

Semantics and Verification 2008

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
- $P | R \sim Q | R$ and $R | P \sim R | Q$ 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

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

Example – Buffer

Buffer of Capacity 1

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

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

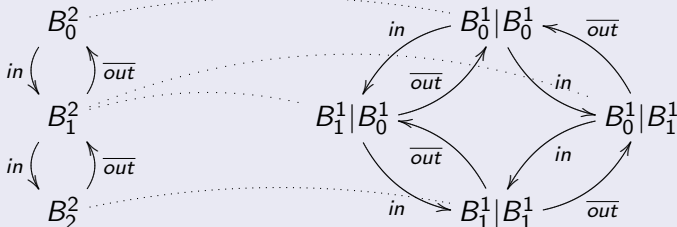
Buffer of Capacity n

$$B_0^n \stackrel{\text{def}}{=} in.B_1^n$$

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

$$B_n^n \stackrel{\text{def}}{=} \overline{out}.B_{n-1}^n$$

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



Example – Buffer

Theorem

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

Proof.

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

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

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



Strong Bisimilarity – Summary

Properties of \sim

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

Question

Should we look any further???

Problems with Internal Actions

Question

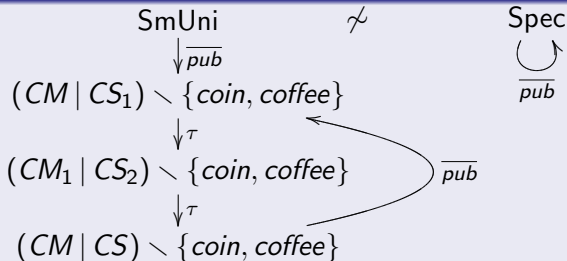
Does $a.\tau.Nil \sim a.Nil$ hold?

NO!

Problem

Strong bisimilarity does not abstract away from τ actions.

Example: $SmUni \not\sim Spec$



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

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

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

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

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 \xRightarrow{a} 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.
 - $a.\tau.P \approx a.P$
 - $P + \tau.P \approx \tau.P$
 - $a.(P + \tau.Q) \approx a.(P + \tau.Q) + a.Q$
 - $P + Q \approx Q + P$ $P|Q \approx Q|P$ $P + Nil \approx P$...
- 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 $\alpha \in \text{Act}$
- $P \mid R \approx Q \mid R$ 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 .

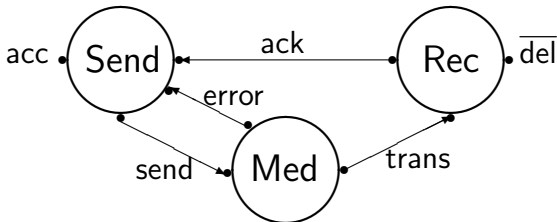
What about choice?

$\tau.a.Nil \approx a.Nil$ 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



Send	$\stackrel{\text{def}}{=}$	acc.Sending	Rec	$\stackrel{\text{def}}{=}$	trans.Del
Sending	$\stackrel{\text{def}}{=}$	$\overline{\text{send}}$.Wait	Del	$\stackrel{\text{def}}{=}$	$\overline{\text{del}}$.Ack
Wait	$\stackrel{\text{def}}{=}$	ack.Send + error.Sending	Ack	$\stackrel{\text{def}}{=}$	$\overline{\text{ack}}$.Rec
		Med	$\stackrel{\text{def}}{=}$	send.Med'	
		Med'	$\stackrel{\text{def}}{=}$	τ .Err + $\overline{\text{trans}}$.Med	
		Err	$\stackrel{\text{def}}{=}$	$\overline{\text{error}}$.Med	

Verification Question

$$\text{Impl} \stackrel{\text{def}}{=} (\text{Send} \mid \text{Med} \mid \text{Rec}) \setminus \{\text{send}, \text{trans}, \text{ack}, \text{error}\}$$
$$\text{Spec} \stackrel{\text{def}}{=} \text{acc}.\overline{\text{del}}.\text{Spec}$$

Question

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

- 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

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

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

```
fire1$ /pack/FS/CWB/cwb
```

```
> help;
```

```
> input "protocol.cwb";
```

```
> vs(5, Impl);
```

```
> sim(Spec);
```

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

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