Concurrency Theory

Winter Semester 2019/20

Lecture 14: Bisimulation as a Fixed Point and Weak Variants

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

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

https://lehrpreis.fsmpi.rwth-aachen.de/

Nominate

Rules

You can nominate lecturers and assistants whose teaching you liked. The candidates should be affiliated with the department of Computer Science.

The category should match the teaching. Professors and Post-Docs, who supervise a lecture themselves, belong to the category „Selbstständige Lehre“ (independent teaching). Assistants, who take care of exercises, substitute lecturers or otherwise support teaching belong to the category „Unterstützende Lehre“ (supporting teaching).

Only those people are eligible, who did not win the award in the last two years. You can find a list of the award winners here (https://www.fsmpi.rwth-aachen.de/pages/studium/lehrpreise/informatik.html).

Thanks!

Nominate someone!

Your nomination counts for Lehrpreis Informatik / Teaching Award Computer Science 2019, currently. The results will be announced at December 6, 2019.

Nominee

Category

Selbständige Lehre

Module

Reason

☐ The nominee is eligible for the award.

Submit

Outline of Lecture 14

Recap: Strong Bisimulation

Summary

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

Recap: Strong Bisimulation

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

Recap: Strong Bisimulation

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 $\sim$

Outline of Lecture 14

Strong Bisimilarity as a Fixed Point

Strong Bisimilarity

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

Strong Bisimilarity as a Fixed Point

Strong Bisimilarity

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

What about checking $\sim$ between two processes?

Strong Bisimilarity

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

What about checking $\sim$ between two processes?

Definition (Strong bisimilarity; Definition 12.2)

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

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

Relation $\sim$ is called strong bisimilarity.

Strong Bisimilarity as a Fixed Point

Strong Bisimilarity

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

What about checking $\sim$ between two processes?

Definition (Strong bisimilarity; Definition 12.2)

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

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

Relation $\sim$ is called strong bisimilarity.

Note that $(2^{Prc \times Prc}, \subseteq)$ is a complete lattice (cf. Definition 4.13) with $\bigcup$ and $\bigcap$ as least upper and greatest lower bound, respectively.

Strong Bisimilarity

Recall: $\sim$ implies trace equivalence, and checking trace equivalence is PSPACE-complete.

What about checking $\sim$ between two processes?

Definition (Strong bisimilarity; Definition 12.2)

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

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

Relation $\sim$ is called strong bisimilarity.

Note that $(2^{Prc \times Prc}, \subseteq)$ is a complete lattice (cf. Definition 4.13) with $\bigcup$ and $\bigcap$ as least upper and greatest lower bound, respectively.

We will show that $\sim$ can be characterised as a fixed point of a monotonic function on this lattice.

Fixed-Point Characterisation of Strong Bisimilarity I

Definition 14.1 (Function on relations)

Let $\rho \subseteq Prc \times Prc$. Let $\mathcal{F} : 2^{Prc \times Prc} \to 2^{Prc \times Prc}$ be defined as follows:
for every $P, Q \in Prc$, $(P, Q) \in \mathcal{F}(\rho)$ iff

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 $P' \,\rho\, Q'$.

Fixed-Point Characterisation of Strong Bisimilarity I

Definition 14.1 (Function on relations)

Let $\rho \subseteq \mathit{Prc} \times \mathit{Prc}$. Let $\mathcal{F} : 2^{\mathit{Prc} \times \mathit{Prc}} \to 2^{\mathit{Prc} \times \mathit{Prc}}$ be defined as follows:
for every $P, Q \in \mathit{Prc}$, $(P, Q) \in \mathcal{F}(\rho)$ iff

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

Intuition: $\mathcal{F}(\rho)$ contains all pairs of processes from which, in one round of the bisimulation game, the defender can ensure that the players reach a current configuration that is contained in ρ. Note that $\mathcal{F}$ is monotonic.

Fixed-Point Characterisation of Strong Bisimilarity I

Definition 14.1 (Function on relations)

Let $\rho \subseteq Prc \times Prc$. Let $\mathcal{F} : 2^{Prc \times Prc} \to 2^{Prc \times Prc}$ be defined as follows:
for every $P, Q \in Prc$, $(P, Q) \in \mathcal{F}(\rho)$ iff

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 $P' \rho Q'$.

Intuition: $\mathcal{F}(\rho)$ contains all pairs of processes from which, in one round of the bisimulation game, the defender can ensure that the players reach a current configuration that is contained in ρ. Note that $\mathcal{F}$ is monotonic.

Corollary 14.2

ρ is a strong bisimulation iff $\rho \subseteq \mathcal{F}(\rho)$, and thus:

$$\sim = \bigcup \{\rho \in Prc \times Prc \mid \rho \subseteq \mathcal{F}(\rho)\}.$$

Fixed-Point Characterisation of Strong Bisimilarity II

Corollary

ρ is a strong bisimulation iff $\rho \subseteq \mathcal{F}(\rho)$, and thus:

$$\sim \; = \bigcup \{\rho \in \mathit{Prc} \times \mathit{Prc} \mid \rho \subseteq \mathcal{F}(\rho)\}.$$

Thus: $\sim$ is the LUB of all post-fixed points of $\mathcal{F}$

Fixed-Point Characterisation of Strong Bisimilarity II

Corollary

ρ is a strong bisimulation iff $\rho \subseteq \mathcal{F}(\rho)$, and thus:

$$\sim\; = \bigcup \{\rho \in Prc \times Prc \mid \rho \subseteq \mathcal{F}(\rho)\}.$$

Thus: $\sim$ is the LUB of all post-fixed points of $\mathcal{F}$

Theorem (Tarski's fixed-point theorem; Definition 5.5)

