Formal Methods in software development

a.a.2017/2018

Prof.Anna Labella

From automata to reactive systems

They are supposed to go on forever as

- Communication protocols
- Operative systems
- Command and control devices

Their features

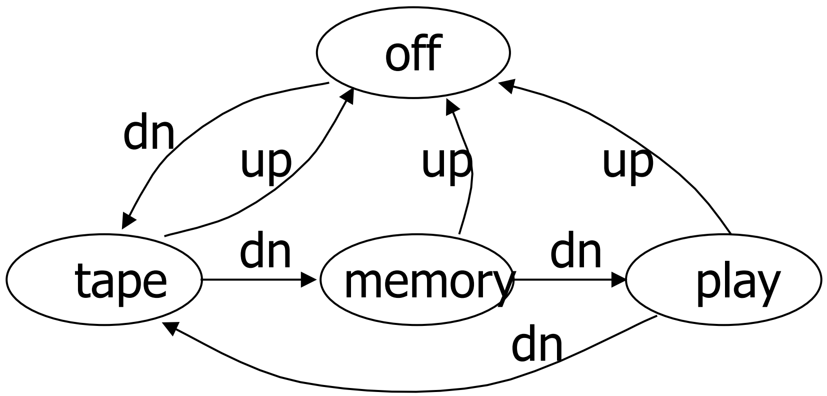
- Communication
- Observability
- Non determinism vs determinism
- Synchronous vs asynchronous

Labeled transition systems

- TS=(Σ,S, Δ, S_0), where
  - Σ a non empty finite alphabet
  - S a non empty finite set of *states*
  - $\Delta \subseteq S \times \Sigma \times S$ is a transition relation,
  - $S_0 \subseteq S$ is the set of initial states
- Similar to a nondeterministic finite state automaton, with possibly more than one initial state, but without terminal states
- Similar to a labeled Kripke model as we have seen in temporal logic

- A transition system generates (finite or infinite) words $w_0w_1w_2$... iff there are states $s_0s_1s_2s_3$... s.t. $s_0 \in S_0$ and $(s_i,w_i,s_{i+1}) \in \Delta$
- A state is identified through the possibilities it offers to go on
- termination and deadlock

Example: a recorder



T = <S, Σ, Δ, s_0>   without terminal states

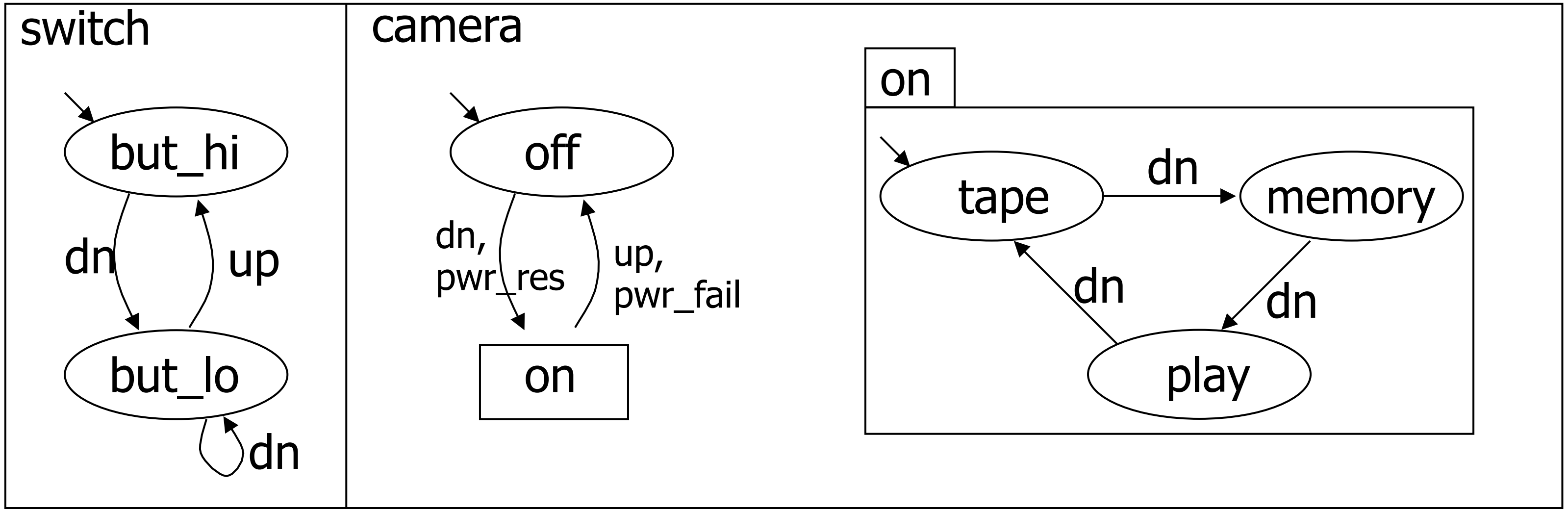
1. Σ={up, dn}
2. S={off, tape, memory, play}
3. Δ={(off,dn,tape), (tape,up,off), (tape,dn,memory), (memory,up,off), (memory,dn,play), (play,dn,tape), (play,up,off)}
4. s_0={off}

