xrightarrow{\alpha} R'$ and $Q' \sigma R'$ (σ strong bisimulation)
  $\Rightarrow \exists R' : R \xrightarrow{\alpha} R'$ and $P'\ (\rho \circ \sigma)\ R'$ (def. $\circ$)
  (analogously for assumption $R \xrightarrow{\alpha} R'$)
  Therefore, $\sim$ is transitive by Definition 4.2.

Properties of Strong Bisimilarity

Lemma 4.7 (Properties of $\sim$)

(1) $\sim$ *is an equivalence relation (i.e., reflexive, symmetric, and transitive).*

(2) $\sim$ *is the coarsest strong bisimulation.*

Proof.

(2) $\sim$ is the coarsest strong bisimulation:
According to Definition 4.2, it suffices to show that strong bisimulations are closed under union, i.e., whenever ρ, σ are bisimulations, then so is $\rho \cup \sigma$. This immediately follows by case distinction. $\square$

Outline of Lecture 4

Lemma 4.8 (Bisimulation on paths)

Whenever we have:

$$
\begin{array}{ccccccccc}
P_0 & \xrightarrow{\alpha_1} & P_1 & \xrightarrow{\alpha_2} & P_2 & \xrightarrow{\alpha_3} & P_3 & \xrightarrow{\alpha_4} & P_4 \ldots\ldots \\
\rho & & & & & & & & \\
Q_0 & & & & & & & &
\end{array}
$$

Bisimulation on Paths

Lemma 4.8 (Bisimulation on paths)

Whenever we have:

$$\begin{array}{ccccccccc} P_0 & \xrightarrow{\alpha_1} & P_1 & \xrightarrow{\alpha_2} & P_2 & \xrightarrow{\alpha_3} & P_3 & \xrightarrow{\alpha_4} & P_4 \ldots\ldots \\ \rho & & & & & & & & \\ Q_0 & & & & & & & & \end{array}$$

this can be completed to

$$\begin{array}{ccccccccc} P_0 & \xrightarrow{\alpha_1} & P_1 & \xrightarrow{\alpha_2} & P_2 & \xrightarrow{\alpha_3} & P_3 & \xrightarrow{\alpha_4} & P_4 \ldots\ldots \\ \rho & & \rho & & \rho & & \rho & & \rho \\ Q_0 & \xrightarrow{\alpha_1} & Q_1 & \xrightarrow{\alpha_2} & Q_2 & \xrightarrow{\alpha_3} & Q_3 & \xrightarrow{\alpha_4} & Q_4 \ldots\ldots \end{array}$$

Bisimulation on Paths

Lemma 4.8 (Bisimulation on paths)

Whenever we have:

$$\begin{array}{ccccccccc} P_0 & \xrightarrow{\alpha_1} & P_1 & \xrightarrow{\alpha_2} & P_2 & \xrightarrow{\alpha_3} & P_3 & \xrightarrow{\alpha_4} & P_4 \ldots\ldots \\ \rho & & & & & & & & \\ Q_0 & & & & & & & & \end{array}$$

this can be completed to

$$\begin{array}{ccccccccc} P_0 & \xrightarrow{\alpha_1} & P_1 & \xrightarrow{\alpha_2} & P_2 & \xrightarrow{\alpha_3} & P_3 & \xrightarrow{\alpha_4} & P_4 \ldots\ldots \\ \rho & & \rho & & \rho & & \rho & & \rho \\ Q_0 & \xrightarrow{\alpha_1} & Q_1 & \xrightarrow{\alpha_2} & Q_2 & \xrightarrow{\alpha_3} & Q_3 & \xrightarrow{\alpha_4} & Q_4 \ldots\ldots \end{array}$$

Proof.

by induction on the length of the path $\square$

Strong Bisimilarity vs. Trace Equivalence

Theorem 4.9

$P \sim Q$ implies that P and Q are trace equivalent. The reverse does generally not hold.

Strong Bisimilarity vs. Trace Equivalence

Theorem 4.9

$P \sim Q$ implies that P and Q are trace equivalent. The reverse does generally not hold.

Proof.

The implication from left to right follows from Lemma 4.8.

