Reactive Systems: Modelling, Specification and Verification

EWSCS'07–Lecture 4

- Properties of strong bisimilarity (reprise)
- Bisimulation games
- Weak bisimilarity and weak bisimulation games
- Properties of weak bisimilarity
- Example: a communication protocol and its modelling in CCS
- Concurrency workbench (CWB)

Strong Bisimilarity

Let $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ be an LTS.

Strong Bisimulation

A binary relation $R \subseteq Proc \times Proc$ is a strong bisimulation iff whenever $(s,t) \in R$ then for each $a \in Act$:

- if $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ for some t' such that $(s',t') \in R$
- if $t \xrightarrow{a} t'$ then $s \xrightarrow{a} s'$ for some s' such that $(s',t') \in R$.

Strong Bisimilarity

Two processes $p_1, p_2 \in Proc$ are strongly bisimilar ($p_1 \sim p_2$) if and only if there exists a strong bisimulation R such that $(p_1, p_2) \in R$.

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

Basic Properties of Strong Bisimilarity

Theorem

$\sim$ *is an equivalence relation (reflexive, symmetric and transitive)*

Theorem

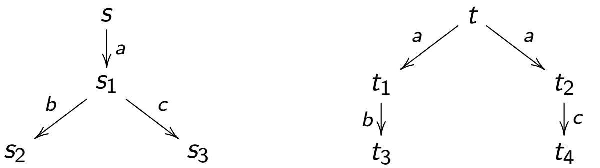
$\sim$ *is the largest strong bisimulation*

Theorem

$s \sim t$ *if and only if for each* $a \in Act$:

- *if* $s \xrightarrow{a} s'$ *then* $t \xrightarrow{a} t'$ *for some* t' *such that* $s' \sim t'$
- *if* $t \xrightarrow{a} t'$ *then* $s \xrightarrow{a} s'$ *for some* s' *such that* $s' \sim t'$.

How to Show Nonbisimilarity?



To prove that $s \not\sim t$:

- Enumerate all binary relations and show that none of them at the same time contains (s, t) and is a strong bisimulation. (Expensive: $2^{|Proc|^2}$ relations.)
- Make certain observations which enable us to disqualify many bisimulation candidates in one step.
- Use the game characterization of strong bisimilarity.

How to Show Nonbisimilarity?

$$\begin{tikzcd} & s \arrow[d, "a"] & \\ & s_1 \arrow[dl, "b"'] \arrow[dr, "c"] & \\ s_2 & & s_3 \end{tikzcd} \qquad \begin{tikzcd} & t \arrow[dl, "a"'] \arrow[dr, "a"] & \\ t_1 \arrow[d, "b"'] & & t_2 \arrow[d, "c"] \\ t_3 & & t_4 \end{tikzcd}$$

To prove that $s \not\sim t$:

- Enumerate all binary relations and show that none of them at the same time contains (s, t) and is a strong bisimulation. (Expensive: $2^{|Proc|^2}$ relations.)
- Make certain observations which enable us to disqualify many bisimulation candidates in one step.
- Use the game characterization of strong bisimilarity.

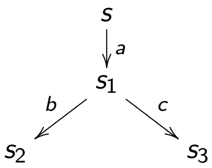
How to Show Nonbisimilarity?

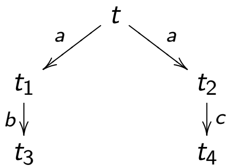
$$
\begin{tikzcd}
 & s \arrow[d, "a"] & \\
 & s_1 \arrow[dl, "b"'] \arrow[dr, "c"] & \\
s_2 & & s_3
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
 & t \arrow[dl, "a"'] \arrow[dr, "a"] & \\
t_1 \arrow[d, "b"'] & & t_2 \arrow[d, "c"] \\
t_3 & & t_4
\end{tikzcd}
$$

To prove that $s \not\sim t$:

- Enumerate all binary relations and show that none of them at the same time contains (s, t) and is a strong bisimulation. (Expensive: $2^{|Proc|^2}$ relations.)
- Make certain observations which enable us to disqualify many bisimulation candidates in one step.
- Use the game characterization of strong bisimilarity.

How to Show Nonbisimilarity?





