Concurrency Theory

Winter Semester 2019/20

Lecture 13: Properties of Strong Bisimulation

Joost-Pieter Katoen and Thomas Noll
Software Modeling and Verification Group
RWTH Aachen University

`https://moves.rwth-aachen.de/teaching/ws-19-20/ct/`

Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Outline of Lecture 13

Strong Bisimulation

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

A binary relation $\rho \subseteq Prc \times Prc$ is a strong bisimulation whenever 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

Definition (Strong bisimilarity)

Processes P and Q are strongly bisimilar, denoted $P \sim Q$, iff there is a strong bisimulation ρ with $P \,\rho\, Q$. Thus,

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

Relation $\sim$ is called strong bisimilarity.

Properties of Strong Bisimilarity

Lemma (Properties of $\sim$)

1. $\sim$ *is an equivalence relation (i.e., reflexive, symmetric, and transitive)*
2. $\sim$ *is the coarsest strong bisimulation*

Proof.

on the board □

Strong Bisimulation vs. Trace Equivalence

Theorem

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

Proof.

The implication from left to right follows from the previous slide.

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 12.5).
Why? No state in Q can perform both b and c. □

Congruence

Theorem (CCS congruence property of $\sim$)

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

$$
\begin{array}{rcll}
\alpha.P & \sim & \alpha.Q & \text{for every action } \alpha \\
P + R & \sim & Q + R & \text{for every process } R \\
P \parallel R & \sim & Q \parallel R & \text{for every process } R \\
P \setminus L & \sim & Q \setminus L & \text{for every set } L \subseteq A \\
P[f] & \sim & Q[f] & \text{for every relabelling } f : A \to A
\end{array}
$$

Proof.

- for $\parallel$: on the board
- for other CCS operators: left as an exercise

□

Outline of Lecture 13

Deadlock Sensitivity of Strong Bisimulation

Definition (Deadlock; cf. Definition 11.6)

Let $P, Q \in Prc$ and $w \in Act^*$ such that $P \xrightarrow{w} Q$ and $Q \not\longrightarrow$. Then Q is called a w-deadlock of P.

Deadlock Sensitivity of Strong Bisimulation

Definition (Deadlock; cf. Definition 11.6)

Let $P, Q \in Prc$ and $w \in Act^*$ such that $P \xrightarrow{w} Q$ and $Q \not\longrightarrow$. Then Q is called a w-deadlock of P.

Definition (Deadlock sensitivity; cf. Definition 11.8)

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

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

Deadlock Sensitivity of Strong Bisimulation

Definition (Deadlock; cf. Definition 11.6)

Let $P, Q \in Prc$ and $w \in Act^*$ such that $P \xrightarrow{w} Q$ and $Q \not\longrightarrow$. Then Q is called a w-deadlock of P.

Definition (Deadlock sensitivity; cf. Definition 11.8)

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

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

Theorem 13.1

$\sim$ is deadlock sensitive.

Deadlock Sensitivity of Strong Bisimulation

Definition (Deadlock; cf. Definition 11.6)

Let $P, Q \in Prc$ and $w \in Act^*$ such that $P \xrightarrow{w} Q$ and $Q \not\longrightarrow$. Then Q is called a w-deadlock of P.

Definition (Deadlock sensitivity; cf. Definition 11.8)

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

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

Theorem 13.1

$\sim$ *is deadlock sensitive.*

Proof.

on the board □

Outline of Lecture 13

Two Buffers

Example 13.2

One-place buffer:

$$\begin{aligned} B_0^1 &= in.B_1^1 \\ B_1^1 &= \overline{out}.B_0^1. \end{aligned}$$

Buffers Revisited

Two Buffers

Example 13.2

One-place buffer:

$$\begin{aligned} B^1_0 &= in.B^1_1 \\ B^1_1 &= \overline{out}.B^1_0. \end{aligned}$$