Parallel transition systems

- Parallel transition system $T=(T_1,\ldots,T_n)$
  - each T_i is a transition system
  - $S_i \cap S_j = \emptyset$
- interleaving semantics
  - on its private alphabet, each T_i can make an independent move
  - synchronization is via common events
- example:

power switch and camcorder mode

Example



- T=(switch, camera)
- {pwr_fail, pwr_res} are private to camera
- synchronization alphabet {up,dn}
- how big is the state space?

The *global transition system* T associated with a parallel transition system $(T_1,\ldots,T_n)$ is defined as $T=(\Sigma, S, \Delta, S_0)$, where

- $\Sigma = \cup\Sigma_i$
- $S = S_1 \times\ldots\times S_n$
- $S_0 = S_{1,0} \times\ldots\times S_{n,0}$, and
- $((s_1,\ldots,s_n),a,(s_1‘,\ldots,s_n‘))\in\Delta$ iff
  - when a is an asynchronous move
    - $a\in\Sigma_i$, $((s_i),a,(s_i‘))\in\Delta_i$, and
    - then $s_k= s_k‘$ for all $k\neq i$
  - when a is the result of a synchronisation of T_i and T_j
    - $((s_i),a_i,(s_i‘))\in\Delta_i$ and $((s_{ji}),a_j,(s_j‘))\in\Delta_j$, and
    - $s_k= s_k‘$ for all $k\neq i,j$

Process Equivalences

Sameness of behaviour = equivalence of states

Many process equivalences have been proposed

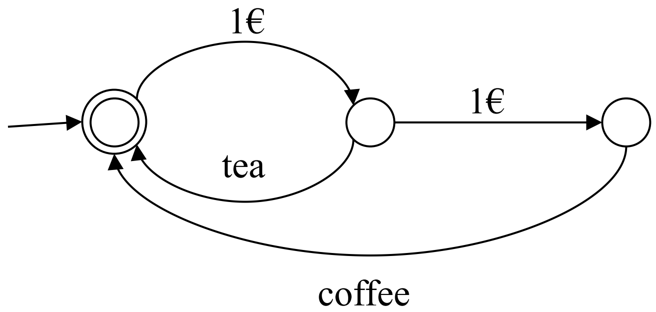
For instance: $q_1 \sim q_2$ iff

- q_1 and q_2 have the same paths, *or*
- q_1 and q_2 may always refuse the same interactions, *or*
- q_1 and q_2 pass the same tests, *or*
- q_1 and q_2 satisfy the same temporal formulas, *or*
- q_1 and q_2 have identical branching structure