Let $(D, \sqsubseteq)$ be a complete lattice and $f : D \rightarrow D$ monotonic. Then f has a least fixed point $\mathrm{fix}(f)$ and a greatest fixed point $\mathrm{FIX}(f)$ given by

$$\mathrm{fix}(f) = \bigsqcap \{d \in D \mid f(d) \sqsubseteq d\} \qquad \text{(GLB of all pre-fixed points of } f\text{)}$$

$$\mathrm{FIX}(f) = \bigsqcup \{d \in D \mid d \sqsubseteq f(d)\} \qquad \text{(LUB of all post-fixed points of } f\text{)}$$

Thus: $\sim\; = \mathrm{FIX}(\mathcal{F})$

Application to Finite LTS

Theorem (Fixed-point theorem for finite lattices; Theorem 5.7)

Let $(D, \sqsubseteq)$ *be a finite complete lattice and* $f : D \to D$ *monotonic. Then*

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

for some $m, M \in \mathbb{N}$ *where* $f^0(d) := d \quad \textit{and} \quad f^{k+1}(d) := f(f^k(d))$.

Strong Bisimilarity as a Fixed Point

Application to Finite LTS

Theorem (Fixed-point theorem for finite lattices; Theorem 5.7)

Let $(D, \sqsubseteq)$ be a finite complete lattice and $f : D \to D$ monotonic. Then

$$\text{fix}(f) = f^m(\bot) \quad \textit{and} \quad \text{FIX}(f) = f^M(\top)$$

for some $m, M \in \mathbb{N}$ where $f^0(d) := d$ and $f^{k+1}(d) := f(f^k(d))$.

Corollary 14.3

For finite-state process P with state space S, $\sim$ can be computed by:

$$\begin{aligned} \sim \; &= \; \bigcap_{i=0}^{\infty} \sim_i \quad \textit{where} \\ \sim_0 \; &:= \; S \times S \\ \sim_{i+1} \; &:= \; \mathcal{F}(\sim_i) \end{aligned}$$

Application to Finite LTS

Theorem (Fixed-point theorem for finite lattices; Theorem 5.7)

Let $(D, \sqsubseteq)$ be a finite complete lattice and $f : D \to D$ monotonic. Then

$$\text{fix}(f) = f^m(\bot) \quad \text{and} \quad \text{FIX}(f) = f^M(\top)$$

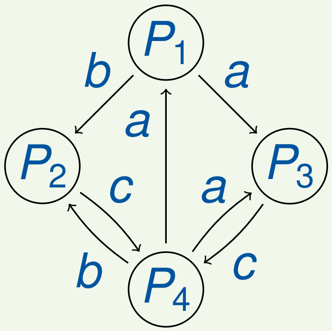
for some $m, M \in \mathbb{N}$ where $f^0(d) := d$ and $f^{k+1}(d) := f(f^k(d))$.

Corollary 14.3

For finite-state process P with state space S, $\sim$ can be computed by:

$$\sim \;=\; \bigcap_{i=0}^{\infty} \sim_i \quad \textit{where}$$
$$\sim_0 \;:=\; S \times S$$
$$\sim_{i+1} \;:=\; \mathcal{F}(\sim_i)$$

Example 14.4



Equivalence classes:

$\sim_0 \;=\; \{\{P_1, P_2, P_3, P_4\}\}$

Application to Finite LTS

Theorem (Fixed-point theorem for finite lattices; Theorem 5.7)

Let $(D, \sqsubseteq)$ *be a finite complete lattice and* $f : D \to D$ *monotonic. Then*

$$\mathrm{fix}(f) = f^m(\bot) \quad \textit{and} \quad \mathrm{FIX}(f) = f^M(\top)$$

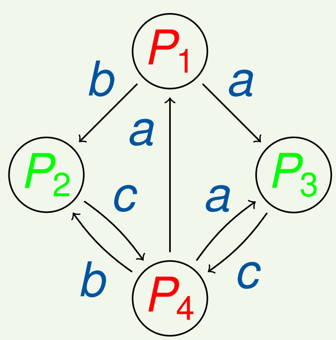
for some $m, M \in \mathbb{N}$ *where* $f^0(d) := d \quad \textit{and} \quad f^{k+1}(d) := f(f^k(d))$.

Corollary 14.3

For finite-state process P *with state space* S, $\sim$ *can be computed by:*

$$\sim \;=\; \bigcap_{i=0}^{\infty} \sim_i \quad \textit{where}$$
$$\sim_0 \;:=\; S \times S$$
$$\sim_{i+1} \;:=\; \mathcal{F}(\sim_i)$$

Example 14.4



Equivalence classes:

$$\sim_0 \;=\; \{\{P_1, P_2, P_3, P_4\}\}$$
$$\sim_1 \;=\; \{\{P_1, P_4\}, \{P_2, P_3\}\}$$

Strong Bisimilarity as a Fixed Point

Application to Finite LTS

Theorem (Fixed-point theorem for finite lattices; Theorem 5.7)

Let $(D, \sqsubseteq)$ *be a finite complete lattice and* $f : D \to D$ *monotonic. Then*

$$\text{fix}(f) = f^m(\bot) \quad \textit{and} \quad \text{FIX}(f) = f^M(\top)$$

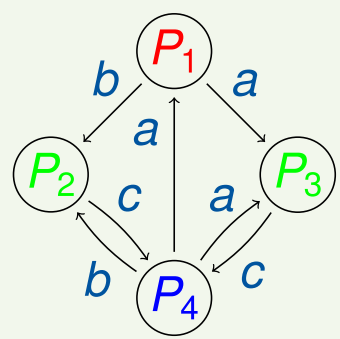
for some $m, M \in \mathbb{N}$ *where* $f^0(d) := d \quad \textit{and} \quad f^{k+1}(d) := f(f^k(d))$.