Two-place buffer:

$$\begin{aligned} B^2_0 &= in.B^2_1 \\ B^2_1 &= in.B^2_2 + \overline{out}.B^2_0 \\ B^2_2 &= \overline{out}.B^2_1. \end{aligned}$$

Two Buffers

Example 13.2

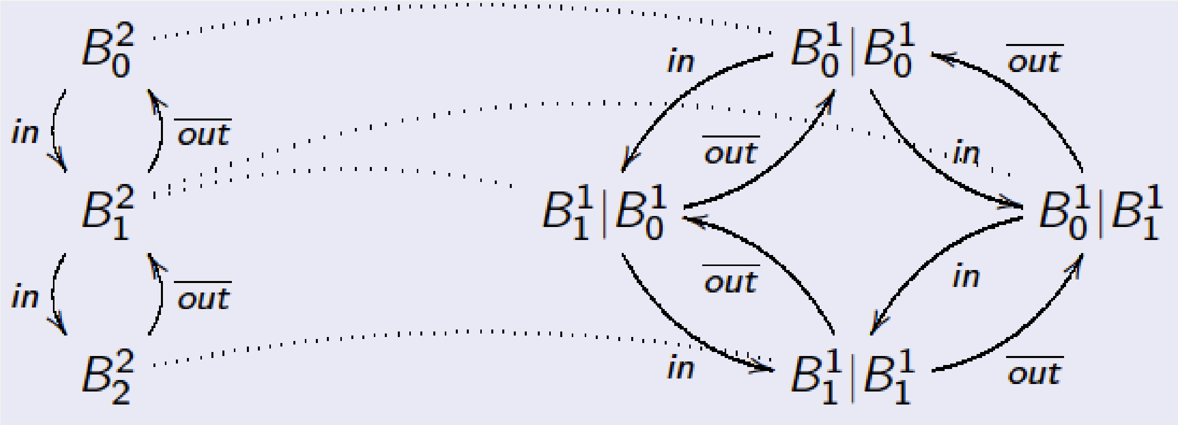
One-place buffer:

$$\begin{aligned} B^1_0 &= in.B^1_1 \\ B^1_1 &= \overline{out}.B^1_0. \end{aligned}$$

Two-place buffer:

$$\begin{aligned} B^2_0 &= in.B^2_1 \\ B^2_1 &= in.B^2_2 + \overline{out}.B^2_0 \\ B^2_2 &= \overline{out}.B^2_1. \end{aligned}$$

$B^2_0 \sim B^1_0 \parallel B^1_0$:



Semaphores I

Example 13.3 (An n-ary semaphore)

Let S_i^n stand for a semaphore for n 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 13.3 (An n-ary semaphore)

Let S^n_i stand for a semaphore for n exclusive resources i of which are taken:

$$\begin{aligned} S^n_0 &= get.S^n_1 \\ S^n_i &= get.S^n_{i+1} + put.S^n_{i-1} \quad \text{for } 0 < i < n \\ S^n_n &= put.S^n_{n-1} \end{aligned}$$

This process is strongly bisimilar to n parallel binary semaphores:

Lemma 13.4

For every $n \in \mathbb{N}_+$, we have: $S^n_0 \sim \underbrace{S^1_0 \parallel \cdots \parallel S^1_0}_{n \text{ times}}$.

Buffers Revisited

Semaphores II

Lemma

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

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 $i_1, i_2, \ldots, i_n \in \{0, 1\}$:

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

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 $i_1, i_2, \ldots, i_n \in \{0, 1\}$:

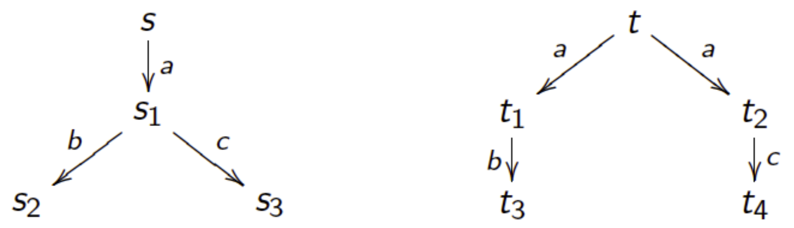
$$\rho = \left\{ \left(S_i^n, S_{i_1}^1 \parallel \cdots \parallel S_{i_n}^1 \right) \;\middle|\; \sum_{j=1}^{n} i_j = i \right\}$$