To prove that $s \not\sim t$:

- Enumerate all binary relations and show that none of them at the same time contains (s, t) and is a strong bisimulation. (Expensive: $2^{|Proc|^2}$ relations.)
- Make certain observations which enable us to disqualify many bisimulation candidates in one step.
- Use the game characterization of strong bisimilarity.

Strong Bisimulation Game

Let $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ be an LTS and $s, t \in Proc$.

We define a two-player game of an 'attacker' and a 'defender' starting from s and t.

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

Intuition

The defender wants to show that s and t are strongly bisimilar while the attacker aims at proving the opposite.

Strong Bisimulation Game

Let $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ be an LTS and $s, t \in Proc$.

We define a two-player game of an 'attacker' and a 'defender' starting from s and t.

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

Intuition

The defender wants to show that s and t are strongly bisimilar while the attacker aims at proving the opposite.

Rules of the Bisimulation Games

Game Rules

In each round the players change the current configuration as follows:

1. the attacker chooses one of the processes in the current configuration and makes an $\xrightarrow{a}$-move for some $a \in Act$, and
2. the defender must respond by making an $\xrightarrow{a}$-move in the other process under the same action a.

The newly reached pair of processes becomes the current configuration. The game then continues by another round.

Result of the Game

- If one player cannot move, the other player wins.
- If the game is infinite, the defender wins.

Rules of the Bisimulation Games

Game Rules

In each round the players change the current configuration as follows:

1. the attacker chooses one of the processes in the current configuration and makes an $\xrightarrow{a}$-move for some $a \in Act$, and
2. the defender must respond by making an $\xrightarrow{a}$-move in the other process under the same action a.

The newly reached pair of processes becomes the current configuration. The game then continues by another round.

Result of the Game

- If one player cannot move, the other player wins.
- If the game is infinite, the defender wins.

Game Characterization of Strong Bisimilarity

Theorem

- States s and t are strongly bisimilar if and only if the defender has a universal winning strategy starting from the configuration (s, t).
- States s and t are not strongly bisimilar if and only if the attacker has a universal winning strategy starting from the configuration (s, t).

Remark

The bisimulation game can be used to prove both bisimilarity and nonbisimilarity of two processes. It very often provides elegant arguments for the negative case.

Game Characterization of Strong Bisimilarity

Theorem

- States s and t are strongly bisimilar if and only if the defender has a universal winning strategy starting from the configuration (s, t).
- States s and t are not strongly bisimilar if and only if the attacker has a universal winning strategy starting from the configuration (s, t).

Remark

The bisimulation game can be used to prove both bisimilarity and nonbisimilarity of two processes. It very often provides elegant arguments for the negative case.

Strong Bisimilarity is a Congruence for CCS Operations

Theorem

Let P and Q be CCS processes such that $P \sim Q$. Then

- $\alpha.P \sim \alpha.Q$ *for each action* $\alpha \in Act$
- $P + R \sim Q + R$ *and* $R + P \sim R + Q$ *for each CCS process* R
- $P \mid R \sim Q \mid R$ *and* $R \mid P \sim R \mid Q$ *for each CCS process* R
- $P[f] \sim Q[f]$ *for each relabelling function* f
- $P \setminus L \sim Q \setminus L$ *for each set of labels* L.

Other Properties of Strong Bisimilarity

The Following Properties Hold for all CCS Processes P, Q, R

- $P + Q \sim Q + P$
- $P \mid Q \sim Q \mid P$
- $P + Nil \sim P$
- $P \mid Nil \sim P$
- $(P + Q) + R \sim P + (Q + R)$
- $(P \mid Q) \mid R \sim P \mid (Q \mid R)$

Example – Buffer

Buffer of Capacity 1

$$B_0^1 \stackrel{\text{def}}{=} in.B_1^1$$
$$B_1^1 \stackrel{\text{def}}{=} \overline{out}.B_0^1$$

Buffer of Capacity n

$$B_0^n \stackrel{\text{def}}{=} in.B_1^n$$
$$B_i^n \stackrel{\text{def}}{=} in.B_{i+1}^n + \overline{out}.B_{i-1}^n \quad \text{for } 0 < i < n$$
$$B_n^n \stackrel{\text{def}}{=} \overline{out}.B_{n-1}^n$$