Corollary 14.3

For finite-state process P *with state space* S, $\sim$ *can be computed by:*

$$\begin{aligned} \sim \; &= \; \bigcap_{i=0}^{\infty} \sim_i \quad \textit{where} \\ \sim_0 \; &:= \; S \times S \\ \sim_{i+1} \; &:= \; \mathcal{F}(\sim_i) \end{aligned}$$

Example 14.4



Equivalence classes:

$$\begin{aligned} \sim_0 &= \{\{P_1, P_2, P_3, P_4\}\} \\ \sim_1 &= \{\{P_1, P_4\}, \{P_2, P_3\}\} \\ \sim_2 &= \{\{P_1\}, \{P_2, P_3\}, \{P_4\}\} \end{aligned}$$

Application to Finite LTS

Theorem (Fixed-point theorem for finite lattices; Theorem 5.7)

Let $(D, \sqsubseteq)$ be a finite complete lattice and $f : D \to D$ monotonic. Then

$$\text{fix}(f) = f^m(\bot) \quad \textit{and} \quad \text{FIX}(f) = f^M(\top)$$

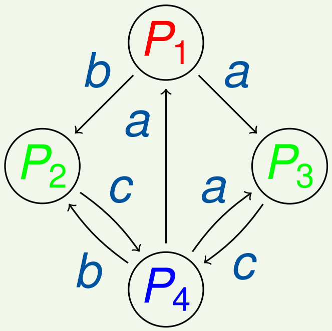
for some $m, M \in \mathbb{N}$ where $f^0(d) := d$ and $f^{k+1}(d) := f(f^k(d))$.

Corollary 14.3

For finite-state process P with state space S, $\sim$ can be computed by:

$$\begin{aligned} \sim \; &= \; \textstyle\bigcap_{i=0}^{\infty} \sim_i \quad \textit{where} \\ \sim_0 \; &:= \; S \times S \\ \sim_{i+1} \; &:= \; \mathcal{F}(\sim_i) \end{aligned}$$

Example 14.4



Equivalence classes:

$$\begin{aligned} \sim_0 &= \{\{P_1, P_2, P_3, P_4\}\} \\ \sim_1 &= \{\{P_1, P_4\}, \{P_2, P_3\}\} \\ \sim_2 &= \{\{P_1\}, \{P_2, P_3\}, \{P_4\}\} \\ \sim_3 &= \sim_2 \end{aligned}$$

Complexity of Checking Strong Bisimilarity

- The previous corollary The fixed yields a polynomial-time algorithm.
- More efficient algorithms do exist, but are not topic of this lecture.

Complexity of Checking Strong Bisimilarity

- The previous corollary The fixed yields a polynomial-time algorithm.
- More efficient algorithms do exist, but are not topic of this lecture.

Theorem 14.5 (Complexity) (Balcázar et al. 1992)

Deciding strong bisimilarity between finite LTSs is P-complete.[1]

[1] Recall that checking trace equivalence is PSPACE-complete.

Outline of Lecture 14

Inadequacy of Strong Bisimilarity

Example 14.6 (Two-place buffers; cf. Example 2.5)

1. Sequential two-place buffer:
$$\begin{aligned} B_0 &= in.B_1 \\ B_1 &= \overline{out}.B_0 + in.B_2 \\ B_2 &= \overline{out}.B_1 \end{aligned}$$
2. Parallel two-place buffer:
$$\begin{aligned} B_\parallel &= (B[f] \parallel B[g]) \setminus com \\ B &= in.\overline{out}.B \end{aligned}$$
$(f := [out \mapsto com], g := [in \mapsto com])$

Inadequacy of Strong Bisimilarity

Example 14.6 (Two-place buffers; cf. Example 2.5)

1. Sequential two-place buffer:

$$B_0 = in.B_1$$
$$B_1 = \overline{out}.B_0 + in.B_2$$
$$B_2 = \overline{out}.B_1$$

2. Parallel two-place buffer:

$$B_\| = (B[f] \parallel B[g]) \setminus com$$
$$B = in.\overline{out}.B$$

$(f := [out \mapsto com], g := [in \mapsto com])$

Observation:

Inadequacy of Strong Bisimilarity

Example 14.6 (Two-place buffers; cf. Example 2.5)

1. Sequential two-place buffer:
$$\begin{aligned} B_0 &= in.B_1 \\ B_1 &= \overline{out}.B_0 + in.B_2 \\ B_2 &= \overline{out}.B_1 \end{aligned}$$
2. Parallel two-place buffer:
$$\begin{aligned} B_\parallel &= (B[f] \parallel B[g]) \setminus com \\ B &= in.\overline{out}.B \end{aligned}$$
$(f := [out \mapsto com], g := [in \mapsto com])$

Observation:

Conclusion

- The requirement in $\sim$ to exactly match all actions is often too strong.
- This suggests to weaken this and not insist on exact matching of τ-actions.
- Rationale: τ-actions are special as they are unobservable.

Inadequacy of Strong Bisimilarity

The Rationales for Abstracting from τ-Actions

- τ-actions are internal and thus unobservable.

The Rationales for Abstracting from τ-Actions

- τ-actions are internal and thus unobservable.
- This is natural in parallel communication resulting in τ:
  - synchronization in CCS is binary handshaking
  - observation means communication with the process
  - thus the result of any communication is unobservable

Inadequacy of Strong Bisimilarity

The Rationales for Abstracting from τ-Actions

- τ-actions are internal and thus unobservable.
- This is natural in parallel communication resulting in τ:
  - synchronization in CCS is binary handshaking
  - observation means communication with the process
  - thus the result of any communication is unobservable
- Strong bisimilarity treats τ-actions as any other action.

Inadequacy of Strong Bisimilarity

The Rationales for Abstracting from τ-Actions

- τ-actions are internal and thus unobservable.
- This is natural in parallel communication resulting in τ:
  - synchronization in CCS is binary handshaking
  - observation means communication with the process
  - thus the result of any communication is unobservable
- Strong bisimilarity treats τ-actions as any other action.
- Can we retain the nice properties of $\sim$ while “abstracting” from τ-actions?