Strong Bisimilarity vs. Trace Equivalence

Theorem 4.9

$P \sim Q$ implies that P and Q are trace equivalent. The reverse does generally not hold.

Proof.

The implication from left to right follows from Lemma 4.8.

Consider the other direction:

- Take $P = a.P_1$ with $P_1 = b.\mathrm{nil} + c.\mathrm{nil}$ and $Q = a.b.nil + a.c.\mathrm{nil}$.

Strong Bisimilarity vs. Trace Equivalence

Theorem 4.9

$P \sim Q$ implies that P and Q are trace equivalent. The reverse does generally not hold.

Proof.

The implication from left to right follows from Lemma 4.8.

Consider the other direction:

- Take $P = a.P_1$ with $P_1 = b.\mathrm{nil} + c.\mathrm{nil}$ and $Q = a.b.nil + a.c.\mathrm{nil}$.
- Then: $Tr(P) = \{\epsilon, a, ab, ac\} = Tr(Q)$.
- Thus, P and Q are trace equivalent.

Strong Bisimilarity vs. Trace Equivalence

Theorem 4.9

$P \sim Q$ implies that P and Q are trace equivalent. The reverse does generally not hold.

Proof.

The implication from left to right follows from Lemma 4.8.

Consider the other direction:

- Take $P = a.P_1$ with $P_1 = b.\text{nil} + c.\text{nil}$ and $Q = a.b.nil + a.c.\text{nil}$.
- Then: $Tr(P) = \{\epsilon, a, ab, ac\} = Tr(Q)$.
- Thus, P and Q are trace equivalent.
- But: $P \not\sim Q$, as there is no state in the LTS of Q that is bisimilar to P_1 (cf. Example 4.6).
- Why? Since no state in Q can perform both b and c.

$\square$

Definition 4.10 (Determinism)

$P \in Prc$ is deterministic whenever for every of its reachable states R it holds:

$$\left(R \xrightarrow{\alpha} R' \text{ and } R \xrightarrow{\alpha} R''\right) \quad \text{implies} \quad R' = R''.$$

Deterministic Transition Systems I

Definition 4.10 (Determinism)

$P \in Prc$ is deterministic whenever for every of its reachable states R it holds:

$$\left(R \xrightarrow{\alpha} R' \text{ and } R \xrightarrow{\alpha} R''\right) \quad \text{implies} \quad R' = R''.$$

Theorem 4.11 (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Deterministic Transition Systems II

Theorem (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Deterministic Transition Systems II

Theorem (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Proof.

By Theorem 4.9, it remains to prove that $Tr(P) = Tr(Q)$ implies $P \sim Q$.

Deterministic Transition Systems II

Theorem (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Proof.

By Theorem 4.9, it remains to prove that $Tr(P) = Tr(Q)$ implies $P \sim Q$.

To this end, we show that

$$\rho := \{(R, S) \mid P \longrightarrow^* R, Q \longrightarrow^* S, Tr(R) = Tr(S)\}$$

is a strong bisimulation.

Deterministic Transition Systems II

Theorem (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Proof.

By Theorem 4.9, it remains to prove that $Tr(P) = Tr(Q)$ implies $P \sim Q$.

To this end, we show that

$$\rho := \{(R, S) \mid P \longrightarrow^* R, Q \longrightarrow^* S, Tr(R) = Tr(S)\}$$

is a strong bisimulation.

- Let $R \rho S$ and $R \stackrel{\alpha}{\longrightarrow} R'$ (reverse implication analogous).

Deterministic Transition Systems II

Theorem (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Proof.

By Theorem 4.9, it remains to prove that $Tr(P) = Tr(Q)$ implies $P \sim Q$.

To this end, we show that

$$\rho := \{(R, S) \mid P \longrightarrow^* R, Q \longrightarrow^* S, Tr(R) = Tr(S)\}$$

is a strong bisimulation.

- Let $R \rho S$ and $R \xrightarrow{\alpha} R'$ (reverse implication analogous).
- As P is deterministic, $\{w \in Tr(R) \mid w = \alpha \ldots\} = \alpha \cdot Tr(R')$.

Deterministic Transition Systems II

Theorem (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Proof.