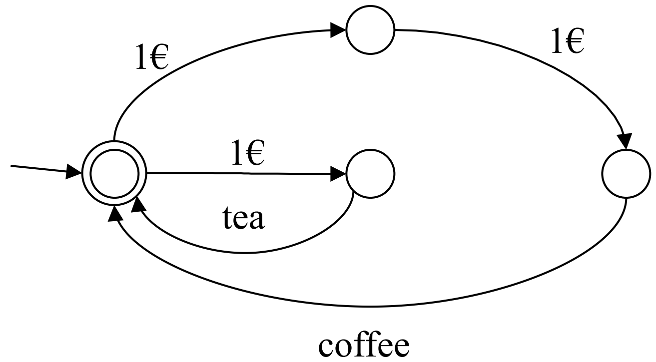
CCS: Focus on <u>bisimulation equivalence</u>

Finite State Automata

- Coffee machine A_1:



- Coffee machine A_2:



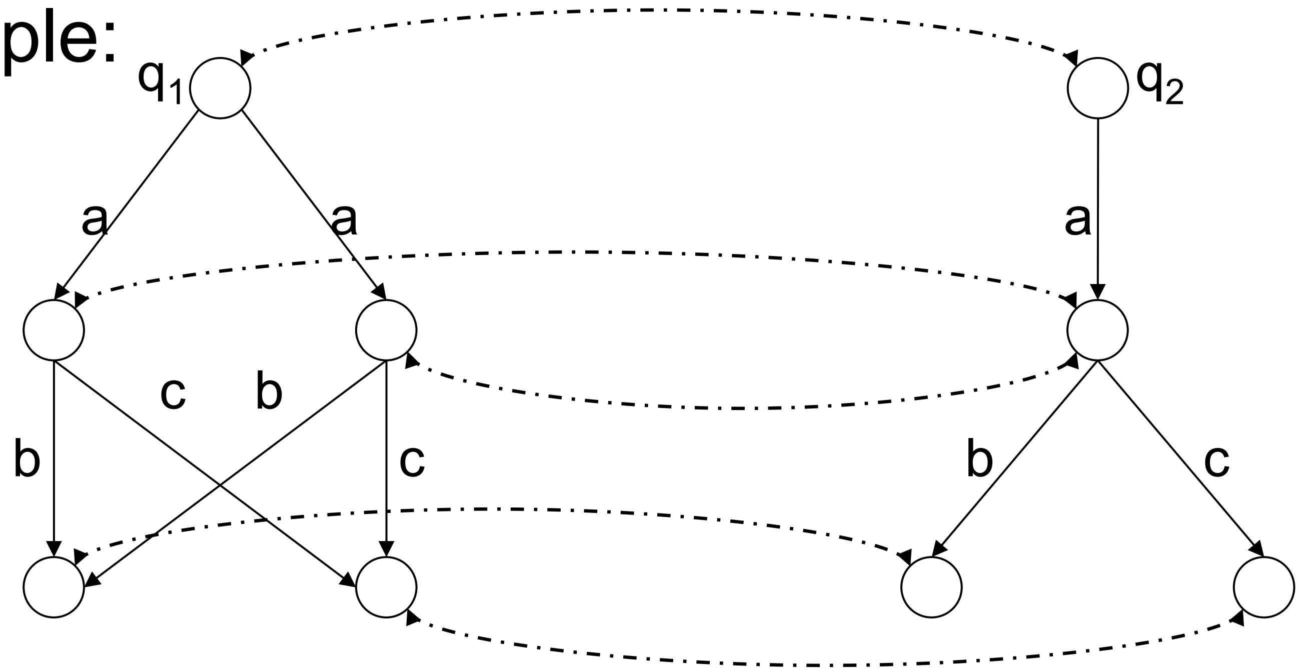
- Are the two machines ”the same”?