Outline of Lecture 14

Weak Transition Relation

Definition 14.7 (Weak transition relation)

For $\alpha \in Act$, $\stackrel{\alpha}{\Longrightarrow} \subseteq Prc \times Prc$ is given by

$$\stackrel{\alpha}{\Longrightarrow} := \begin{cases} \left(\stackrel{\tau}{\longrightarrow}\right)^* \circ \stackrel{\alpha}{\longrightarrow} \circ \left(\stackrel{\tau}{\longrightarrow}\right)^* & \text{if } \alpha \neq \tau \\ \left(\stackrel{\tau}{\longrightarrow}\right)^* & \text{if } \alpha = \tau. \end{cases}$$

where $\left(\stackrel{\tau}{\longrightarrow}\right)^*$ denotes the reflexive and transitive closure of relation $\stackrel{\tau}{\longrightarrow}$.

Weak Transition Relation

Definition 14.7 (Weak transition relation)

For $\alpha \in Act$, $\stackrel{\alpha}{\Longrightarrow} \subseteq Prc \times Prc$ is given by

$$\stackrel{\alpha}{\Longrightarrow} := \begin{cases} \left(\stackrel{\tau}{\longrightarrow}\right)^* \circ \stackrel{\alpha}{\longrightarrow} \circ \left(\stackrel{\tau}{\longrightarrow}\right)^* & \text{if } \alpha \neq \tau \\ \left(\stackrel{\tau}{\longrightarrow}\right)^* & \text{if } \alpha = \tau. \end{cases}$$

where $\left(\stackrel{\tau}{\longrightarrow}\right)^*$ denotes the reflexive and transitive closure of relation $\stackrel{\tau}{\longrightarrow}$.

Informal meaning

- If $\alpha \neq \tau$, then $s \stackrel{\alpha}{\Longrightarrow} t$ means that from s we can get to t by doing zero or more τ actions, followed by the action α, followed by zero or more τ actions.

Weak Transition Relation

Definition 14.7 (Weak transition relation)

For $\alpha \in Act$, $\stackrel{\alpha}{\Longrightarrow} \subseteq Prc \times Prc$ is given by

$$\stackrel{\alpha}{\Longrightarrow} := \begin{cases} \left(\stackrel{\tau}{\longrightarrow}\right)^* \circ \stackrel{\alpha}{\longrightarrow} \circ \left(\stackrel{\tau}{\longrightarrow}\right)^* & \text{if } \alpha \neq \tau \\ \left(\stackrel{\tau}{\longrightarrow}\right)^* & \text{if } \alpha = \tau. \end{cases}$$

where $\left(\stackrel{\tau}{\longrightarrow}\right)^*$ denotes the reflexive and transitive closure of relation $\stackrel{\tau}{\longrightarrow}$.

Informal meaning

- If $\alpha \neq \tau$, then $s \stackrel{\alpha}{\Longrightarrow} t$ means that from s we can get to t by doing zero or more τ actions, followed by the action α, followed by zero or more τ actions.
- If $\alpha = \tau$, then $s \stackrel{\alpha}{\Longrightarrow} t$ means that from s we can reach t by doing zero or more τ actions.

Weak Bisimulation

Definition 14.8 (Weak bisimulation) (Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a weak bisimulation whenever for every $(P, Q) \in \rho$ and $\alpha \in Act$ (including $\alpha = \tau$):

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

Weak Bisimulation

Definition 14.8 (Weak bisimulation) (Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a weak bisimulation whenever for every $(P, Q) \in \rho$ and $\alpha \in Act$ (including $\alpha = \tau$):

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

Definition 14.9 (Weak bisimilarity)

Processes P and Q are weakly bisimilar, denoted $P \approx Q$, iff there is a weak bisimulation ρ with $P \rho Q$.

Weak Bisimulation

Definition 14.8 (Weak bisimulation) (Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a weak bisimulation whenever for every $(P, Q) \in \rho$ and $\alpha \in Act$ (including $\alpha = \tau$):

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

Definition 14.9 (Weak bisimilarity)

Processes P and Q are weakly bisimilar, denoted $P \approx Q$, iff there is a weak bisimulation ρ with $P \,\rho\, Q$. Thus,

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

Weak Bisimulation

Definition 14.8 (Weak bisimulation) (Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a weak bisimulation whenever for every $(P, Q) \in \rho$ and $\alpha \in Act$ (including $\alpha = \tau$):

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

Definition 14.9 (Weak bisimilarity)

Processes P and Q are weakly bisimilar, denoted $P \approx Q$, iff there is a weak bisimulation ρ with $P \,\rho\, Q$. Thus,

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

Relation $\approx$ is called observational equivalence or weak bisimilarity.

Explanation

Definition (Weak bisimulation) (Milner 1989)

A binary relation $\rho \subseteq Prc \times Prc$ is a weak bisimulation whenever for every $(P, Q) \in \rho$ and $\alpha \in Act$ (including $\alpha = \tau$):

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

Remark

Each clause in the definition of weak bisimulation subsumes two cases:

- $P \xrightarrow{\alpha} P'$ where $\alpha \neq \tau$:
  implies ex. $Q' \in Prc$ such that $Q\, (\xrightarrow{\tau})^* \xrightarrow{\alpha} (\xrightarrow{\tau})^*\, Q'$ and $P' \rho Q'$
- $P \xrightarrow{\tau} P'$:
  implies ex. $Q' \in Prc$ such that $Q\, (\xrightarrow{\tau})^*\, Q'$ and $P' \rho Q'$
  (where $Q' = Q$ is admissible)

Examples

Example 14.10

1. Let $P = \tau.Q$ with $Q = a.\text{nil}$.
   - obviously $P \not\sim Q$; claim: $P \approx Q$
   - proof: $\rho = \{(P, Q), (Q, Q), (\text{nil}, \text{nil})\}$ is a weak bisimulation with $P \, \rho \, Q$