By Theorem 4.9, it remains to prove that $Tr(P) = Tr(Q)$ implies $P \sim Q$.

To this end, we show that

$$\rho := \{(R, S) \mid P \longrightarrow^* R, Q \longrightarrow^* S, Tr(R) = Tr(S)\}$$

is a strong bisimulation.

- Let $R \rho S$ and $R \xrightarrow{\alpha} R'$ (reverse implication analogous).
- As P is deterministic, $\{w \in Tr(R) \mid w = \alpha \ldots\} = \alpha \cdot Tr(R')$.
- As $Tr(R) = Tr(S)$, there ex. $w \in Tr(S)$ such that $w = \alpha \ldots$.

Deterministic Transition Systems II

Theorem (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Proof.

By Theorem 4.9, it remains to prove that $Tr(P) = Tr(Q)$ implies $P \sim Q$.

To this end, we show that

$$\rho := \{(R, S) \mid P \longrightarrow^* R, Q \longrightarrow^* S, Tr(R) = Tr(S)\}$$

is a strong bisimulation.

- Let $R \rho S$ and $R \xrightarrow{\alpha} R'$ (reverse implication analogous).
- As P is deterministic, $\{w \in Tr(R) \mid w = \alpha \ldots\} = \alpha \cdot Tr(R')$.
- As $Tr(R) = Tr(S)$, there ex. $w \in Tr(S)$ such that $w = \alpha \ldots$.
- Hence ex. $S' \in Prc$ with $S \xrightarrow{\alpha} S'$.

Deterministic Transition Systems II

Theorem (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Proof.

By Theorem 4.9, it remains to prove that $Tr(P) = Tr(Q)$ implies $P \sim Q$.

To this end, we show that

$$\rho := \{(R, S) \mid P \longrightarrow^* R, Q \longrightarrow^* S, Tr(R) = Tr(S)\}$$

is a strong bisimulation.

- Let $R \rho S$ and $R \xrightarrow{\alpha} R'$ (reverse implication analogous).
- As P is deterministic, $\{w \in Tr(R) \mid w = \alpha \ldots\} = \alpha \cdot Tr(R')$.
- As $Tr(R) = Tr(S)$, there ex. $w \in Tr(S)$ such that $w = \alpha \ldots$.
- Hence ex. $S' \in Prc$ with $S \xrightarrow{\alpha} S'$.
- Again by determinism, $\{w \in Tr(S) \mid w = \alpha \ldots\} = \alpha \cdot Tr(S')$.

Deterministic Transition Systems II

Theorem (Determinism implies coincidence of $\sim$ and trace equiv.) (Park 1981)

For deterministic P and Q: $P \sim Q$ iff $Tr(P) = Tr(Q)$.

Proof.

By Theorem 4.9, it remains to prove that $Tr(P) = Tr(Q)$ implies $P \sim Q$.

To this end, we show that

$$\rho := \{(R, S) \mid P \longrightarrow^* R, Q \longrightarrow^* S, Tr(R) = Tr(S)\}$$

is a strong bisimulation.

- Let $R \rho S$ and $R \xrightarrow{\alpha} R'$ (reverse implication analogous).
- As P is deterministic, $\{w \in Tr(R) \mid w = \alpha \ldots\} = \alpha \cdot Tr(R')$.
- As $Tr(R) = Tr(S)$, there ex. $w \in Tr(S)$ such that $w = \alpha \ldots$.
- Hence ex. $S' \in Prc$ with $S \xrightarrow{\alpha} S'$.
- Again by determinism, $\{w \in Tr(S) \mid w = \alpha \ldots\} = \alpha \cdot Tr(S')$.
- Altogether, $Tr(R') = Tr(S')$ and thus $R' \rho S'$, which completes the proof. $\square$

Outline of Lecture 4

Congruence I

Theorem 4.12 (CCS congruence property of $\sim$)

Strong bisimilarity $\sim$ *is a CCS congruence, that is, whenever* $P, Q \in Prc$ *such that* $P \sim Q$*,*

$$\begin{array}{rl} \alpha.P \sim \alpha.Q & \text{for every } \alpha \in Act \\ P + R \sim Q + R & \text{for every } R \in Prc \\ P \parallel R \sim Q \parallel R & \text{for every } R \in Prc \\ P \setminus L \sim Q \setminus L & \text{for every } L \subseteq A \\ P[f] \sim Q[f] & \text{for every } f : A \to A \end{array}$$

Congruence II

Proof.

