Semantics and Verification 2006

Lecture 4

- properties of strong bisimilarity
- weak bisimilarity and weak bisimulation games
- properties of weak bisimilarity
- example: a communication protocol and its modelling in CCS
- concurrency workbench (CWB)

Strong Bisimilarity – Properties

Strong Bisimilarity is a Congruence for All CCS Operators

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.

Following Properties Hold for any CCS Processes P, Q and 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

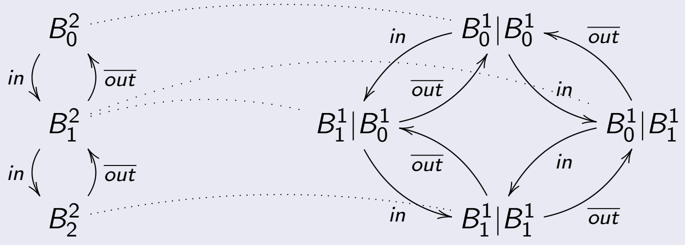
Buffer of Capacity 1

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

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

Theorem

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

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???

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

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

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

The attacker is still using only $\xrightarrow{a}$ 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

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.

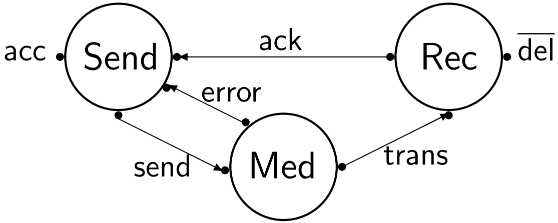
What about choice?

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

Conclusion

Weak bisimilarity is not a congruence for CCS.

Case Study: Communication Protocol



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

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

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).

CCS Expressions in CWB

CCS Definitions

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

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

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

```
borg$ /pack/FS/CWB/cwb

> help;

> input "protocol.cwb";

> vs(5,Impl);

> sim(Spec);

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

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