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

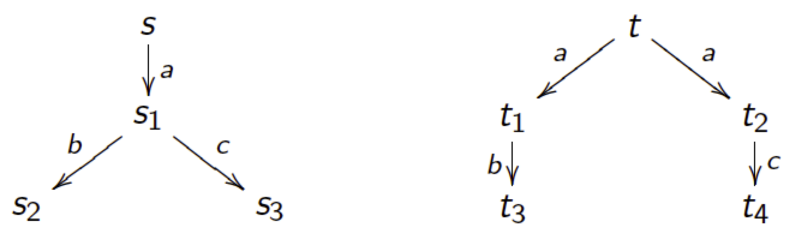
Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Outline of Lecture 13

How to Show Non-Bisimilarity?



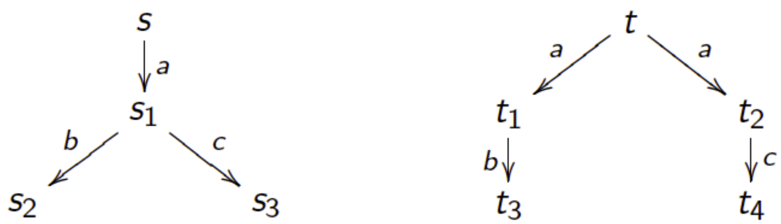
How to Show Non-Bisimilarity?



Alternatives to prove that $s \not\sim t$

- Enumerate all binary relations and show that none of those containing (s, t) is a strong bisimulation.

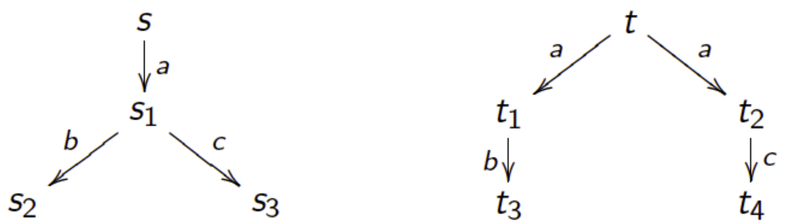
How to Show Non-Bisimilarity?



Alternatives to prove that $s \not\sim t$

- Enumerate all binary relations and show that none of those containing (s, t) is a strong bisimulation. (This is expensive, as there are 2^{k^2} binary relations on Prc with $|Prc| = k$.)

How to Show Non-Bisimilarity?

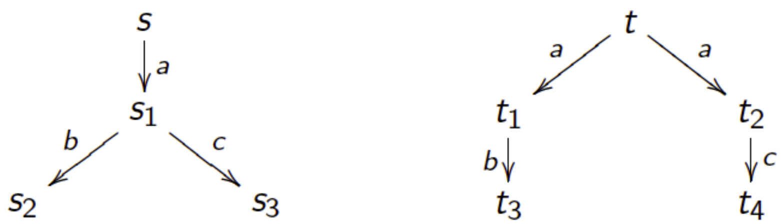


Alternatives to prove that $s \not\sim t$

- Enumerate all binary relations and show that none of those containing (s, t) is a strong bisimulation. (This is expensive, as there are 2^{k^2} binary relations on Prc with $|Prc| = k$.)
- Make certain observations which will enable to disqualify many bisimulation candidates in one step.

Strong Bisimilarity as a Game

How to Show Non-Bisimilarity?