Example: $B_0^2 \sim B_0^1 | B_0^1$

Example – Buffer

Buffer of Capacity 1

$$B^1_0 \stackrel{\text{def}}{=} in.B^1_1$$
$$B^1_1 \stackrel{\text{def}}{=} \overline{out}.B^1_0$$

Buffer of Capacity n

$$B^n_0 \stackrel{\text{def}}{=} in.B^n_1$$
$$B^n_i \stackrel{\text{def}}{=} in.B^n_{i+1} + \overline{out}.B^n_{i-1} \quad \text{for } 0 < i < n$$
$$B^n_n \stackrel{\text{def}}{=} \overline{out}.B^n_{n-1}$$

Example: $B^2_0 \sim B^1_0|B^1_0$

Example – Buffer

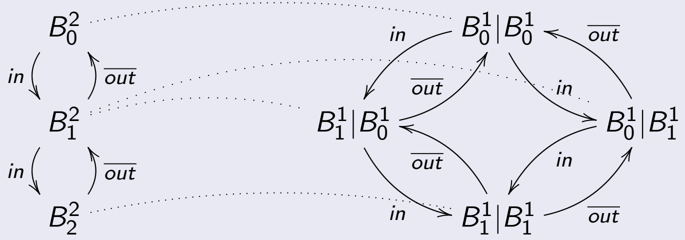
Buffer of Capacity 1

$$B_0^1 \stackrel{\text{def}}{=} in.B_1^1$$
$$B_1^1 \stackrel{\text{def}}{=} \overline{out}.B_0^1$$

Buffer of Capacity n

$$B_0^n \stackrel{\text{def}}{=} in.B_1^n$$
$$B_i^n \stackrel{\text{def}}{=} in.B_{i+1}^n + \overline{out}.B_{i-1}^n \quad \text{for } 0 < i < n$$
$$B_n^n \stackrel{\text{def}}{=} \overline{out}.B_{n-1}^n$$

Example: $B_0^2 \sim B_0^1 | B_0^1$



Example – Buffer

Theorem

For all natural numbers n: $B_0^n \sim \underbrace{B_0^1|B_0^1|\cdots|B_0^1}_{n\ times}$

Proof.

Construct the following binary relation where $i_1, i_2, \ldots, i_n \in \{0,1\}$.

$$R = \{\big(B_i^n,\ B_{i_1}^1|B_{i_2}^1|\cdots|B_{i_n}^1\big) \mid \sum_{j=1}^{n} i_j = i\}$$

- $\big(B_0^n,\ B_0^1|B_0^1|\cdots|B_0^1\big) \in R$
- R is strong bisimulation

$\square$

Example – Buffer

Theorem

For all natural numbers n: $B_0^n \sim \underbrace{B_0^1 | B_0^1 | \cdots | B_0^1}_{n\ times}$

Proof.

Construct the following binary relation where $i_1, i_2, \ldots, i_n \in \{0, 1\}$.

$$R = \{ \big(B_i^n ,\ B_{i_1}^1 | B_{i_2}^1 | \cdots | B_{i_n}^1 \big) \mid \sum_{j=1}^{n} i_j = i \}$$

- $\big(B_0^n ,\ B_0^1 | B_0^1 | \cdots | B_0^1 \big) \in R$
- R is strong bisimulation

$\square$

Strong Bisimilarity – Summary

Properties of $\sim$

- an equivalence relation
- the largest strong bisimulation
- a congruence
- enough to prove some natural rules like
  - $P|Q \sim Q|P$
  - $P|Nil \sim P$
  - $(P|Q)|R \sim Q|(P|R)$
  - $\cdots$

Question

Should we look any further???

Strong Bisimilarity – Summary

Properties of $\sim$

- an equivalence relation
- the largest strong bisimulation
- a congruence
- enough to prove some natural rules like
  - $P|Q \sim Q|P$
  - $P|Nil \sim P$
  - $(P|Q)|R \sim Q|(P|R)$
  - $\cdots$

Question

Should we look any further???

Problems with Internal Actions

Question

Does $a.\tau.Nil \sim a.Nil$ hold? NO!

Problem

Strong bisimilarity does not abstract away from τ actions.