We only consider parallel composition and prove $P \parallel R \sim Q \parallel R$ by showing that

$$\rho := \{(P' \parallel R', Q' \parallel R') \mid P' \sim Q', R \longrightarrow^* R'\}$$

is a strong bisimulation.

Congruence II

Proof.

We only consider parallel composition and prove $P \parallel R \sim Q \parallel R$ by showing that

$$\rho := \{(P' \parallel R', Q' \parallel R') \mid P' \sim Q', R \longrightarrow^* R'\}$$

is a strong bisimulation.

To this aim, let $(P' \parallel R')\ \rho\ (Q' \parallel R')$.

- If $P' \parallel R' \xrightarrow{\alpha} S'$, the following cases are possible:

Congruence II

Proof.

We only consider parallel composition and prove $P \parallel R \sim Q \parallel R$ by showing that

$$\rho := \{(P' \parallel R', Q' \parallel R') \mid P' \sim Q', R \longrightarrow^* R'\}$$

is a strong bisimulation.

To this aim, let $(P' \parallel R')\ \rho\ (Q' \parallel R')$.

- If $P' \parallel R' \xrightarrow{\alpha} S'$, the following cases are possible:
  (1) $P' \xrightarrow{\alpha} P''$ and $S' = P'' \parallel R'$:
  Since $P' \sim Q'$, there ex. Q'' such that $Q' \xrightarrow{\alpha} Q''$ and $P'' \sim Q''$.
  Thus, $Q' \parallel R' \xrightarrow{\alpha} Q'' \parallel R'$ and $S'\ \rho\ (Q'' \parallel R')$.

Congruence II

Proof.

We only consider parallel composition and prove $P \parallel R \sim Q \parallel R$ by showing that

$$\rho := \{(P' \parallel R', Q' \parallel R') \mid P' \sim Q', R \longrightarrow^* R'\}$$

is a strong bisimulation.

To this aim, let $(P' \parallel R') \, \rho \, (Q' \parallel R')$.

- If $P' \parallel R' \xrightarrow{\alpha} S'$, the following cases are possible:

  (1) $P' \xrightarrow{\alpha} P''$ and $S' = P'' \parallel R'$:
  Since $P' \sim Q'$, there ex. Q'' such that $Q' \xrightarrow{\alpha} Q''$ and $P'' \sim Q''$.
  Thus, $Q' \parallel R' \xrightarrow{\alpha} Q'' \parallel R'$ and $S' \, \rho \, (Q'' \parallel R')$.

  (2) $R' \xrightarrow{\alpha} R''$ and $S' = P' \parallel R''$:
  Here $Q' \parallel R' \xrightarrow{\alpha} Q' \parallel R''$ and $S' \, \rho \, (Q' \parallel R'')$.

Proof.

We only consider parallel composition and prove $P \parallel R \sim Q \parallel R$ by showing that

$$\rho := \{(P' \parallel R', Q' \parallel R') \mid P' \sim Q', R \longrightarrow^* R'\}$$

is a strong bisimulation.

To this aim, let $(P' \parallel R')\ \rho\ (Q' \parallel R')$.

- If $P' \parallel R' \xrightarrow{\alpha} S'$, the following cases are possible:
  (1) $P' \xrightarrow{\alpha} P''$ and $S' = P'' \parallel R'$:
  Since $P' \sim Q'$, there ex. Q'' such that $Q' \xrightarrow{\alpha} Q''$ and $P'' \sim Q''$.
  Thus, $Q' \parallel R' \xrightarrow{\alpha} Q'' \parallel R'$ and $S'\ \rho\ (Q'' \parallel R')$.
  (2) $R' \xrightarrow{\alpha} R''$ and $S' = P' \parallel R''$:
  Here $Q' \parallel R' \xrightarrow{\alpha} Q' \parallel R''$ and $S'\ \rho\ (Q' \parallel R'')$.
  (3) $\alpha = \tau$, $P' \xrightarrow{\lambda} P''$, $R' \xrightarrow{\overline{\lambda}} R''$ (for some $\lambda \in A \cup \overline{A}$) and $S' = P'' \parallel R''$:
  combination of (1) and (2).