Bisimulation Equivalence

Intuition: $q_1 \sim q_2$ iff q_1 and q_2 have same branching structure

Idea: Find relation which will relate two states with the same transition structure, and make sure the relation is preserved

Example:



Strong Bisimulation Equivalence

Given: Labelled transition system $T = (Q,\Sigma,R)$
Looking for a relation $S \subseteq Q \times Q$ on states

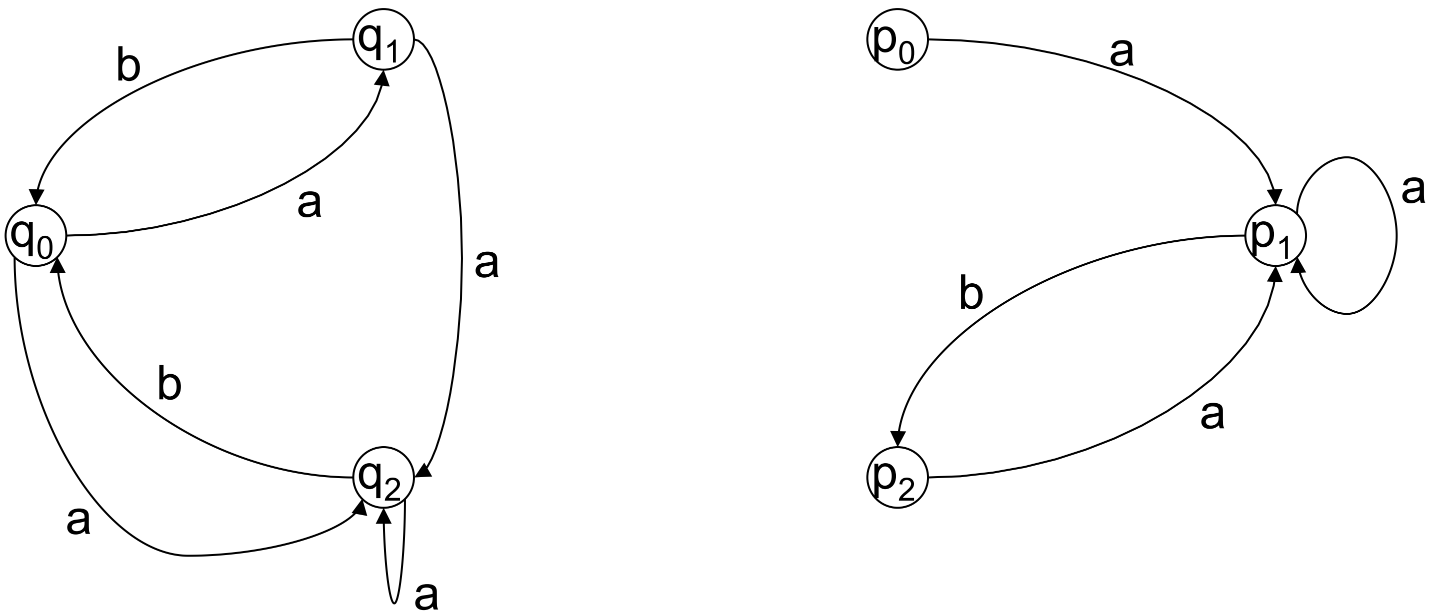
S is a *strong bisimulation relation* if whenever $q_1 \; S \; q_2$ then:

- $q_1 \rightarrow^{\alpha} q_1'$ implies $q_2 \rightarrow^{\alpha} q_2'$ for some q_2' such that $q_1' \; S \; q_2'$
- $q_2 \rightarrow^{\alpha} q_2'$ implies $q_1 \rightarrow^{\alpha} q_1'$ for some q_1' such that $q_1' \; S \; q_2'$

q_1 and q_2 are *strongly bisimilar* iff $q_1 \; S \; q_2$ for some strong bisimulation relation S