Examples

Example 14.10

1. Let $P = \tau.Q$ with $Q = a.\mathrm{nil}$.
   - obviously $P \not\sim Q$; claim: $P \approx Q$
   - proof: $\rho = \{(P, Q), (Q, Q), (\mathrm{nil}, \mathrm{nil})\}$ is a weak bisimulation with $P \,\rho\, Q$
2. More general: for every $P \in Prc$, $P \approx \tau.P$.
   Proof: $\rho = \{(P, \tau.P)\} \cup id_{Prc}$ is a weak bisimulation:
   i. every transition $P \xrightarrow{\alpha} P'$ can be simulated by $\tau.P \xrightarrow{\tau} P \xrightarrow{\alpha} P'$ (i.e., $\tau.P \xRightarrow{\alpha} P'$) with $P' \,\rho\, P' \in \rho$ (since $id_{Prc} \subseteq \rho$)

Examples

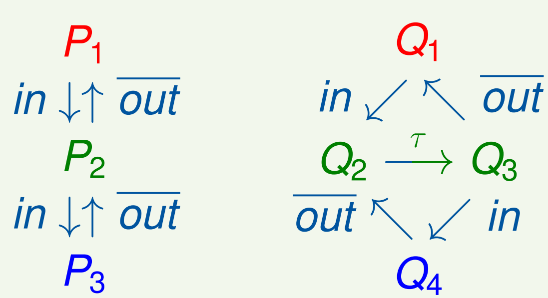
Example 14.10

1. Let $P = \tau.Q$ with $Q = a.\mathrm{nil}$.
   - obviously $P \not\sim Q$; claim: $P \approx Q$
   - proof: $\rho = \{(P, Q), (Q, Q), (\mathrm{nil}, \mathrm{nil})\}$ is a weak bisimulation with $P \,\rho\, Q$
2. More general: for every $P \in Prc$, $P \approx \tau.P$.
   Proof: $\rho = \{(P, \tau.P)\} \cup id_{Prc}$ is a weak bisimulation:
   1. every transition $P \xrightarrow{\alpha} P'$ can be simulated by $\tau.P \xrightarrow{\tau} P \xrightarrow{\alpha} P'$ (i.e., $\tau.P \xRightarrow{\alpha} P'$) with $P' \,\rho\, P' \in \rho$ (since $id_{Prc} \subseteq \rho$)
   2. the only transition of $\tau.P$ is $\tau.P \xrightarrow{\tau} P$; it is simulated by $P \xRightarrow{\tau} P$ with $P \,\rho\, P \in \rho$ (since $id_{Prc} \subseteq \rho$)

Examples

Example 14.10

1. Let $P = \tau.Q$ with $Q = a.\mathrm{nil}$.
   - obviously $P \not\sim Q$; claim: $P \approx Q$
   - proof: $\rho = \{(P, Q), (Q, Q), (\mathrm{nil}, \mathrm{nil})\}$ is a weak bisimulation with $P \, \rho \, Q$
2. More general: for every $P \in Prc$, $P \approx \tau.P$.
   Proof: $\rho = \{(P, \tau.P)\} \cup id_{Prc}$ is a weak bisimulation:
   i. every transition $P \xrightarrow{\alpha} P'$ can be simulated by $\tau.P \xrightarrow{\tau} P \xrightarrow{\alpha} P'$ (i.e., $\tau.P \stackrel{\alpha}{\Longrightarrow} P'$) with $P' \, \rho \, P' \in \rho$ (since $id_{Prc} \subseteq \rho$)
   ii. the only transition of $\tau.P$ is $\tau.P \xrightarrow{\tau} P$; it is simulated by $P \stackrel{\tau}{\Longrightarrow} P$ with $P \, \rho \, P \in \rho$ (since $id_{Prc} \subseteq \rho$)
3. Sequential and parallel two-place buffer are weakly bisimilar:



$$\rho = \{(P_1, Q_1), (P_2, Q_2), (P_2, Q_3), (P_3, Q_4)\}$$

Outline of Lecture 14

Divergence

Example 14.11 (A polling process) (Koomen 1982)

$$\begin{aligned} A? &= a.\mathsf{nil} + \tau.B? \\ B? &= b.\mathsf{nil} + \tau.A? \end{aligned}$$

Divergence

Example 14.11 (A polling process) (Koomen 1982)

$$\begin{aligned} A? &= a.\mathrm{nil} + \tau.B? \\ B? &= b.\mathrm{nil} + \tau.A? \end{aligned}$$

- Claim: $A? \approx B? \approx a.\mathrm{nil} + b.\mathrm{nil}$

Divergence

Example 14.11 (A polling process) (Koomen 1982)

$$\begin{aligned} A? &= a.\text{nil} + \tau.B? \\ B? &= b.\text{nil} + \tau.A? \end{aligned}$$

- Claim: $A? \approx B? \approx a.\text{nil} + b.\text{nil}$
- But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ is a τ-loop, whereas $a.\text{nil} + b.\text{nil}$ does not have a loop (not even a τ-loop).

Divergence

Example 14.11 (A polling process) (Koomen 1982)

$$A? = a.\mathsf{nil} + \tau.B?$$
$$B? = b.\mathsf{nil} + \tau.A?$$

- Claim: $A? \approx B? \approx a.\mathsf{nil} + b.\mathsf{nil}$
- But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ is a τ-loop, whereas $a.\mathsf{nil} + b.\mathsf{nil}$ does not have a loop (not even a τ-loop).
- Thus, $\approx$ assumes that if a process can escape from a τ-loop, it eventually will do so.[2] Divergence is a τ-loop.

[2] This is called fair abstraction from divergence.