Example: SmUni $\not\sim$ Spec

$$
\begin{tikzcd}
\mathrm{SmUni} \arrow[d, "\overline{pub}"] & \not\sim & \mathrm{Spec} \arrow[loop below, "\overline{pub}"] \\
(CM \mid CS_1) \smallsetminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM_1 \mid CS_2) \smallsetminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM \mid CS) \smallsetminus \{coin, coffee\} \arrow[uu, bend right, "\overline{pub}"'] & &
\end{tikzcd}
$$

Problems with Internal Actions

Question

Does $a.\tau.Nil \sim a.Nil$ hold? NO!

Problem

Strong bisimilarity does not abstract away from τ actions.

Example: SmUni $\not\sim$ Spec

$$\begin{tikzcd}
\mathrm{SmUni} \arrow[d, "\overline{pub}"] & \not\sim & \mathrm{Spec} \arrow[loop below, "\overline{pub}"] \\
(CM \mid CS_1) \setminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM_1 \mid CS_2) \setminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM \mid CS) \setminus \{coin, coffee\} \arrow[uu, bend right=60, "\overline{pub}"'] & &
\end{tikzcd}$$

Problems with Internal Actions

Question

Does $a.\tau.Nil \sim a.Nil$ hold? NO!

Problem

Strong bisimilarity does not abstract away from τ actions.

Example: SmUni $\not\sim$ Spec

$$\begin{tikzcd}
\text{SmUni} \arrow[d, "\overline{pub}"] & \not\sim & \text{Spec} \arrow[loop below, "\overline{pub}"] \\
(CM \mid CS_1) \setminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM_1 \mid CS_2) \setminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM \mid CS) \setminus \{coin, coffee\} \arrow[uu, bend right, "\overline{pub}"'] & &
\end{tikzcd}$$

Problems with Internal Actions

Question

Does $a.\tau.Nil \sim a.Nil$ hold? NO!

Problem

Strong bisimilarity does not abstract away from τ actions.

Example: SmUni $\not\sim$ Spec

$$
\begin{tikzcd}
\mathrm{SmUni} \arrow[d, "\overline{pub}"] & \not\sim & \mathrm{Spec} \arrow[loop below, "\overline{pub}"] \\
(CM \mid CS_1) \setminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM_1 \mid CS_2) \setminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM \mid CS) \setminus \{coin, coffee\} \arrow[uu, bend right=60, "\overline{pub}"'] & &
\end{tikzcd}
$$

Weak Transition Relation

Let $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ be an LTS such that $\tau \in Act$.

Definition of Weak Transition Relation

$$\stackrel{a}{\Longrightarrow} = \begin{cases} (\xrightarrow{\tau})^* \circ \xrightarrow{a} \circ (\xrightarrow{\tau})^* & \text{if } a \neq \tau \\ (\xrightarrow{\tau})^* & \text{if } a = \tau \end{cases}$$

What does $s \stackrel{a}{\Longrightarrow} t$ informally mean?

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

Weak Transition Relation

Let $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ be an LTS such that $\tau \in Act$.

Definition of Weak Transition Relation

$$\overset{a}{\Longrightarrow} = \begin{cases} (\xrightarrow{\tau})^* \circ \xrightarrow{a} \circ (\xrightarrow{\tau})^* & \text{if } a \neq \tau \\ (\xrightarrow{\tau})^* & \text{if } a = \tau \end{cases}$$

What does $s \overset{a}{\Longrightarrow} t$ informally mean?

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

Weak Bisimilarity

Let $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ be an LTS such that $\tau \in Act$.

Weak Bisimulation

A binary relation $R \subseteq Proc \times Proc$ is a weak bisimulation iff whenever $(s, t) \in R$ then for each $a \in Act$ (including τ):

- if $s \xrightarrow{a} s'$ then $t \stackrel{a}{\Longrightarrow} t'$ for some t' such that $(s', t') \in R$
- if $t \xrightarrow{a} t'$ then $s \stackrel{a}{\Longrightarrow} s'$ for some s' such that $(s', t') \in R$.

Weak Bisimilarity

Two processes $p_1, p_2 \in Proc$ are weakly bisimilar ($p_1 \approx p_2$) if and only if there exists a weak bisimulation R such that $(p_1, p_2) \in R$.

$$\approx \; = \; \bigcup\{R \mid R \text{ is a weak bisimulation}\}$$

Weak Bisimilarity

Let $(Proc, Act, \{\xrightarrow{a} \mid a \in Act\})$ be an LTS such that $\tau \in Act$.