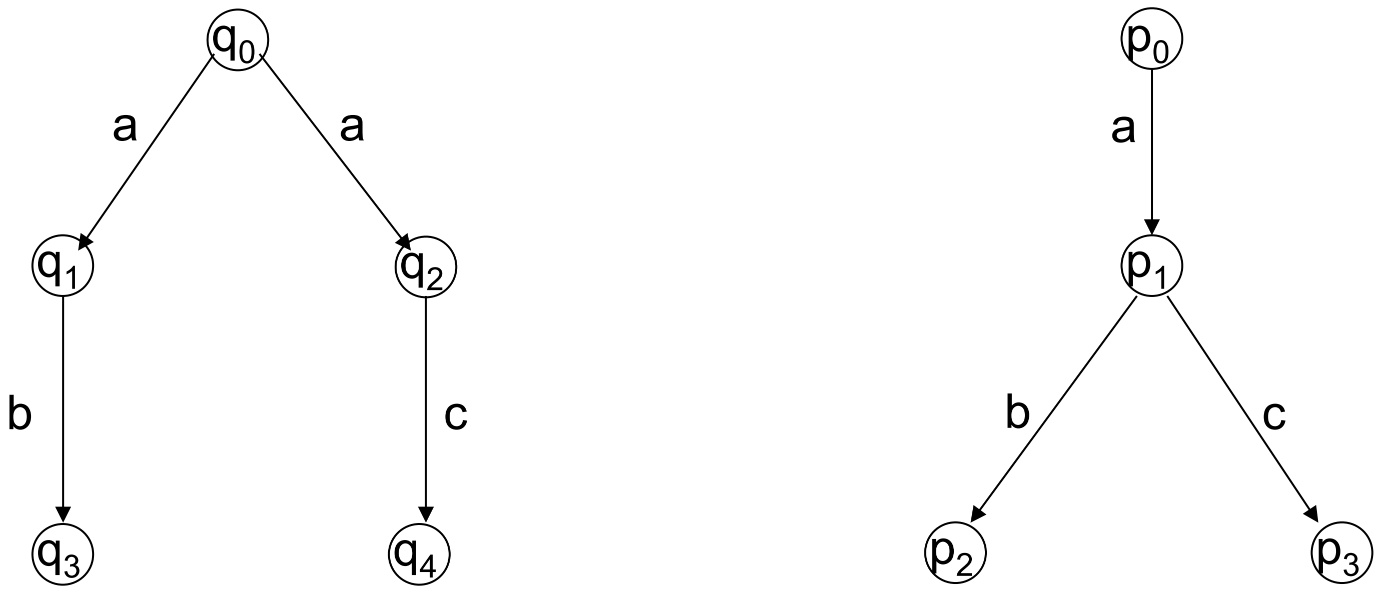
$q_1 \sim q_2$: q_1 and q_2 are strongly bisimilar

Exercise



Does $q_0 \sim p_0$ hold?

Exercise



Does $q_0 \sim p_0$ hold?

Weak Transitions

What to do about internal activity?

τ: Transition label for activity which is not externally visible

- $q \Rightarrow^{\varepsilon} q'$ iff $q = q_0 \rightarrow^{\tau} q_1 \rightarrow^{\tau} \ldots \rightarrow^{\tau} q_n = q'$, $n \geq 0$
  $q \Rightarrow^{\tau} q'$ iff $q \Rightarrow^{\varepsilon} q'$
  $q \Rightarrow^{\alpha} q'$ iff $q \Rightarrow^{\varepsilon} q_1 \rightarrow^{\alpha} q_2 \Rightarrow^{\varepsilon} q'$ $(\alpha \neq \tau)$

Beware that $\Rightarrow^{\tau} = \Rightarrow^{\varepsilon}$ (non-standard notation)

Observational equivalence, v.1.0: Bisimulation equivalence with $\Rightarrow$ in place of $\rightarrow$

Let $q_1 \approx q_2$ iff $q_1 \sim q_2$ with $\Rightarrow^{\alpha}$ in place of $\rightarrow^{\alpha}$