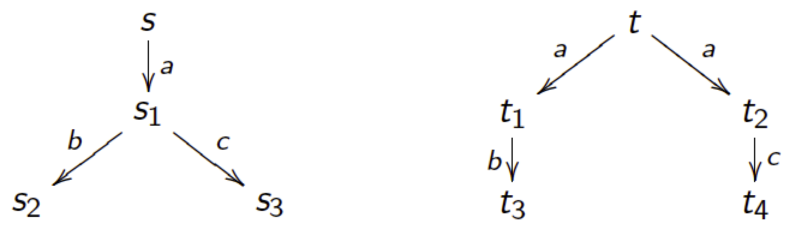


Alternatives to prove that $s \not\sim t$

- Enumerate all binary relations and show that none of those containing (s, t) is a strong bisimulation. (This is expensive, as there are 2^{k^2} binary relations on Prc with $|Prc| = k$.)
- Make certain observations which will enable to disqualify many bisimulation candidates in one step. (Yields heuristics – how about completeness?)

Strong Bisimilarity as a Game

How to Show Non-Bisimilarity?



Alternatives to prove that $s \not\sim t$

- Enumerate all binary relations and show that none of those containing (s, t) is a strong bisimulation. (This is expensive, as there are 2^{k^2} binary relations on Prc with $|Prc| = k$.)
- Make certain observations which will enable to disqualify many bisimulation candidates in one step. (Yields heuristics – how about completeness?)
- Use game characterisation of strong bisimilarity.

Strong Bisimilarity as a Game

The Strong Bisimulation Game

Let $(Prc, Act, \longrightarrow)$ be an LTS and $s, t \in Prc$. Question: does $s \sim t$?

Strong Bisimilarity as a Game

The Strong Bisimulation Game

Let $(Prc, Act, \longrightarrow)$ be an LTS and $s, t \in Prc$. Question: does $s \sim t$?

We define a game with two players: an “attacker” and a “defender”.

Strong Bisimilarity as a Game

The Strong Bisimulation Game

Let $(Prc, Act, \longrightarrow)$ be an LTS and $s, t \in Prc$. Question: does $s \sim t$?

We define a game with two players: an "attacker" and a "defender".

- The game is played in rounds and configurations of the game are pairs of states from $Prc \times Prc$.

Strong Bisimilarity as a Game

The Strong Bisimulation Game

Let $(\mathit{Prc}, \mathit{Act}, \longrightarrow)$ be an LTS and $s, t \in \mathit{Prc}$. Question: does $s \sim t$?

We define a game with two players: an “attacker” and a “defender”.

- The game is played in rounds and configurations of the game are pairs of states from $\mathit{Prc} \times \mathit{Prc}$.
- In each round exactly one configuration is called current.

Strong Bisimilarity as a Game

The Strong Bisimulation Game

Let $(\mathit{Prc}, \mathit{Act}, \longrightarrow)$ be an LTS and $s, t \in \mathit{Prc}$. Question: does $s \sim t$?

We define a game with two players: an “attacker” and a “defender”.

- The game is played in rounds and configurations of the game are pairs of states from $\mathit{Prc} \times \mathit{Prc}$.
- In each round exactly one configuration is called current.
- Initially, the configuration (s, t) is the current one.

Strong Bisimilarity as a Game

The Strong Bisimulation Game

Let $(Prc, Act, \longrightarrow)$ be an LTS and $s, t \in Prc$. Question: does $s \sim t$?

We define a game with two players: an “attacker” and a “defender”.

- The game is played in rounds and configurations of the game are pairs of states from $Prc \times Prc$.
- In each round exactly one configuration is called current.
- Initially, the configuration (s, t) is the current one.

Intuition

The defender wants to show that $s \sim t$ while the attacker aims to prove the opposite.

Rules of the Bisimulation Game

Rules

In each round, the current configuration (s, t) is changed as follows:

1. the attacker chooses one of the two processes in the current configuration, say t, and makes an $\xrightarrow{\alpha}$-move for some $\alpha \in Act$ to t', say,

Rules of the Bisimulation Game

Rules

In each round, the current configuration (s, t) is changed as follows:

1. the attacker chooses one of the two processes in the current configuration, say t, and makes an $\xrightarrow{\alpha}$-move for some $\alpha \in Act$ to t', say, and
2. the defender must respond by making an $\xrightarrow{\alpha}$-move in the other process s of the current configuration under the same action α, yielding $s \xrightarrow{\alpha} s'$.