Weak Bisimulation

A binary relation $R \subseteq Proc \times Proc$ is a weak bisimulation iff whenever $(s, t) \in R$ then for each $a \in Act$ (including τ):

- if $s \xrightarrow{a} s'$ then $t \stackrel{a}{\Longrightarrow} t'$ for some t' such that $(s', t') \in R$
- if $t \xrightarrow{a} t'$ then $s \stackrel{a}{\Longrightarrow} s'$ for some s' such that $(s', t') \in R$.

Weak Bisimilarity

Two processes $p_1, p_2 \in Proc$ are weakly bisimilar ($p_1 \approx p_2$) if and only if there exists a weak bisimulation R such that $(p_1, p_2) \in R$.

$$\approx \; = \; \cup\{R \mid R \text{ is a weak bisimulation}\}$$

Weak Bisimulation Game

Definition

All the same except that

- defender can now answer using $\stackrel{a}{\Longrightarrow}$ moves.

The attacker is still using only $\stackrel{a}{\longrightarrow}$ moves.

Theorem

- States s and t are weakly bisimilar if and only if the defender has a universal winning strategy starting from the configuration (s, t).
- States s and t are not weakly bisimilar if and only if the attacker has a universal winning strategy starting from the configuration (s, t).

Weak Bisimulation Game

Definition

All the same except that

- defender can now answer using $\stackrel{a}{\Longrightarrow}$ moves.

The attacker is still using only $\stackrel{a}{\longrightarrow}$ moves.

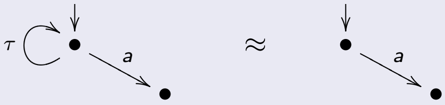
Theorem

- States s and t are weakly bisimilar if and only if the defender has a universal winning strategy starting from the configuration (s, t).
- States s and t are not weakly bisimilar if and only if the attacker has a universal winning strategy starting from the configuration (s, t).

Weak Bisimilarity – Properties

Properties of $\approx$

- an equivalence relation
- the largest weak bisimulation
- validates lots of natural laws, e.g.
  - $a.\tau.P \approx a.P$
  - $P + \tau.P \approx \tau.P$
  - $a.(P + \tau.Q) \approx a.(P + \tau.Q) + a.Q$
  - $P + Q \approx Q + P \qquad P|Q \approx Q|P \qquad P + Nil \approx P \qquad \ldots$
- strong bisimilarity is included in weak bisimilarity ($\sim \subseteq \approx$)
- abstracts from τ loops



Case Study: Communication Protocol

$$
\begin{aligned}
\mathsf{Send} &\stackrel{\mathrm{def}}{=} \mathsf{acc}.\mathsf{Sending} \\
\mathsf{Sending} &\stackrel{\mathrm{def}}{=} \overline{\mathsf{send}}.\mathsf{Wait} \\
\mathsf{Wait} &\stackrel{\mathrm{def}}{=} \mathsf{ack}.\mathsf{Send} + \mathsf{error}.\mathsf{Sending}
\end{aligned}
\qquad
\begin{aligned}
\mathsf{Rec} &\stackrel{\mathrm{def}}{=} \mathsf{trans}.\mathsf{Del} \\
\mathsf{Del} &\stackrel{\mathrm{def}}{=} \overline{\mathsf{del}}.\mathsf{Ack} \\
\mathsf{Ack} &\stackrel{\mathrm{def}}{=} \overline{\mathsf{ack}}.\mathsf{Rec}
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{Med} &\stackrel{\mathrm{def}}{=} \mathsf{send}.\mathsf{Med}' \\
\mathsf{Med}' &\stackrel{\mathrm{def}}{=} \tau.\mathsf{Err} + \overline{\mathsf{trans}}.\mathsf{Med} \\
\mathsf{Err} &\stackrel{\mathrm{def}}{=} \overline{\mathsf{error}}.\mathsf{Med}
\end{aligned}
$$

