

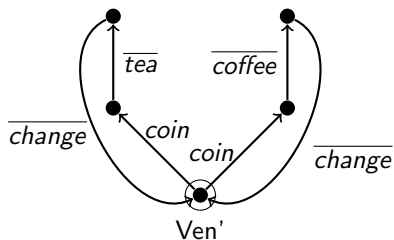
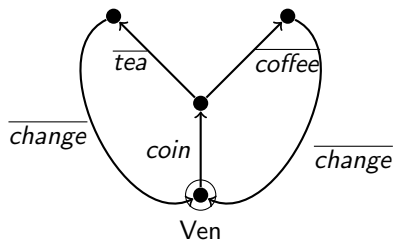
Topics in Concurrency

Lectures 4

Glynn Winskel

28 January 2020

Two vending machine implementations



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

Specification and correctness:

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

Language equivalences

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

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

such that

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

- 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.

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

(Strong) bisimilarity is an equivalence on states

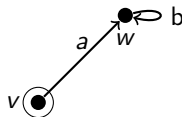
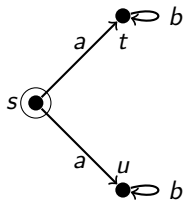
$$p \sim q \quad \text{iff} \quad p R q \text{ for some (strong) bisimulation } R$$

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

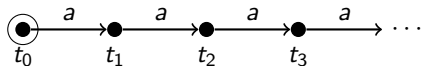
- $a \parallel b \sim a.b + b.a$
- On transition systems, $s \sim v$ where



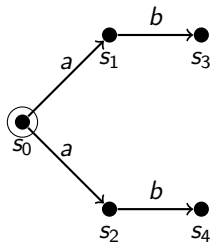
Examples: Looping



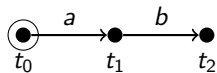
? ~?



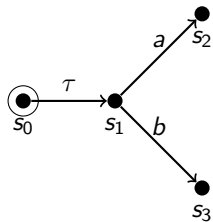
Examples: Inessential branching



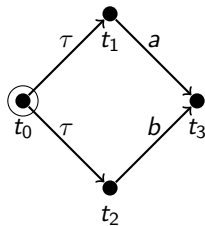
? ~?



Examples: Internal choice



? ~?



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 \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 $q \sim \sum_{j \in J} \beta_j.q_j$.

$$\begin{aligned} p \setminus L &\sim \sum \{ \alpha_i.(p_i \setminus L) \mid \alpha_i \notin L \} \\ p[f] &\sim \sum \{ f(\alpha_i).(p_i[f]) \mid i \in I \} \\ p \parallel q &\sim \sum_{i \in I} \alpha_i.(p_i \parallel q) + \sum_{j \in J} \beta_j.(p \parallel q_j) \\ &\quad + \sum \{ \tau.(p_i \parallel q_j) \mid \alpha_i = \bar{\beta}_j \} \end{aligned}$$

Strong bisimilarity and specifications

An example:

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

$$P_1 \stackrel{\text{def}}{=} \overline{get}.a_1.b_1.\overline{put}.P_1$$

$$P_2 \stackrel{\text{def}}{=} \overline{get}.a_2.b_2.\overline{put}.P_2$$

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

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

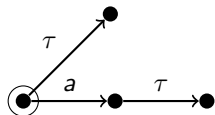
$$Spec' \stackrel{\text{def}}{=} \tau.a_1.b_1.\tau.Spec' + \tau.a_2.b_2.\tau.Spec'$$

Do we have

$$? \quad Sys \sim Spec \quad ? \quad \text{or} \quad ? \quad Sys \sim Spec' \quad ?$$

Weak bisimulation

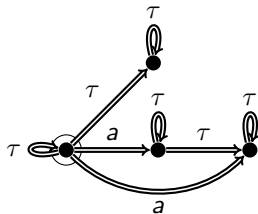
Hiding τ actions



$$\xRightarrow{\tau} \stackrel{\text{def}}{=} (\xrightarrow{\tau}^*)$$

$$\xRightarrow{a} \stackrel{\text{def}}{=} (\xRightarrow{\tau} \xrightarrow{a} \xRightarrow{\tau})$$

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 $\forall \alpha, p'. \quad p \xRightarrow{\alpha} p' \implies \exists q'. \quad q \xRightarrow{\alpha} q' \ \& \ p' R q'$
- 2 $\forall \alpha, q'. \quad q \xRightarrow{\alpha} q' \implies \exists p'. \quad p \xRightarrow{\alpha} p' \ \& \ p' R q'$

Weak bisimulation is not a congruence \rightsquigarrow observational congruence.