Rules of the Bisimulation Game

Rules

In each round, the current configuration (s, t) is changed as follows:

1. the attacker chooses one of the two processes in the current configuration, say t, and makes an $\xrightarrow{\alpha}$-move for some $\alpha \in Act$ to t', say, and
2. the defender must respond by making an $\xrightarrow{\alpha}$-move in the other process s of the current configuration under the same action α, yielding $s \xrightarrow{\alpha} s'$.

The pair of processes (s', t') becomes the new current configuration.
The game continues with another round.

Strong Bisimilarity as a Game

Rules of the Bisimulation Game

Rules

In each round, the current configuration (s, t) is changed as follows:

1. the attacker chooses one of the two processes in the current configuration, say t, and makes an $\stackrel{\alpha}{\longrightarrow}$-move for some $\alpha \in Act$ to t', say, and
2. the defender must respond by making an $\stackrel{\alpha}{\longrightarrow}$-move in the other process s of the current configuration under the same action α, yielding $s \stackrel{\alpha}{\longrightarrow} s'$.

The pair of processes (s', t') becomes the new current configuration.
The game continues with another round.

Results

1. If one player cannot move, the other player wins:
   - attacker cannot move if $s \not\rightarrow$ and $t \not\rightarrow$
   - defender cannot move if no matching transition available

Rules of the Bisimulation Game

Rules

In each round, the current configuration (s, t) is changed as follows:

1. the attacker chooses one of the two processes in the current configuration, say t, and makes an $\xrightarrow{\alpha}$-move for some $\alpha \in Act$ to t', say, and
2. the defender must respond by making an $\xrightarrow{\alpha}$-move in the other process s of the current configuration under the same action α, yielding $s \xrightarrow{\alpha} s'$.

The pair of processes (s', t') becomes the new current configuration.
The game continues with another round.

Results

1. If one player cannot move, the other player wins:
   - attacker cannot move if $s \not\rightarrow$ and $t \not\rightarrow$
   - defender cannot move if no matching transition available
2. If the game can be played *ad infinitum*, the defender wins.

Examples

Example 13.5 (Bisimulation games)

1. Use the game characterisation to show $P \sim Q$ where

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

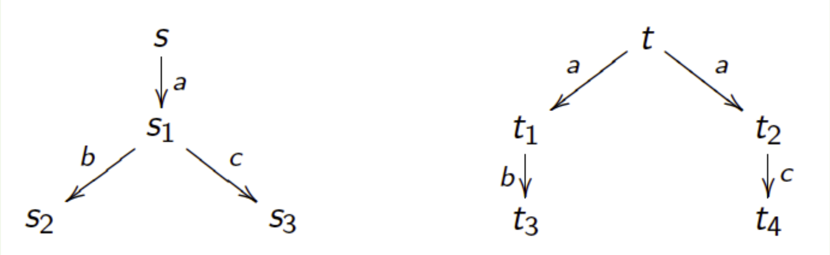
Examples

Example 13.5 (Bisimulation games)

1. Use the game characterisation to show $P \sim Q$ where

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

2. Use the game characterisation to show that $s \not\sim t$ where:



Two winning strategies for attacker in configuration (s, t):

- start with $s \xrightarrow{a} s_1$
- start with $t \xrightarrow{a} t_1$

Strong Bisimilarity as a Game

Game Characterisation of Bisimulation

Theorem 13.6 (Game characterisation of bisimulation) (Stirling 1995, Thomas 1993)

1. $s \sim t$ *iff the defender has a universal winning strategy from configuration* (s, t).
2. $s \not\sim t$ *iff the attacker has a universal winning strategy from configuration* (s, t).

(By means of a universal winning strategy, a player can always win, regardless of how the other player selects their moves.)