Divergence

Example 14.11 (A polling process) (Koomen 1982)

$$A? = a.\mathrm{nil} + \tau.B?$$
$$B? = b.\mathrm{nil} + \tau.A?$$

- Claim: $A? \approx B? \approx a.\mathrm{nil} + b.\mathrm{nil}$
- But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ is a τ-loop, whereas $a.\mathrm{nil} + b.\mathrm{nil}$ does not have a loop (not even a τ-loop).
- Thus, $\approx$ assumes that if a process can escape from a τ-loop, it eventually will do so.[2] Divergence is a τ-loop.
- Also note that $Div \approx \mathrm{nil}$ where $Div = \tau.Div$.

[2] This is called fair abstraction from divergence.

Divergence

Example 14.11 (A polling process) (Koomen 1982)

$$A? = a.\mathsf{nil} + \tau.B?$$
$$B? = b.\mathsf{nil} + \tau.A?$$

- Claim: $A? \approx B? \approx a.\mathsf{nil} + b.\mathsf{nil}$
- But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ is a τ-loop, whereas $a.\mathsf{nil} + b.\mathsf{nil}$ does not have a loop (not even a τ-loop).
- Thus, $\approx$ assumes that if a process can escape from a τ-loop, it eventually will do so.[2] Divergence is a τ-loop.
- Also note that $Div \approx \mathsf{nil}$ where $Div = \tau.Div$.
- Thus, a deadlock process is weakly bisimilar to a process that can only diverge.

[2] This is called fair abstraction from divergence.

Divergence

Example 14.11 (A polling process) (Koomen 1982)

$$A? = a.\mathrm{nil} + \tau.B?$$
$$B? = b.\mathrm{nil} + \tau.A?$$

- Claim: $A? \approx B? \approx a.\mathrm{nil} + b.\mathrm{nil}$
- But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ is a τ-loop, whereas $a.\mathrm{nil} + b.\mathrm{nil}$ does not have a loop (not even a τ-loop).
- Thus, $\approx$ assumes that if a process can escape from a τ-loop, it eventually will do so.[2] Divergence is a τ-loop.
- Also note that $Div \approx \mathrm{nil}$ where $Div = \tau.Div$.
- Thus, a deadlock process is weakly bisimilar to a process that can only diverge.
- This is justified by the fact that “observations” can only be made by interacting with the process.

[2] This is called fair abstraction from divergence.

Properties of Weak Bisimilarity

Lemma 14.12 (Properties of $\approx$)

1. $P \sim Q$ *implies* $P \approx Q$.
2. $\approx$ *is an equivalence relation (reflexive, symmetric, transitive).*
3. $\approx$ *is the largest weak bisimulation.*
4. $\approx$ *is (non-τ) deadlock sensitive.*[3]
5. $\approx$ *abstracts from τ-loops.*

[3] Where w-deadlocks are considered on observable traces – see following slide.

Properties of Weak Bisimilarity

Lemma 14.12 (Properties of $\approx$)

1. *$P \sim Q$ implies $P \approx Q$.*
2. *$\approx$ is an equivalence relation (reflexive, symmetric, transitive).*
3. *$\approx$ is the largest weak bisimulation.*
4. *$\approx$ is (non-τ) deadlock sensitive.*[3]
5. *$\approx$ abstracts from τ-loops.*

Proof.

1. Straightforward (as $\xrightarrow{\alpha} \subseteq \xRightarrow{\alpha}$)
2. Similar to Lemma 12.6(1) for $\sim$
3. Similar to Lemma 12.6(2) for $\sim$
4. Similar to Theorem 13.1 for $\sim$
5. Previous slide

□

[3] Where *w*-deadlocks are considered on observable traces – see following slide.

Weak Bisimilarity vs. Trace Equivalence

Definition 14.13 (Observational trace language)

The observational trace language of $P \in Prc$ is defined by:

$$ObsTr(P) := \{\widehat{w} \in (Act \setminus \{\tau\})^* \mid \exists P' \in Prc.\, P \xrightarrow{w} P'\}$$

where $\widehat{w}$ is obtained from w by removing all τ-actions.

Weak Bisimilarity vs. Trace Equivalence

Definition 14.13 (Observational trace language)

The observational trace language of $P \in Prc$ is defined by:

$$ObsTr(P) := \{\widehat{w} \in (Act \setminus \{\tau\})^* \mid \exists P' \in Prc.\ P \overset{w}{\longrightarrow} P'\}$$

where $\widehat{w}$ is obtained from w by removing all τ-actions.

Definition 14.14 (Observational trace equivalence)

$P, Q \in Prc$ are observational trace equivalent if $ObsTr(P) = ObsTr(Q)$.

Weak Bisimilarity vs. Trace Equivalence

Definition 14.13 (Observational trace language)

The observational trace language of $P \in Prc$ is defined by:

$$ObsTr(P) := \{\widehat{w} \in (Act \setminus \{\tau\})^* \mid \exists P' \in Prc. P \xrightarrow{w} P'\}$$

where $\widehat{w}$ is obtained from w by removing all τ-actions.

Definition 14.14 (Observational trace equivalence)

$P, Q \in Prc$ are observational trace equivalent if $ObsTr(P) = ObsTr(Q)$.

Theorem 14.15

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

Properties of Weak Bisimilarity

Weak Bisimilarity vs. Trace Equivalence

Definition 14.13 (Observational trace language)

The observational trace language of $P \in Prc$ is defined by:

$$ObsTr(P) := \{\widehat{w} \in (Act \setminus \{\tau\})^* \mid \exists P' \in Prc.\ P \overset{w}{\longrightarrow} P'\}$$

where $\widehat{w}$ is obtained from w by removing all τ-actions.

Definition 14.14 (Observational trace equivalence)

$P, Q \in Prc$ are observational trace equivalent if $ObsTr(P) = ObsTr(Q)$.

