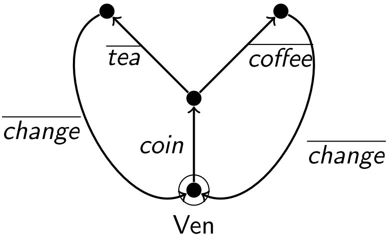
Topics in Concurrency

Lecture 4

Jonathan Hayman

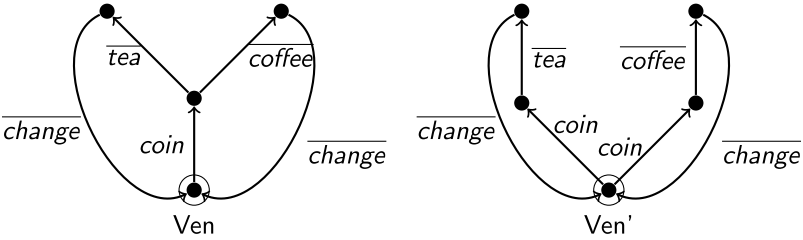
20 February 2015

Two vending machine implementations



$$User \stackrel{\mathrm{def}}{=} \overline{coin}.coffee.change.\overline{work}$$

Two vending machine implementations



$$User \stackrel{\text{def}}{=} \overline{coin}.coffee.change.\overline{work}$$

Specification and correctness:

- Assertions and logic (e.g. $(User \parallel Ven) \setminus \{coin, change, coffee, tea\}$ always outputs *work*)
- Equivalence

Language equivalences

- A trace of a process p is a (possibly infinite) sequence of actions

$$(a_1, a_2, \ldots, a_i, a_{i+1}, \ldots)$$

  such that

$$p \xrightarrow{a_1} p_1 \xrightarrow{a_2} \ldots p_{i-1} \xrightarrow{a_i} p_i \xrightarrow{a_{i+1}} \ldots$$

- Two processes are trace equivalent iff they have the same sets of traces

Language equivalences

- A trace of a process p is a (possibly infinite) sequence of actions

$$(a_1, a_2, \ldots, a_i, a_{i+1}, \ldots)$$

  such that

$$p \xrightarrow{a_1} p_1 \xrightarrow{a_2} \ldots p_{i-1} \xrightarrow{a_i} p_i \xrightarrow{a_{i+1}} \ldots$$

- Two processes are trace equivalent iff they have the same sets of traces

- Are Ven and Ven' trace equivalent?
- Are $(User \parallel Ven) \setminus \{coin, change, coffee, tea\}$ and $(User \parallel Ven') \setminus \{coin, change, coffee, tea\}$ trace equivalent?

Completed trace equivalence

- A trace is maximal if it cannot be extended (it is either infinite or ends in a state from which there is no transition)
- Processes are completed trace equivalent iff they have the same sets of *maximal* traces.

Completed trace equivalence

- A trace is maximal if it cannot be extended (it is either infinite or ends in a state from which there is no transition)
- Processes are completed trace equivalent iff they have the same sets of *maximal* traces.

- Are Ven and Ven' completed trace equivalent?
- Are $(\mathit{User} \parallel \mathit{Ven}) \setminus \{\mathit{coin}, \mathit{change}, \mathit{coffee}, \mathit{tea}\}$ and $(\mathit{User} \parallel \mathit{Ven}') \setminus \{\mathit{coin}, \mathit{change}, \mathit{coffee}, \mathit{tea}\}$ completed trace equivalent?

Completed trace equivalence

- A trace is maximal if it cannot be extended (it is either infinite or ends in a state from which there is no transition)
- Processes are completed trace equivalent iff they have the same sets of *maximal* traces.

- Are Ven and Ven' completed trace equivalent?
- Are $(User \parallel Ven) \setminus \{coin, change, coffee, tea\}$ and $(User \parallel Ven') \setminus \{coin, change, coffee, tea\}$ completed trace equivalent?

A more subtle form of equivalence is needed to reason compositionally about processes

Bisimulation — a process equivalence

To

- support equational reasoning
- simplify verification

Strong bisimulation

A (strong) bisimulation is a relation R between states for which
If $p\ R\ q$ then:

1. $\forall \alpha, p'. \quad p \xrightarrow{\alpha} p' \quad \Longrightarrow$
   $\exists q'. \quad q \xrightarrow{\alpha} q' \quad \& \quad p'\ R\ q'$
2. $\forall \alpha, q'. \quad q \xrightarrow{\alpha} q' \quad \Longrightarrow$
   $\exists p'. \quad p \xrightarrow{\alpha} p' \quad \& \quad p'\ R\ q'$