Congruence II

Proof.

We only consider parallel composition and prove $P \parallel R \sim Q \parallel R$ by showing that

$$\rho := \{(P' \parallel R', Q' \parallel R') \mid P' \sim Q', R \longrightarrow^* R'\}$$

is a strong bisimulation.

To this aim, let $(P' \parallel R') \, \rho \, (Q' \parallel R')$.

- If $P' \parallel R' \xrightarrow{\alpha} S'$, the following cases are possible:
  (1) $P' \xrightarrow{\alpha} P''$ and $S' = P'' \parallel R'$:
  Since $P' \sim Q'$, there ex. Q'' such that $Q' \xrightarrow{\alpha} Q''$ and $P'' \sim Q''$.
  Thus, $Q' \parallel R' \xrightarrow{\alpha} Q'' \parallel R'$ and $S' \, \rho \, (Q'' \parallel R')$.
  (2) $R' \xrightarrow{\alpha} R''$ and $S' = P' \parallel R''$:
  Here $Q' \parallel R' \xrightarrow{\alpha} Q' \parallel R''$ and $S' \, \rho \, (Q' \parallel R'')$.
  (3) $\alpha = \tau$, $P' \xrightarrow{\lambda} P''$, $R' \xrightarrow{\overline{\lambda}} R''$ (for some $\lambda \in A \cup \overline{A}$) and $S' = P'' \parallel R''$:
  combination of (1) and (2).
- $Q' \parallel R' \xrightarrow{\alpha} T'$: analogous

$\square$

Outline of Lecture 4

Deadlock Sensitivity of Strong Bisimilarity

Definition (Deadlock; cf. Definition 3.8)

Let $P, Q \in Prc$ and $w \in Act^*$ such that

$$P \xrightarrow{w} Q \quad \text{and} \quad Q \not\longrightarrow .$$

Then Q is called a *w-deadlock* of P.

Definition (Deadlock sensitivity; cf. Definition 3.10)

Relation $\equiv \subseteq Prc \times Prc$ is *deadlock sensitive* whenever:

$$P \equiv Q \quad \text{implies} \quad (\forall w \in Act^* : P \text{ has a } w\text{-deadlock iff } Q \text{ has a } w\text{-deadlock}) .$$

Deadlock Sensitivity of Strong Bisimilarity

Definition (Deadlock sensitivity; cf. Definition 3.10)

Relation $\equiv \subseteq Prc \times Prc$ is deadlock sensitive whenever:

$$P \equiv Q \quad \text{implies} \quad (\forall w \in Act^* : P \text{ has a } w\text{-deadlock iff } Q \text{ has a } w\text{-deadlock}) .$$

Theorem 4.13

$\sim$ is deadlock sensitive.

Deadlock Sensitivity of Strong Bisimilarity

Definition (Deadlock sensitivity; cf. Definition 3.10)

Relation $\equiv \subseteq Prc \times Prc$ is deadlock sensitive whenever:

$$P \equiv Q \quad \text{implies} \quad (\forall w \in Act^* : P \text{ has a } w\text{-deadlock iff } Q \text{ has a } w\text{-deadlock}) .$$

Theorem 4.13

$\sim$ is deadlock sensitive.

Proof.

Let $P \sim Q$.

- We assume that, for some $w \in Act^*$, P has a w-deadlock but Q does not (or vice versa).

Definition (Deadlock sensitivity; cf. Definition 3.10)

Relation $\equiv \subseteq Prc \times Prc$ is deadlock sensitive whenever:

$$P \equiv Q \quad \text{implies} \quad (\forall w \in Act^* : P \text{ has a } w\text{-deadlock iff } Q \text{ has a } w\text{-deadlock}) .$$

Theorem 4.13

$\sim$ is deadlock sensitive.

Proof.

Let $P \sim Q$.