Game Characterisation of Bisimulation

Theorem 13.6 (Game characterisation of bisimulation) (Stirling 1995, Thomas 1993)

1. $s \sim t$ *iff the defender has a universal winning strategy from configuration* (s, t).
2. $s \not\sim t$ *iff the attacker has a universal winning strategy from configuration* (s, t).

(By means of a universal winning strategy, a player can always win, regardless of how the other player selects their moves.)

Proof.

by relating winning strategy of defender/attacker to existence/non-existence of strong bisimulation relation □

Game Characterisation of Bisimulation

Theorem 13.6 (Game characterisation of bisimulation) (Stirling 1995, Thomas 1993)

1. $s \sim t$ *iff the defender has a universal winning strategy from configuration* (s, t).
2. $s \not\sim t$ *iff the attacker has a universal winning strategy from configuration* (s, t).

(By means of a universal winning strategy, a player can always win, regardless of how the other player selects their moves.)

Proof.

by relating winning strategy of defender/attacker to existence/non-existence of strong bisimulation relation □

A bisimulation game can be used to prove bisimilarity as well as non-bisimilarity.[1] It often provides elegant arguments for $s \not\sim t$.

[1] In the following lectures, we will present yet another method to check this.

Outline of Lecture 13

Strong Simulation

Observation: sometimes, the concept of strong bisimulation is too strong (example: extending a system by new features).

Strong Simulation

Observation: sometimes, the concept of strong bisimulation is too strong (example: extending a system by new features).

Definition 13.7 (Strong simulation)

- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \xrightarrow{\alpha} P'$, there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$.

Strong Simulation

Observation: sometimes, the concept of strong bisimulation is too strong (example: extending a system by new features).

Definition 13.7 (Strong simulation)

- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \xrightarrow{\alpha} P'$, there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$.
- Q strongly simulates P, denoted $P \sqsubseteq Q$, if there exists a strong simulation ρ such that $P \,\rho\, Q$. Relation $\sqsubseteq$ is called strong similarity.

Strong Simulation

Observation: sometimes, the concept of strong bisimulation is too strong (example: extending a system by new features).

Definition 13.7 (Strong simulation)

- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \stackrel{\alpha}{\longrightarrow} P'$, there exists $Q' \in Prc$ such that $Q \stackrel{\alpha}{\longrightarrow} Q'$ and $P' \, \rho \, Q'$.
- Q strongly simulates P, denoted $P \sqsubseteq Q$, if there exists a strong simulation ρ such that $P \, \rho \, Q$. Relation $\sqsubseteq$ is called strong similarity.
- P and Q are strongly simulation equivalent if $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Strong Simulation

Observation: sometimes, the concept of strong bisimulation is too strong (example: extending a system by new features).

Definition 13.7 (Strong simulation)

- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \xrightarrow{\alpha} P'$, there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$.
- Q strongly simulates P, denoted $P \sqsubseteq Q$, if there exists a strong simulation ρ such that $P \,\rho\, Q$. Relation $\sqsubseteq$ is called strong similarity.
- P and Q are strongly simulation equivalent if $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Thus: if Q strongly simulates P, then whatever transition P takes, Q can match it with retaining all of P's options.

Strong Simulation

Observation: sometimes, the concept of strong bisimulation is too strong (example: extending a system by new features).

Definition 13.7 (Strong simulation)

- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \xrightarrow{\alpha} P'$, there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \, \rho \, Q'$.
- Q strongly simulates P, denoted $P \sqsubseteq Q$, if there exists a strong simulation ρ such that $P \, \rho \, Q$. Relation $\sqsubseteq$ is called strong similarity.
- P and Q are strongly simulation equivalent if $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Thus: if Q strongly simulates P, then whatever transition P takes, Q can match it with retaining all of P's options.

But: P does not need to be able to match each transition of Q!

Simulation: Example

Definition (Strong simulation)

- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \xrightarrow{\alpha} P'$, there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$.