(Strong) bisimilarity is an equivalence on states

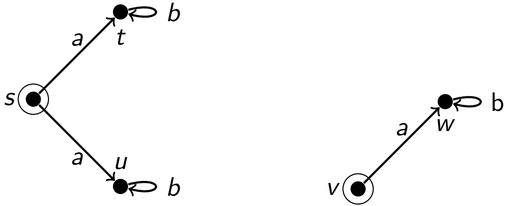
$p \sim q$ *iff* $p\ R\ q$ for some (strong) bisimulation R

Exhibiting bisimilarity

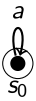
To show $p_1 \sim p_2$, we give a relation R such that R is a bisimulation and $p_1 \; R \; p_2$.

Examples: Give bisimulations to show

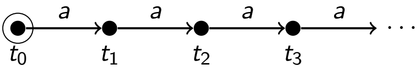
- $a \parallel b \sim a.b + b.a$
- On transition systems, $s \sim v$ where



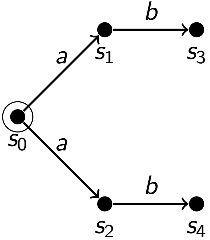
Examples: Looping



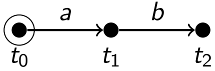
$? \sim ?$



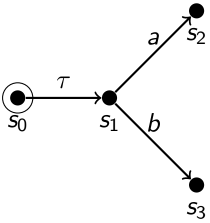
Examples: Inessential branching



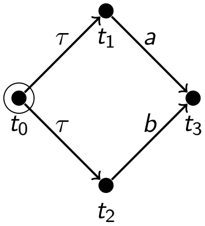
? $\sim$?



Examples: Internal choice



$? \sim ?$



Bisimulations

If R, S, R_i for $i \in I$ are strong bisimulations then so are:

1. Id, the identity relation the set of states of any transition system
2. R^{op}, the converse/opposite relation
3. $R \circ S$, the composition (when the transition systems involved match up so that the composition makes sense)
4. $\bigcup_{i \in I} R_i$, the union (when the relations are over the same transition systems)

(1)–(3) imply that $\sim$ is an equivalence relation, and (4) that $\sim$ is a bisimulation.

Equational properties of bisimulation

$+$ and $\|$ are commutative and associative w.r.t. $\sim$, with unit **nil**

If $p \sim q$ then:

- $\alpha.p \sim \alpha.q$
- $p + r \sim q + r$
- $p \parallel r \sim q \parallel r$
- $p \setminus L \sim q \setminus L$
- $p[f] \sim q[f]$

Equational properties of bisimulation

$+$ and $\parallel$ are commutative and associative w.r.t. $\sim$, with unit **nil**

If $p \sim q$ then:

- $\alpha.p \sim \alpha.q$
- $p + r \sim q + r$
- $p \parallel r \sim q \parallel r$
- $p \setminus L \sim q \setminus L$
- $p[f] \sim q[f]$

...bisimilarity is a congruence

Expansion laws for CCS

In general,

$$p \sim \sum\{\alpha.p' \mid p \xrightarrow{\alpha} p'\}$$

We can use this to remove everything but prefixing and sums:

Suppose $p \sim \sum_{i \in I} \alpha_i.p_i$ and $q \sim \sum_{j \in J} \beta_j.q_j$.

$$p \setminus L \quad \sim \quad \sum\{\alpha_i.(p_i \setminus L) \mid \alpha_i \notin L\}$$

Expansion laws for CCS

In general,

$$p \sim \sum \{\alpha.p' \mid p \xrightarrow{\alpha} p'\}$$

We can use this to remove everything but prefixing and sums:

Suppose $p \sim \sum_{i \in I} \alpha_i.p_i$ and $q \sim \sum_{j \in J} \beta_j.q_j$.

$$\begin{aligned} p \setminus L &\sim \sum \{\alpha_i.(p_i \setminus L) \mid \alpha_i \notin L\} \\ p[f] &\sim \sum \{f(\alpha_i).(p_i[f]) \mid i \in I\} \end{aligned}$$

Expansion laws for CCS

In general,

$$p \sim \sum \{\alpha.p' \mid p \xrightarrow{\alpha} p'\}$$

We can use this to remove everything but prefixing and sums:

Suppose $p \sim \sum_{i \in I} \alpha_i.p_i$ and $q \sim \sum_{j \in J} \beta_j.q_j$.