- We assume that, for some $w \in Act^*$, P has a w-deadlock but Q does not (or vice versa).
- Thus, there exists $P' \in Prc$ such that $P \xrightarrow{w} P'$ and $P' \not\longrightarrow$.
- Moreover, for all $Q' \in Prc$ with $Q \xrightarrow{w} Q'$ there exist $\alpha \in Act$ and $Q'' \in Prc$ such that $Q' \xrightarrow{\alpha} Q''$.

Deadlock Sensitivity of Strong Bisimilarity

Definition (Deadlock sensitivity; cf. Definition 3.10)

Relation $\equiv \subseteq Prc \times Prc$ is deadlock sensitive whenever:

$$P \equiv Q \quad \text{implies} \quad (\forall w \in Act^* : P \text{ has a } w\text{-deadlock iff } Q \text{ has a } w\text{-deadlock}) .$$

Theorem 4.13

$\sim$ is deadlock sensitive.

Proof.

Let $P \sim Q$.

- We assume that, for some $w \in Act^*$, P has a w-deadlock but Q does not (or vice versa).
- Thus, there exists $P' \in Prc$ such that $P \xrightarrow{w} P'$ and $P' \not\longrightarrow$.
- Moreover, for all $Q' \in Prc$ with $Q \xrightarrow{w} Q'$ there exist $\alpha \in Act$ and $Q'' \in Prc$ such that $Q' \xrightarrow{\alpha} Q''$.
- For $P \xrightarrow{w} P'$, Lemma 4.8 (bisimulation on paths) yields Q' with $Q \xrightarrow{w} Q'$ and $P' \sim Q'$.
- Thus $P' \not\longrightarrow$ and $Q' \xrightarrow{\alpha} Q''$ cannot hold at the same time. $\lightning$

$\square$

Outline of Lecture 4

Example 4.14 (An n-ary semaphore)

S_i^n stands for a semaphore for n identical, exclusive resources i of which are taken:

$$\begin{aligned} S_0^n &= get.S_1^n \\ S_i^n &= get.S_{i+1}^n + put.S_{i-1}^n \quad \text{for } 0 < i < n \\ S_n^n &= put.S_{n-1}^n \end{aligned}$$

Semaphores I

Example 4.14 (An n-ary semaphore)

S_i^n stands for a semaphore for n identical, exclusive resources i of which are taken:

$$\begin{array}{rcl} S_0^n & = & get.S_1^n \\ S_i^n & = & get.S_{i+1}^n + put.S_{i-1}^n \quad \text{for } 0 < i < n \\ S_n^n & = & put.S_{n-1}^n \end{array}$$

This process is strongly bisimilar to n parallel binary semaphores:

Lemma 4.15

For every $n \in \mathbb{N}_+$, *we have:* $S_0^n \sim \underbrace{S_0^1 \parallel \cdots \parallel S_0^1}_{n \text{ times}}$.

Semaphores II

Lemma

For every $n \in \mathbb{N}_+$*, we have:* $S_0^n \sim \underbrace{S_0^1 \parallel \cdots \parallel S_0^1}_{n \text{ times}}$.

Semaphores II

Lemma

For every $n \in \mathbb{N}_+$, *we have:* $S_0^n \sim \underbrace{S_0^1 \parallel \cdots \parallel S_0^1}_{n \text{ times}}$.

Proof.

Consider the following binary relation where $i_1, \ldots, i_n \in \{0, 1\}$:

$$\rho = \left\{ \left(S_k^n, S_{i_1}^1 \parallel \cdots \parallel S_{i_n}^1\right) \;\middle|\; \sum_{j=1}^{n} i_j = k \right\}$$

Semaphores II

Lemma

For every $n \in \mathbb{N}_+$, we have: $S_0^n \sim \underbrace{S_0^1 \parallel \cdots \parallel S_0^1}_{n \text{ times}}$.

Proof.

Consider the following binary relation where $i_1, \ldots, i_n \in \{0, 1\}$:

$$\rho = \left\{ \left(S_k^n, S_{i_1}^1 \parallel \cdots \parallel S_{i_n}^1\right) \;\middle|\; \sum_{j=1}^{n} i_j = k \right\}$$

Then: ρ is a strong bisimulation and $(S_0^n, \underbrace{S_0^1 \parallel \cdots \parallel S_0^1}_{n \text{ times}}) \in \rho$. □