Concurrency Theory

Winter 2025/26

Lecture 6: Properties of Weak 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 6

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 is played *ad infinitum*, the defender wins.

Game Characterisation of Bisimulation

Theorem (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 □

Thus, 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] Later we will present yet another method to check this.

Strong Simulation

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

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 $Q \sqsubseteq P$.

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

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

Strong Simulation and Bisimilarity

Lemma (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!

Weak Bisimulation

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

Definition (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 weak bisimilarity or observational equivalence.

Outline of Lecture 6

Divergence I

Example 6.1 (A polling process) (Koomen 1982)

A process that is willing to receive on port a or port b, and then terminates:

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

Divergence I

Example 6.1 (A polling process) (Koomen 1982)

A process that is willing to receive on port a or port b, and then terminates:

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

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

Divergence I

Example 6.1 (A polling process) (Koomen 1982)

A process that is willing to receive on port a or port b, and then terminates:

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

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

Divergence I

Example 6.1 (A polling process) (Koomen 1982)

A process that is willing to receive on port a or port b, and then terminates:

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

- Claim: $A? \approx B? \approx a.\text{nil} + b.\text{nil}$
- But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ forms a τ-loop, whereas $a.\text{nil} + b.\text{nil}$ 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.[a] Divergence (“livelock”) is a τ-loop.

Example 6.1 (A polling process) (Koomen 1982)

A process that is willing to receive on port a or port b, and then terminates:

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

- Claim: $A? \approx B? \approx a.\mathrm{nil} + b.\mathrm{nil}$
- But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ forms 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.[a] Divergence (“livelock”) is a τ-loop.
- Also note that $Div \approx \mathrm{nil}$ where $Div = \tau.Div$.

Divergence I

Example 6.1 (A polling process) (Koomen 1982)

A process that is willing to receive on port a or port b, and then terminates:

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

- Claim: $A? \approx B? \approx a.\text{nil} + b.\text{nil}$
- But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ forms a τ-loop, whereas $a.\text{nil} + b.\text{nil}$ 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.[a] Divergence (“livelock”) is a τ-loop.
- Also note that $Div \approx \text{nil}$ where $Div = \tau.Div$.
- Thus, a deadlocked process is weakly bisimilar to a process that can only diverge.

Divergence I

Example 6.1 (A polling process) (Koomen 1982)

A process that is willing to receive on port a or port b, and then terminates:

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

- Claim: $A? \approx B? \approx a.\mathrm{nil} + b.\mathrm{nil}$
- But note that $A? \xrightarrow{\tau} B? \xrightarrow{\tau} A?$ forms 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.[a] Divergence ("livelock") is a τ-loop.
- Also note that $Div \approx \mathrm{nil}$ where $Div = \tau.Div$.
- Thus, a deadlocked 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.

Divergence II

Example 6.2 (A simple communication protocol)

Observation: (unbounded) retransmission after communication failures can also cause τ-loops.

$$
\begin{aligned}
\mathit{Spec} &= \mathit{acc}.\overline{\mathit{del}}.\mathit{Spec};\\
\mathit{Impl} &= (\mathit{Send} \parallel \mathit{Med} \parallel \mathit{Rec}) \setminus \{\mathit{send}, \mathit{trans}, \mathit{ack}, \mathit{error}\};\\
\mathit{Send} &= \mathit{acc}.\mathit{Sending};\\
\mathit{Sending} &= \overline{\mathit{send}}.\mathit{Wait};\\
\mathit{Wait} &= \mathit{ack}.\mathit{Send} + \mathit{error}.\mathit{Sending};\\
\mathit{Med} &= \mathit{send}.\mathit{Trans};\\
\mathit{Trans} &= \overline{\mathit{trans}}.\mathit{Med} + \tau.\mathit{Err};\\
\mathit{Err} &= \overline{\mathit{error}}.\mathit{Med};\\
\mathit{Rec} &= \mathit{trans}.\mathit{Del};\\
\mathit{Del} &= \overline{\mathit{del}}.\mathit{Ack};\\
\mathit{Ack} &= \overline{\mathit{ack}}.\mathit{Rec};
\end{aligned}
$$

(analyse using CAAL tool)

Properties of Weak Bisimilarity

Lemma 6.3 (Properties of $\approx$)

(1) $P \sim Q$ *implies* $P \approx Q$.