Theorem 14.15

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

Proof.

similar to Theorem 12.8 □

Properties of Weak Bisimilarity

Milner's τ-Laws

Lemma 14.16 (Milner's τ-laws)

$$\begin{aligned} \alpha.\tau.P &\approx \alpha.P \\ P + \tau.P &\approx \tau.P \\ \alpha.(P + \tau.Q) &\approx \alpha.(P + \tau.Q) + \alpha.Q \end{aligned}$$

Milner's τ-Laws

Lemma 14.16 (Milner's τ-laws)

$$
\begin{aligned}
\alpha.\tau.P &\approx \alpha.P \\
P + \tau.P &\approx \tau.P \\
\alpha.(P + \tau.Q) &\approx \alpha.(P + \tau.Q) + \alpha.Q
\end{aligned}
$$

Proof.

by constructing appropriate weak bisimulation relations (left as an exercise) □

Properties of Weak Bisimilarity

Congruence

Lemma 14.17 (Partial CCS congruence property of $\approx$)

Whenever $P, Q \in Prc$ *such that* $P \approx Q$,

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

Properties of Weak Bisimilarity

Congruence

Lemma 14.17 (Partial CCS congruence property of $\approx$)

Whenever $P, Q \in Prc$ *such that* $P \approx Q$,

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

Proof.

omitted □

Properties of Weak Bisimilarity

Congruence

Lemma 14.17 (Partial CCS congruence property of $\approx$)

Whenever $P, Q \in Prc$ such that $P \approx Q$,

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

Proof.

omitted □

What about choice?

- $\tau.a.\text{nil} \approx a.\text{nil}$ (cf. Ex. 14.10(1)) and $b.\text{nil} \approx b.\text{nil}$ (reflexivity)

Properties of Weak Bisimilarity

Congruence

Lemma 14.17 (Partial CCS congruence property of $\approx$)

Whenever $P, Q \in Prc$ such that $P \approx Q$,

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

Proof.

omitted □

What about choice?

- $\tau.a.\text{nil} \approx a.\text{nil}$ (cf. Ex. 14.10(1)) and $b.\text{nil} \approx b.\text{nil}$ (reflexivity)
- but $\tau.a.\text{nil} + b.\text{nil} \not\approx a.\text{nil} + b.\text{nil}$ (why?).

Properties of Weak Bisimilarity

Congruence

Lemma 14.17 (Partial CCS congruence property of $\approx$)

Whenever $P, Q \in Prc$ such that $P \approx Q$,

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

Proof.

omitted □

What about choice?

- $\tau.a.\mathrm{nil} \approx a.\mathrm{nil}$ (cf. Ex. 14.10(1)) and $b.\mathrm{nil} \approx b.\mathrm{nil}$ (reflexivity)
- but $\tau.a.\mathrm{nil} + b.\mathrm{nil} \not\approx a.\mathrm{nil} + b.\mathrm{nil}$ (why?).
- Thus, weak bisimilarity is not a CCS congruence, which motivates a slight adaptation of $\approx$.

Outline of Lecture 14

Observation Congruence

Definition 14.18 (Observation congruence) (Milner 1989)

$P, Q \in \mathit{Prc}$ are observationally congruent, denoted $P \approx^c Q$, if for every $\alpha \in \mathit{Act}$ (including $\alpha = \tau$):

1. if $P \xrightarrow{\alpha} P'$, then there is a sequence of transitions $Q \overset{\tau}{\Longrightarrow} \circ \xrightarrow{\alpha} \circ \overset{\tau}{\Longrightarrow} Q'$ such that $P' \approx Q'$ and
2. if $Q \xrightarrow{\alpha} Q'$, then there is a sequence of transitions $P \overset{\tau}{\Longrightarrow} \circ \xrightarrow{\alpha} \circ \overset{\tau}{\Longrightarrow} P'$ such that $P' \approx Q'$.

Observation Congruence

Definition 14.18 (Observation congruence) (Milner 1989)

$P, Q \in \mathit{Prc}$ are observationally congruent, denoted $P \approx^c Q$, if for every $\alpha \in \mathit{Act}$ (including $\alpha = \tau$):

1. if $P \xrightarrow{\alpha} P'$, then there is a sequence of transitions $Q \overset{\tau}{\Longrightarrow} \circ \xrightarrow{\alpha} \circ \overset{\tau}{\Longrightarrow} Q'$ such that $P' \approx Q'$ and
2. if $Q \xrightarrow{\alpha} Q'$, then there is a sequence of transitions $P \overset{\tau}{\Longrightarrow} \circ \xrightarrow{\alpha} \circ \overset{\tau}{\Longrightarrow} P'$ such that $P' \approx Q'$.

Remark

- $\approx^c$ differs from $\approx$ only in that $\approx^c$ requires τ-moves by P or Q to be mimicked by at least one τ-move in the other process.
- This only applies to the first step; the successors just have to satisfy $P' \approx Q'$ (and not necessarily $P' \approx^c Q'$).

Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Observation Congruence

Definition 14.18 (Observation congruence) (Milner 1989)

$P, Q \in \mathit{Prc}$ are observationally congruent, denoted $P \approx^c Q$, if for every $\alpha \in \mathit{Act}$ (including $\alpha = \tau$):

1. if $P \xrightarrow{\alpha} P'$, then there is a sequence of transitions $Q \stackrel{\tau}{\Longrightarrow} \circ \xrightarrow{\alpha} \circ \stackrel{\tau}{\Longrightarrow} Q'$ such that $P' \approx Q'$ and
2. if $Q \xrightarrow{\alpha} Q'$, then there is a sequence of transitions $P \stackrel{\tau}{\Longrightarrow} \circ \xrightarrow{\alpha} \circ \stackrel{\tau}{\Longrightarrow} P'$ such that $P' \approx Q'$.

Remark

- $\approx^c$ differs from $\approx$ only in that $\approx^c$ requires τ-moves by P or Q to be mimicked by at least one τ-move in the other process.
- This only applies to the first step; the successors just have to satisfy $P' \approx Q'$ (and not necessarily $P' \approx^c Q'$).
- Thus: if $P \stackrel{\tau}{\not\longrightarrow}$ and $Q \stackrel{\tau}{\not\longrightarrow}$, then $P \approx^c Q$ iff $P \approx Q$.