Simulation: Example

Definition (Strong simulation)

- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \xrightarrow{\alpha} P'$, there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$.
- Q strongly simulates P, denoted $P \sqsubseteq Q$, if there exists a strong simulation ρ such that $P \,\rho\, Q$. Relation $\sqsubseteq$ is called strong similarity.

Simulation: Example

Definition (Strong simulation)

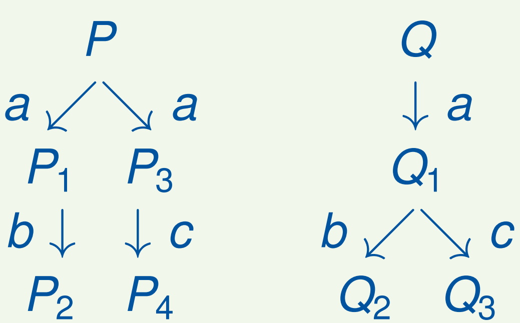
- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \xrightarrow{\alpha} P'$, there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$.
- Q strongly simulates P, denoted $P \sqsubseteq Q$, if there exists a strong simulation ρ such that $P \,\rho\, Q$. Relation $\sqsubseteq$ is called strong similarity.
- P and Q are strongly simulation equivalent if $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Simulation: Example

Definition (Strong simulation)

- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \xrightarrow{\alpha} P'$, there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \, \rho \, Q'$.
- Q strongly simulates P, denoted $P \sqsubseteq Q$, if there exists a strong simulation ρ such that $P \, \rho \, Q$. Relation $\sqsubseteq$ is called strong similarity.
- P and Q are strongly simulation equivalent if $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Example 13.8



Q strongly simulates P, but not vice versa

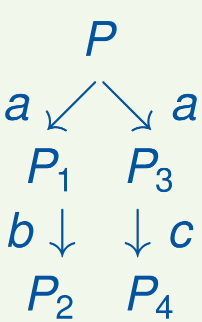
Simulation Equivalence

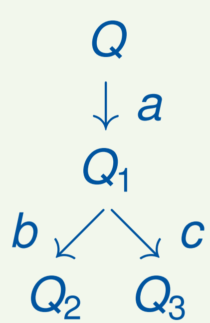
Simulation: Example

Definition (Strong simulation)

- Relation $\rho \subseteq Prc \times Prc$ is a strong simulation if, whenever $(P, Q) \in \rho$ and $P \xrightarrow{\alpha} P'$, there exists $Q' \in Prc$ such that $Q \xrightarrow{\alpha} Q'$ and $P' \,\rho\, Q'$.
- Q strongly simulates P, denoted $P \sqsubseteq Q$, if there exists a strong simulation ρ such that $P \,\rho\, Q$. Relation $\sqsubseteq$ is called strong similarity.
- P and Q are strongly simulation equivalent if $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Example 13.8





Q strongly simulates P, but not vice versa

This yields that:

$$a.b.\mathrm{nil} + a.c.\mathrm{nil} \;\sqsubseteq\; a.(b.\mathrm{nil} + c.\mathrm{nil})$$
$$a.(b.\mathrm{nil} + c.\mathrm{nil}) \;\not\sqsubseteq\; a.b.\mathrm{nil} + a.c.\mathrm{nil}.$$

Strong Simulation and Bisimilarity

Lemma 13.9 (Bisimilarity implies simulation equivalence)

If $P \sim Q$, then $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Strong Simulation and Bisimilarity

Lemma 13.9 (Bisimilarity implies simulation equivalence)

If $P \sim Q$, then $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Proof.

A strong bisimulation $\rho \subseteq Prc \times Prc$ for $P \sim Q$ is a strong simulation for both directions. □

Simulation Equivalence

Strong Simulation and Bisimilarity

Lemma 13.9 (Bisimilarity implies simulation equivalence)

If $P \sim Q$, then $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Proof.

A strong bisimulation $\rho \subseteq Prc \times Prc$ for $P \sim Q$ is a strong simulation for both directions. □

Caveat: the converse does not generally hold!