$$\begin{array}{rcl}
p \setminus L & \sim & \sum \{\alpha_i.(p_i \setminus L) \mid \alpha_i \notin L\} \\
p[f] & \sim & \sum \{f(\alpha_i).(p_i[f]) \mid i \in I\} \\
p \parallel q & \sim & \displaystyle\sum_{i \in I} \alpha_i.(p_i \parallel q) + \sum_{j \in J} \beta_j.(p \parallel q_j) \\
& + & \sum \{\tau.(p_i \parallel q_j) \mid \alpha_i = \overline{\beta_j}\}
\end{array}$$

Strong bisimilarity and specifications

An example:

$$Sem \stackrel{\mathrm{def}}{=} get.put.Sem$$

$$P_1 \stackrel{\mathrm{def}}{=} \overline{get}.a_1.a_2.\overline{put}.P_1$$
$$P_2 \stackrel{\mathrm{def}}{=} \overline{get}.b_1.b_2.\overline{put}.P_2$$

$$Sys \stackrel{\mathrm{def}}{=} (Sem \parallel P_1 \parallel P_2) \setminus \{get, put\}$$

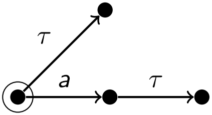
$$Spec \stackrel{\mathrm{def}}{=} \tau.a_1.a_2.Spec + \tau.b_1.b_2.Spec$$

Do we have

$$? \qquad Sys \sim Spec \qquad ?$$

Weak bisimulation

Hiding τ actions

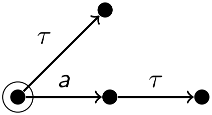


$$\stackrel{\tau}{\Rightarrow} \stackrel{\text{def}}{=} (\stackrel{\tau}{\rightarrow}^*)$$

$$\stackrel{a}{\Rightarrow} \stackrel{\text{def}}{=} (\stackrel{\tau}{\Rightarrow}\stackrel{a}{\rightarrow}\stackrel{\tau}{\Rightarrow})$$

Weak bisimulation

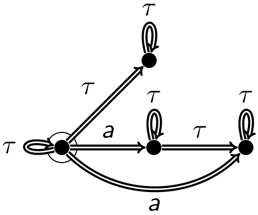
Hiding τ actions



$$\stackrel{\tau}{\Rightarrow} \stackrel{\text{def}}{=} (\stackrel{\tau}{\rightarrow}^{*})$$

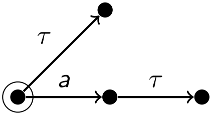
$$\stackrel{a}{\Rightarrow} \stackrel{\text{def}}{=} (\stackrel{\tau}{\Rightarrow}\stackrel{a}{\rightarrow}\stackrel{\tau}{\Rightarrow})$$

We get a transition system



Weak bisimulation

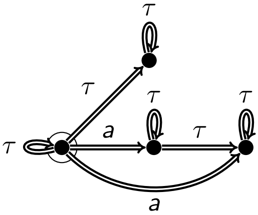
Hiding τ actions



$$\stackrel{\tau}{\Rightarrow} \stackrel{\text{def}}{=} (\stackrel{\tau}{\to}^{*})$$

$$\stackrel{a}{\Rightarrow} \stackrel{\text{def}}{=} (\stackrel{\tau}{\Rightarrow}\stackrel{a}{\to}\stackrel{\tau}{\Rightarrow})$$

We get a transition system



Weak bisimulation is bisimulation w.r.t. $\Rightarrow$

Weak bisimulation

A weak bisimulation is a relation R between states for which
If $p \; R \; q$ then:

1. $\forall \alpha, p'. \quad p \stackrel{\alpha}{\Rightarrow} p' \quad \Longrightarrow$
   $\exists q'. \quad q \stackrel{\alpha}{\Rightarrow} q' \quad \& \quad p' \; R \; q'$
2. $\forall \alpha, q'. \quad q \stackrel{\alpha}{\Rightarrow} q' \quad \Longrightarrow$
   $\exists p'. \quad p \stackrel{\alpha}{\Rightarrow} p' \quad \& \quad p' \; R \; q'$

Weak bisimulation

A weak bisimulation is a relation R between states for which
If $p \; R \; q$ then:

1. $\forall \alpha, p'. \quad p \stackrel{\alpha}{\Rightarrow} p' \quad \Longrightarrow$
   $\exists q'. \quad q \stackrel{\alpha}{\Rightarrow} q' \quad \& \quad p' \; R \; q'$
2. $\forall \alpha, q'. \quad q \stackrel{\alpha}{\Rightarrow} q' \quad \Longrightarrow$
   $\exists p'. \quad p \stackrel{\alpha}{\Rightarrow} p' \quad \& \quad p' \; R \; q'$

Weak bisimulation is not a congruence $\rightsquigarrow$ observational congruence.