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

(3) $\approx$ *is the largest weak bisimulation.*

(4) $\approx$ *abstracts from* τ*-loops.*

Properties of Weak Bisimilarity

Lemma 6.3 (Properties of $\approx$)

(1) $P \sim Q$ *implies* $P \approx Q$.

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

(3) $\approx$ *is the largest weak bisimulation.*

(4) $\approx$ *abstracts from* τ*-loops.*

Proof.

(1) straightforward (as $\xrightarrow{\alpha} \subseteq \xRightarrow{\alpha}$)

(2) similar to Lemma 4.7(1) for $\sim$

(3) similar to Lemma 4.7(2) for $\sim$

(4) see Examples 6.1 and 6.2

□

Milner's τ-Laws

Lemma 6.4 (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 6.4 (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) □

Outline of Lecture 6

Weak Bisimilarity vs. Trace Equivalence

Definition 6.5 (Observational trace language)

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

$$ObsTr(P) := \{\widehat{w} \in (A \cup \overline{A})^* \mid \text{there ex. } P' \in Prc : P \overset{w}{\longrightarrow} P'\}$$

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

Weak Bisimilarity vs. Trace Equivalence

Definition 6.5 (Observational trace language)

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

$$ObsTr(P) := \{\widehat{w} \in (A \cup \overline{A})^* \mid \text{there ex. } P' \in Prc : P \xrightarrow{w} P'\}$$

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

Definition 6.6 (Observational trace equivalence)

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

Weak Bisimilarity vs. Trace Equivalence

Definition 6.5 (Observational trace language)

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

$$ObsTr(P) := \{\widehat{w} \in \left(A \cup \overline{A}\right)^* \mid \text{there ex. } P' \in Prc : P \xrightarrow{w} P'\}$$

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

Definition 6.6 (Observational trace equivalence)

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

Theorem 6.7

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

Proof.

similar to Theorem 4.9 □

Outline of Lecture 6

Observational Deadlocks

Definition 6.8 (Observational deadlock)

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

$$P \xrightarrow{w} Q \text{ and there exists no } Q' \in Prc \text{ and } \lambda \in A \cup \overline{A} \text{ such that } Q \stackrel{\lambda}{\Longrightarrow} Q'.$$

Then Q is called an observational $\widehat{w}$-deadlock of P.

Observational Deadlocks

Definition 6.8 (Observational deadlock)

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

$$P \xrightarrow{w} Q \text{ and there exists no } Q' \in Prc \text{ and } \lambda \in A \cup \overline{A} \text{ such that } Q \stackrel{\lambda}{\Longrightarrow} Q'.$$

Then Q is called an observational $\widehat{w}$-deadlock of P.

Definition 6.9 (Observational deadlock sensitivity)

Relation $\equiv\; \subseteq Prc \times Prc$ is observationally deadlock sensitive whenever $P \equiv Q$ implies for every $v \in \left(A \cup \overline{A}\right)^*$:

$$P \text{ has an observational } v\text{-deadlock} \quad \text{iff} \quad Q \text{ has an observational } v\text{-deadlock}.$$

Observational Deadlocks

Definition 6.8 (Observational deadlock)

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

$$P \xrightarrow{w} Q \text{ and there exists no } Q' \in Prc \text{ and } \lambda \in A \cup \overline{A} \text{ such that } Q \overset{\lambda}{\Longrightarrow} Q'.$$

Then Q is called an observational $\widehat{w}$-deadlock of P.