Examples

Example 14.19

1. Sequential and parallel two-place buffer:

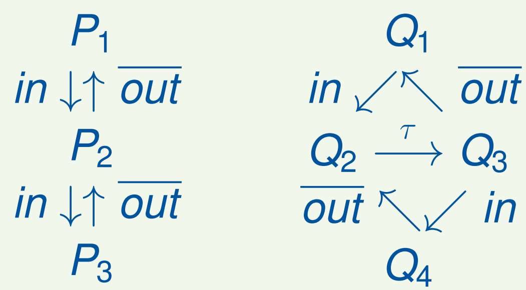
$$
\begin{array}{c}
P_1 \\
in \downarrow\uparrow \overline{out} \\
P_2 \\
in \downarrow\uparrow \overline{out} \\
P_3
\end{array}
\qquad\qquad
\begin{array}{ccccc}
 & & Q_1 & & \\
 & in \swarrow & & \nwarrow \overline{out} & \\
Q_2 & & \xrightarrow{\tau} & & Q_3 \\
 & \overline{out} \nwarrow & & \swarrow in & \\
 & & Q_4 & &
\end{array}
$$

$P_1 \approx^c Q_1$ since $P_1 \approx Q_1$ (cf. Example 14.10(3)) and neither P_1 nor Q_1 has initial τ-steps.

Examples

Example 14.19

1. Sequential and parallel two-place buffer:



   $P_1 \approx^c Q_1$ since $P_1 \approx Q_1$ (cf. Example 14.10(3)) and neither P_1 nor Q_1 has initial τ-steps.

2. $\tau.a.\text{nil} \not\approx^c a.\text{nil}$ (since $\tau.a.\text{nil} \xrightarrow{\tau}$ but $a.\text{nil} \not\xrightarrow{\tau}$);
   thus the counterexample to congruence of $\approx$ for $+$ does not apply.

Observation Congruence

Examples

Example 14.19

1. Sequential and parallel two-place buffer:

$$\begin{array}{ccc} & P_1 & \\ in & \downarrow\uparrow & \overline{out} \\ & P_2 & \\ in & \downarrow\uparrow & \overline{out} \\ & P_3 & \end{array} \qquad\qquad \begin{array}{ccccc} & & Q_1 & & \\ & in \swarrow & & \nwarrow \overline{out} & \\ Q_2 & & \xrightarrow{\tau} & & Q_3 \\ & \overline{out} \nwarrow & & \swarrow in & \\ & & Q_4 & & \end{array}$$

   $P_1 \approx^c Q_1$ since $P_1 \approx Q_1$ (cf. Example 14.10(3)) and neither P_1 nor Q_1 has initial τ-steps.
2. $\tau.a.\text{nil} \not\approx^c a.\text{nil}$ (since $\tau.a.\text{nil} \xrightarrow{\tau}$ but $a.\text{nil} \not\xrightarrow{\tau}$);
   thus the counterexample to congruence of $\approx$ for $+$ does not apply.
3. $a.\tau.\text{nil} \approx^c a.\text{nil}$ (since $\tau.\text{nil} \approx \text{nil}$).

Properties of Observation Congruence

Lemma 14.20

For every $P, Q \in Prc$,

1. $\approx^c$ *is an equivalence relation*
2. $P \sim Q$ *implies* $P \approx^c Q$, *and* $P \approx^c Q$ *implies* $P \approx Q$
3. $\approx^c$ *is a CCS congruence*
4. $\approx^c$ *is (non-τ) deadlock-sensitive*
5. $P \approx^c Q$ *if and only if* $P + R \approx Q + R$ *for every* $R \in Prc$
6. $P \approx Q$ *if and only if* ($P \approx^c Q$ *or* $P \approx^c \tau.Q$ *or* $\tau.P \approx^c Q$)

Proof.

omitted □

Observation Congruence

Properties of Observation Congruence

Lemma 14.20

For every $P, Q \in Prc$,

1. $\approx^c$ *is an equivalence relation*
2. $P \sim Q$ *implies* $P \approx^c Q$, *and* $P \approx^c Q$ *implies* $P \approx Q$
3. $\approx^c$ *is a CCS congruence*
4. $\approx^c$ *is (non-τ) deadlock-sensitive*
5. $P \approx^c Q$ *if and only if* $P + R \approx Q + R$ *for every* $R \in Prc$
6. $P \approx Q$ *if and only if* ($P \approx^c Q$ *or* $P \approx^c \tau.Q$ *or* $\tau.P \approx^c Q$)

Proof.

omitted □

Note: (5) states that $\approx^c$ is the "minimal fix" to establish congruence of $\approx$.

Outline of Lecture 14

Weak Bisimilarity as a Game

Rules

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

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

Weak Bisimilarity as a Game

Rules

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

1. the attacker chooses one of the 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 $s \xRightarrow{\alpha} s'$.

Weak Bisimilarity as a Game

Rules

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

1. the attacker chooses one of the 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 $\stackrel{\alpha}{\Longrightarrow}$-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.

Weak Bisimilarity as a Game

Rules

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

1. the attacker chooses one of the 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 $\stackrel{\alpha}{\Longrightarrow}$-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
2. If the game can be played *ad infinitum*, the defender wins.

Game Characterisation of Weak Bisimilarity

Theorem 14.21 (Game characterisation of weak bisimilarity) (Stirling 1995, Thomas 1993)

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

Theorem 14.21 (Game characterisation of weak bisimilarity) (Stirling 1995, Thomas 1993)

1. $s \approx t$ *iff the defender has a universal winning strategy from configuration* (s, t).
2. $s \not\approx t$ *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 weak bisimulation relation □