Case Study: Communication Protocol

$$\begin{aligned}
\mathsf{Send} &\stackrel{\text{def}}{=} \mathsf{acc}.\mathsf{Sending} \\
\mathsf{Sending} &\stackrel{\text{def}}{=} \overline{\mathsf{send}}.\mathsf{Wait} \\
\mathsf{Wait} &\stackrel{\text{def}}{=} \mathsf{ack}.\mathsf{Send} + \mathsf{error}.\mathsf{Sending}
\end{aligned}
\qquad
\begin{aligned}
\mathsf{Rec} &\stackrel{\text{def}}{=} \mathsf{trans}.\mathsf{Del} \\
\mathsf{Del} &\stackrel{\text{def}}{=} \overline{\mathsf{del}}.\mathsf{Ack} \\
\mathsf{Ack} &\stackrel{\text{def}}{=} \overline{\mathsf{ack}}.\mathsf{Rec}
\end{aligned}$$

$$\begin{aligned}
\mathsf{Med} &\stackrel{\text{def}}{=} \mathsf{send}.\mathsf{Med}' \\
\mathsf{Med}' &\stackrel{\text{def}}{=} \tau.\mathsf{Err} + \overline{\mathsf{trans}}.\mathsf{Med} \\
\mathsf{Err} &\stackrel{\text{def}}{=} \overline{\mathsf{error}}.\mathsf{Med}
\end{aligned}$$

Verification Question

$$\mathsf{Impl} \stackrel{\mathrm{def}}{=} (\mathsf{Send} \mid \mathsf{Med} \mid \mathsf{Rec}) \smallsetminus \{\mathsf{send}, \mathsf{trans}, \mathsf{ack}, \mathsf{error}\}$$

$$\mathsf{Spec} \stackrel{\mathrm{def}}{=} \mathsf{acc}.\overline{\mathsf{del}}.\mathsf{Spec}$$

Question

$$\mathsf{Impl} \stackrel{?}{\approx} \mathsf{Spec}$$

1. Draw the LTS of Impl and Spec and prove (by hand) the equivalence.
2. Use Concurrency WorkBench (CWB).

Verification Question

$$\mathsf{Impl} \stackrel{\mathrm{def}}{=} (\mathsf{Send} \mid \mathsf{Med} \mid \mathsf{Rec}) \smallsetminus \{\mathsf{send}, \mathsf{trans}, \mathsf{ack}, \mathsf{error}\}$$

$$\mathsf{Spec} \stackrel{\mathrm{def}}{=} \mathsf{acc}.\overline{\mathsf{del}}.\mathsf{Spec}$$

Question

$$\mathsf{Impl} \stackrel{?}{\approx} \mathsf{Spec}$$

1. Draw the LTS of Impl and Spec and prove (by hand) the equivalence.
2. Use Concurrency WorkBench (CWB).

Verification Question

$$\mathsf{Impl} \stackrel{\text{def}}{=} (\mathsf{Send} \mid \mathsf{Med} \mid \mathsf{Rec}) \smallsetminus \{\mathsf{send}, \mathsf{trans}, \mathsf{ack}, \mathsf{error}\}$$

$$\mathsf{Spec} \stackrel{\text{def}}{=} \mathsf{acc}.\overline{\mathsf{del}}.\mathsf{Spec}$$

Question

$$\mathsf{Impl} \stackrel{?}{\approx} \mathsf{Spec}$$

1. Draw the LTS of Impl and Spec and prove (by hand) the equivalence.
2. Use Concurrency WorkBench (CWB).

Verification Question

$$\mathrm{Impl} \stackrel{\mathrm{def}}{=} (\mathrm{Send} \mid \mathrm{Med} \mid \mathrm{Rec}) \smallsetminus \{\mathrm{send}, \mathrm{trans}, \mathrm{ack}, \mathrm{error}\}$$

$$\mathrm{Spec} \stackrel{\mathrm{def}}{=} \mathrm{acc}.\overline{\mathrm{del}}.\mathrm{Spec}$$

Question

$$\mathrm{Impl} \stackrel{?}{\approx} \mathrm{Spec}$$

1. Draw the LTS of Impl and Spec and prove (by hand) the equivalence.
2. Use Concurrency WorkBench (CWB).

Verification Question

$$\mathsf{Impl} \stackrel{\text{def}}{=} (\mathsf{Send} \mid \mathsf{Med} \mid \mathsf{Rec}) \smallsetminus \{\mathsf{send}, \mathsf{trans}, \mathsf{ack}, \mathsf{error}\}$$

$$\mathsf{Spec} \stackrel{\text{def}}{=} \mathsf{acc}.\overline{\mathsf{del}}.\mathsf{Spec}$$

