Semantics and Verification

Lecture 4

23 February 2010

Overview

Last lecture:

- Behavioral equivalences: strong bisimilarity

This lecture:

- Weak bisimilarity
- Introduction to Concurrency Workbench

Next lecture:

- Hennessy-Milner logic

Behavioral Equivalences: Weak Bisimilarity

Behavioral Equivalences (R)

Main Idea

Two processes are behaviorally equivalent if and only if an external observer cannot tell them apart.

– black-box experiments

Strong Bisimilarity (R)

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 $(p_1, p_2) \in R$.

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

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 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

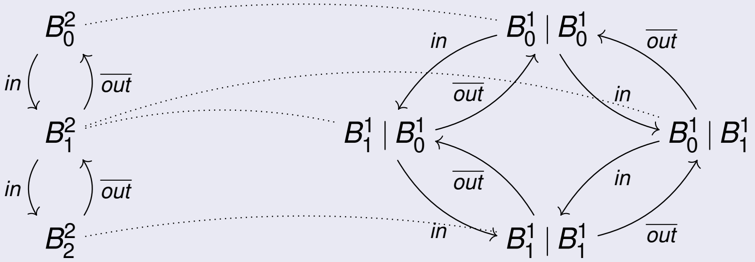
Buffer of Capacity 1

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

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: $B_0^2 \sim B_0^1 \mid B_0^1$



Example (contd.)

Theorem

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

Proof.

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

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

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

$\square$

Summary

Properties of strong bisimilarity

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

Question

Should we look any further???

Problems with Internal Actions

Question

Does $a.\tau.Nil \sim a.Nil$ hold?

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

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

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 $(p_1, p_2) \in R$.

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

Example

Example: SmUni ≈ Spec

$$
\begin{tikzcd}
\text{SmUni} \arrow[d, "\overline{pub}"] & \approx & \text{Spec} \arrow[loop below, "\overline{pub}"] \\
(CM \mid CS_1) \setminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM_1 \mid CS_2) \setminus \{coin, coffee\} \arrow[d, "\tau"] & & \\
(CM \mid CS) \setminus \{coin, coffee\} \arrow[uu, bend right=60, swap, "\overline{pub}"] & &
\end{tikzcd}
$$

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).

Properties

Properties of weak bisimilarity

- an equivalence relation
- the largest weak bisimulation
- validates lots of natural laws, e.g.
  - $a.\tau.P \approx a.P$
  - $P + \tau.P \approx \tau.P$
  - $a.(P + \tau.Q) \approx a.(P + \tau.Q) + a.Q$
  - $P + Q \approx Q + P \qquad P \mid Q \approx Q \mid P \qquad P + Nil \approx P \qquad \ldots$
- 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.*

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: Simple Mutual Exclusion Algorithm

Two concurrent processes, P1 and P2, communicate via a shared variable k to avoid being at the same time in the critical section:

P1:

```
while true do
  if k=1 then
    enter critical section
    ...
    exit critical section
    k:=2
  endif
endfor
```

P2:

```
while true do
  if k=2 then
    enter critical section
    ...
    exit critical section
    k:=1
  endif
endfor
```

CCS Model of the Algorithm

Boolean variable k

– can be read and written with value 1 or 2:

$$K1 \stackrel{\text{def}}{=} \overline{kr1}.K1 + kw1.K1 + kw2.K2$$
$$K2 \stackrel{\text{def}}{=} \overline{kr2}.K2 + kw1.K1 + kw2.K2$$

Process P1

$$P1 \stackrel{\text{def}}{=} kr2.P1 + kr1.P12$$
$$P12 \stackrel{\text{def}}{=} enter1.exit1.\overline{kw2}.P1$$

Process P2

$$P2 \stackrel{\text{def}}{=} kr1.P2 + kr2.P22$$
$$P22 \stackrel{\text{def}}{=} enter2.exit2.\overline{kw1}.P2$$

Whole algorithm

$$Impl \stackrel{\text{def}}{=} (P1|P2|K1) \smallsetminus \{kr1, kr2, kw1, kw2\}$$

Verification Question

$$Spec \stackrel{\text{def}}{=} enter1.exit1.Spec + enter2.exit2.Spec$$

Question

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

Will use Concurrency Workbench for model checking.

(But could also have done it by hand.)

CCS in CWB

CCS | CWB

$$K1 \stackrel{\text{def}}{=} \overline{kr1}.K1 + kw1.K1 + kw2.K2$$

```
agent K1 = 'kr1.K1 + kw1.K1 + kw2.K2;
```

$$K2 \stackrel{\text{def}}{=} \overline{kr2}.K2 + kw1.K1 + kw2.K2$$

```
agent K2 = 'kr2.K2 + kw1.K1 + kw2.K2;
```

$$Impl \stackrel{\text{def}}{=} (P1|P2|K1) \setminus \{kr1, kr2, kw1, kw2\}$$

```
set L = {kr1, kr2, kw1, kw2};
agent Impl = (P1 | P2 | K1 )  L;
```

(Rest see transcript.)