Strong Simulation and Bisimilarity

Lemma 13.9 (Bisimilarity implies simulation equivalence)

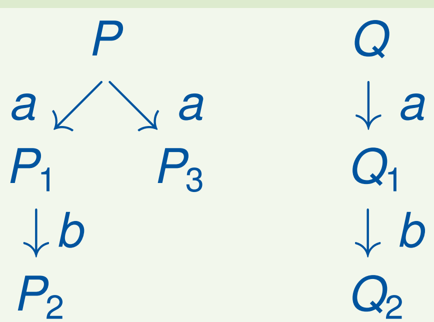
If $P \sim Q$, then $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Proof.

A strong bisimulation $\rho \subseteq Prc \times Prc$ for $P \sim Q$ is a strong simulation for both directions. □

Caveat: the converse does not generally hold!

Example 13.10



Strong Simulation and Bisimilarity

Lemma 13.9 (Bisimilarity implies simulation equivalence)

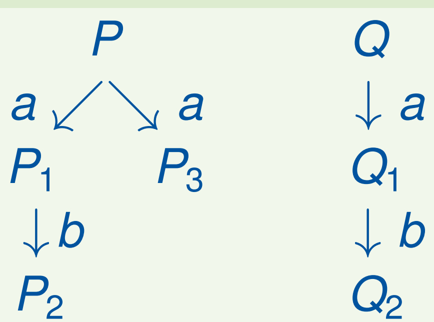
If $P \sim Q$*, then* $P \sqsubseteq Q$ *and* $Q \sqsubseteq P$.

Proof.

A strong bisimulation $\rho \subseteq Prc \times Prc$ for $P \sim Q$ is a strong simulation for both directions. $\square$

Caveat: the converse does not generally hold!

Example 13.10



$P \sqsubseteq Q$ and $Q \sqsubseteq P$,

Simulation Equivalence

Strong Simulation and Bisimilarity

Lemma 13.9 (Bisimilarity implies simulation equivalence)

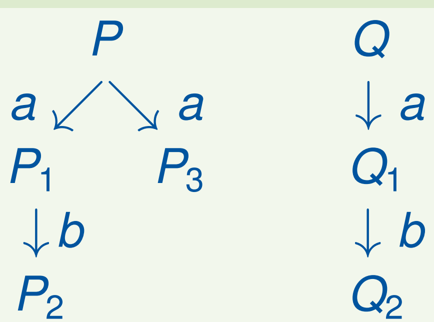
If $P \sim Q$, then $P \sqsubseteq Q$ and $Q \sqsubseteq P$.

Proof.

A strong bisimulation $\rho \subseteq Prc \times Prc$ for $P \sim Q$ is a strong simulation for both directions. □

Caveat: the converse does not generally hold!

Example 13.10



$P \sqsubseteq Q$ and $Q \sqsubseteq P$, but $P \not\sim Q$

Reason: $\sim$ allows the attacker to switch sides at each step!

Outline of Lecture 13

Summary

- Strong bisimulation of processes is based on mutually mimicking each other

Epilogue

Summary

- Strong bisimulation of processes is based on mutually mimicking each other
- Strong bisimilarity $\sim$:
  1. is the largest strong bisimulation
  2. is an equivalence relation
  3. is strictly coarser than LTS isomorphism
  4. is strictly finer than trace equivalence
  5. is a CCS congruence
  6. is deadlock sensitive
  7. can be checked using a two-player game

Epilogue

Summary

- Strong bisimulation of processes is based on mutually mimicking each other
- Strong bisimilarity $\sim$:
  1. is the largest strong bisimulation
  2. is an equivalence relation
  3. is strictly coarser than LTS isomorphism
  4. is strictly finer than trace equivalence
  5. is a CCS congruence
  6. is deadlock sensitive
  7. can be checked using a two-player game
- Strong similarity $\sqsubseteq$:
  1. is a one-way strong bisimilarity
  2. bi-directional version (strong simulation equivalence) is strictly coarser than $\sim$

Overview of Some Behavioural Equivalences