Question

$$\mathsf{Impl} \stackrel{?}{\approx} \mathsf{Spec}$$

1. Draw the LTS of Impl and Spec and prove (by hand) the equivalence.
2. Use Concurrency WorkBench (CWB).

CCS Expressions in CWB

CCS Definitions

$\mathsf{Med} \stackrel{\text{def}}{=} \mathsf{send}.\mathsf{Med}'$
$\mathsf{Med}' \stackrel{\text{def}}{=} \tau.\mathsf{Err} + \overline{\mathsf{trans}}.\mathsf{Med}$
$\mathsf{Err} \stackrel{\text{def}}{=} \overline{\mathsf{error}}.\mathsf{Med}$
$\vdots$
$\mathsf{Impl} \stackrel{\text{def}}{=} (\mathsf{Send} \mid \mathsf{Med} \mid \mathsf{Rec}) \smallsetminus \{\mathsf{send}, \mathsf{trans}, \mathsf{ack}, \mathsf{error}\}$

$\mathsf{Spec} \stackrel{\text{def}}{=} \mathsf{acc}.\overline{\mathsf{del}}.\mathsf{Spec}$

CWB Program (protocol.cwb)

```
agent Med = send.Med';
agent Med' = (tau.Err + 'trans.Med);
agent Err = 'error.Med;
⋮
set L = {send, trans, ack, error};
agent Impl = (Send | Med | Rec) \ L;

agent Spec = acc.'del.Spec;
```

CWB Session

```
[luca@vel5638 CWB]$ ./xccscwb.x86-linux

> help;

> input "protocol.cwb";

> vs(5,Impl);

> sim(Spec);

> eq(Spec,Impl);              ** weak bisimilarity **

> strongeq(Spec,Impl);        ** strong bisimilarity **
```

Is Weak Bisimilarity a Congruence for CCS?

Theorem

Let P and Q be CCS processes such that $P \approx Q$. Then

- $\alpha.P \approx \alpha.Q$ *for each action* $\alpha \in Act$
- $P \mid R \approx Q \mid R$ *and* $R \mid P \approx R \mid Q$ *for each CCS process* R
- $P[f] \approx Q[f]$ *for each relabelling function* f
- $P \setminus L \approx Q \setminus L$ *for each set of labels* L.

What about choice?

$\tau.a.Nil \approx a.Nil$ but $\tau.a.Nil + b.Nil \not\approx a.Nil + b.Nil$

Conclusion

Weak bisimilarity is not a congruence for CCS.

Is Weak Bisimilarity a Congruence for CCS?

Theorem

Let P and Q be CCS processes such that $P \approx Q$. Then

- $\alpha.P \approx \alpha.Q$ *for each action* $\alpha \in Act$
- $P \mid R \approx Q \mid R$ *and* $R \mid P \approx R \mid Q$ *for each CCS process* R
- $P[f] \approx Q[f]$ *for each relabelling function* f
- $P \setminus L \approx Q \setminus L$ *for each set of labels* L.

What about choice?

$\tau.a.Nil \approx a.Nil$ but $\tau.a.Nil + b.Nil \not\approx a.Nil + b.Nil$

Conclusion

Weak bisimilarity is not a congruence for CCS.

Is Weak Bisimilarity a Congruence for CCS?

Theorem

Let P and Q be CCS processes such that $P \approx Q$. Then

- $\alpha.P \approx \alpha.Q$ *for each action* $\alpha \in Act$
- $P \mid R \approx Q \mid R$ *and* $R \mid P \approx R \mid Q$ *for each CCS process* R
- $P[f] \approx Q[f]$ *for each relabelling function* f
- $P \setminus L \approx Q \setminus L$ *for each set of labels* L.

What about choice?

$\tau.a.Nil \approx a.Nil$ but $\tau.a.Nil + b.Nil \not\approx a.Nil + b.Nil$

Conclusion

Weak bisimilarity is not a congruence for CCS.