Definition 6.9 (Observational deadlock sensitivity)

Relation $\equiv \subseteq Prc \times Prc$ is observationally deadlock sensitive whenever $P \equiv Q$ implies for every $v \in (A \cup \overline{A})^*$:

$$P \text{ has an observational } v\text{-deadlock} \quad \text{iff} \quad Q \text{ has an observational } v\text{-deadlock}.$$

Theorem 6.10

$\approx$ is observationally deadlock sensitive.

Proof.

similar to Theorem 4.13 for $\sim$ □

Outline of Lecture 6

Congruence

Lemma 6.11 (Partial CCS congruence property of $\approx$)

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

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

Congruence

Lemma 6.11 (Partial CCS congruence property of $\approx$)

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

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

Proof.

omitted □

Lemma 6.11 (Partial CCS congruence property of $\approx$)

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

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

Proof.

omitted □

What about choice?

- $\tau.a.\text{nil} \approx a.\text{nil}$ (cf. Example 5.11(1)) and $b.\text{nil} \approx b.\text{nil}$ (reflexivity).

Congruence

Lemma 6.11 (Partial CCS congruence property of $\approx$)

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

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

Proof.

omitted □

What about choice?

- $\tau.a.\mathrm{nil} \approx a.\mathrm{nil}$ (cf. Example 5.11(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?)

Congruence

Lemma 6.11 (Partial CCS congruence property of $\approx$)

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

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

Proof.

omitted □

What about choice?

- $\tau.a.\text{nil} \approx a.\text{nil}$ (cf. Example 5.11(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?)
- Thus, weak bisimilarity is not a CCS congruence, which motivates a slight adaptation of $\approx$.

Outline of Lecture 6

Observation Congruence

Definition 6.12 (Observation congruence) (Milner 1989)

$P, Q \in Prc$ are observationally congruent, denoted $P \approx^c Q$, if for every $\alpha \in Act$ (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 $P' \approx Q'$.

Observation Congruence

Definition 6.12 (Observation congruence) (Milner 1989)

$P, Q \in Prc$ are observationally congruent, denoted $P \approx^c Q$, if for every $\alpha \in Act$ (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.
- Moreover, this only applies to the first step; the successors just have to satisfy $P' \approx Q'$ (and not necessarily $P' \approx^c Q'$).

Observation Congruence

Definition 6.12 (Observation congruence) (Milner 1989)

$P, Q \in Prc$ are observationally congruent, denoted $P \approx^c Q$, if for every $\alpha \in Act$ (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.
- Moreover, 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 6.13

(1) Sequential and parallel two-place buffer:

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

$P_1 \approx^c Q_1$ since $P_1 \approx Q_1$ (cf. Example 5.11(3)) and neither P_1 nor Q_1 has initial τ-steps.

Examples

Example 6.13

(1) Sequential and parallel two-place buffer:

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

$P_1 \approx^c Q_1$ since $P_1 \approx Q_1$ (cf. Example 5.11(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.

Example 6.13

(1) Sequential and parallel two-place buffer:

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

$P_1 \approx^c Q_1$ since $P_1 \approx Q_1$ (cf. Example 5.11(3)) and neither P_1 nor Q_1 has initial τ-steps.

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

(3) $a.\tau.\mathrm{nil} \approx^c a.\mathrm{nil}$ (since $\tau.\mathrm{nil} \approx \mathrm{nil}$).

Properties of Observation Congruence

Lemma 6.14

For every $P, Q \in Prc$,

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

Properties of Observation Congruence

Lemma 6.14

For every $P, Q \in Prc$,

1. $P \sim Q$ *implies* $P \approx^c Q$, *and* $P \approx^c Q$ *implies* $P \approx Q$
2. $\approx^c$ *is an equivalence relation*
3. $\approx^c$ *is a CCS congruence*
4. $\approx^c$ *is observationally 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 6

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 $\stackrel{\alpha}{\Longrightarrow}$-move in the other process s of the current configuration under the same action α, yielding $s \stackrel{\alpha}{\Longrightarrow} s'$.

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 $\xRightarrow{\alpha}$-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.

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 $s \xRightarrow{\alpha} s'$.

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 6.15 (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 6.15 (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 □