Observational Equivalence

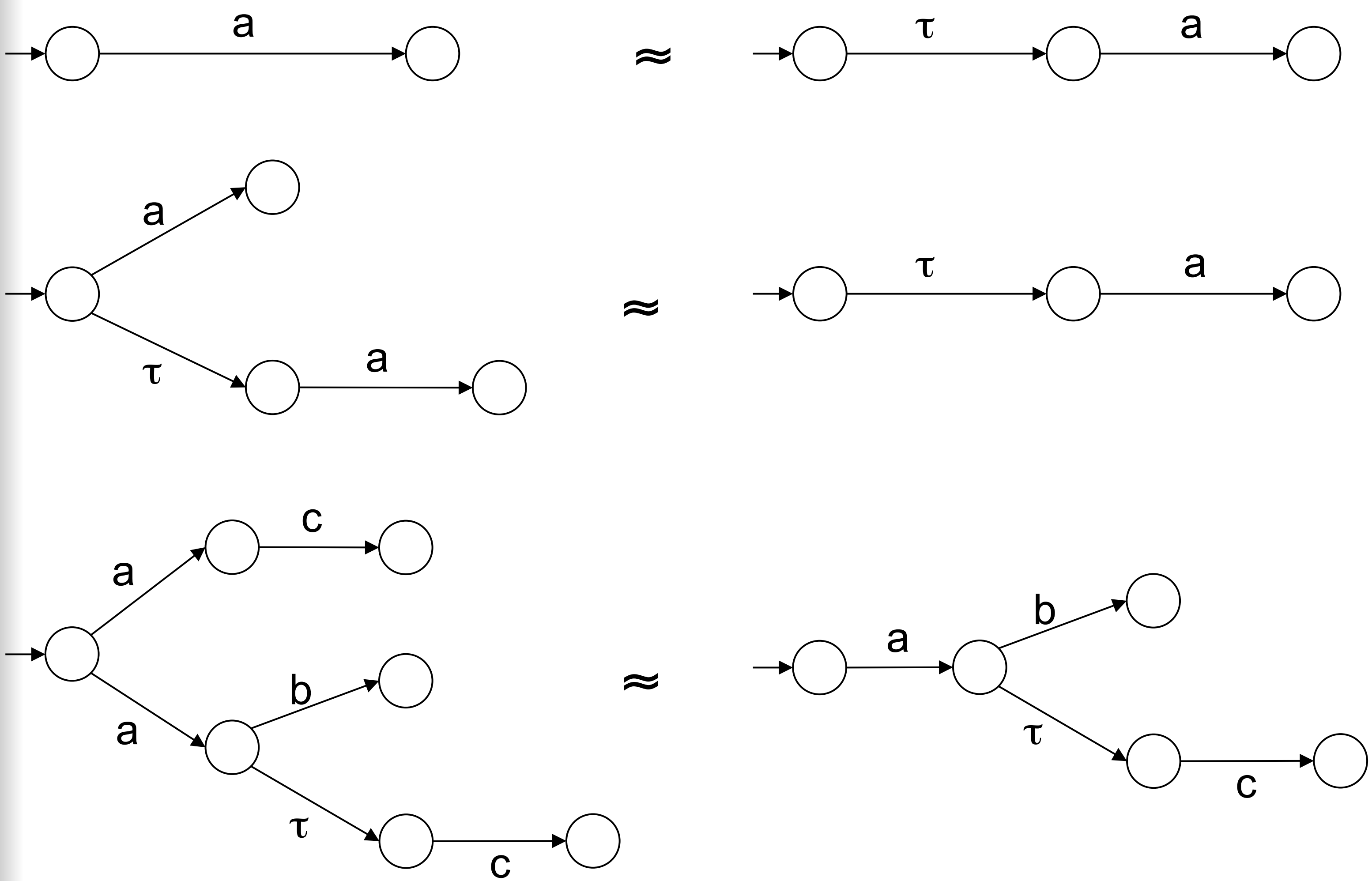
Let $S \subseteq Q \times Q$. The relation S is a *weak bisimulation relation* if whenever q_1 S q_2 then:

- $q_1 \rightarrow^{\alpha} q_1'$ implies $q_2 \Rightarrow^{\alpha} q_2'$ for some q_2' such that q_1' S q_2'
- $q_2 \rightarrow^{\alpha} q_2'$ implies $q_1 \Rightarrow^{\alpha} q_1'$ for some q_1' such that q_1' S q_2'

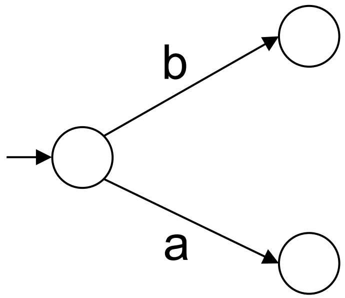
q_1 and q_2 are *observationally equivalent*, or *weakly bisimulation equivalent,* if q_1 S q_2 for some weak bisimulation relation S

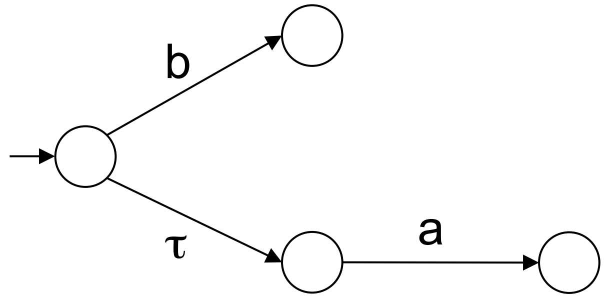
$q_1 \approx q_2$: q_1 and q_2 are observationally equivalent/weakly bisimilar

Exercises

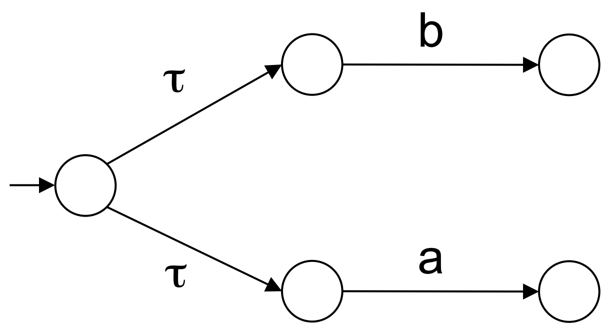


Exercises





All three are inequivalent



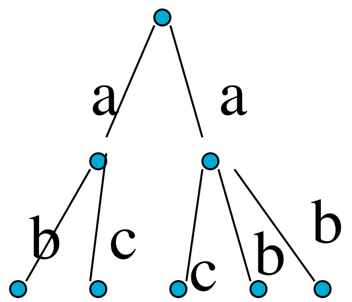
Bisimulations

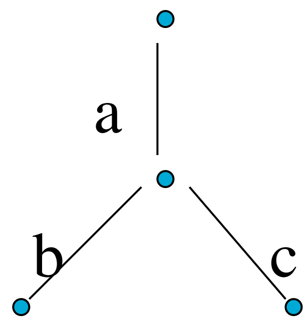
- Strong
- Weak
- Branching

Bisimulations

- Strong

If a process/state can do a move, then the other one can do the same and viceversa.

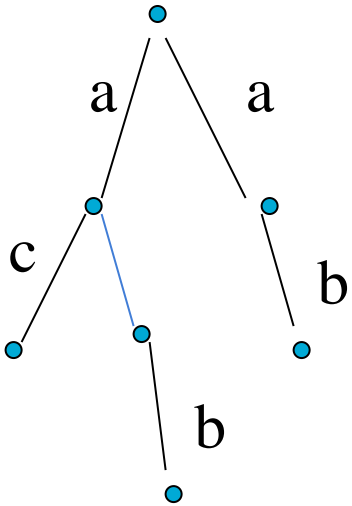


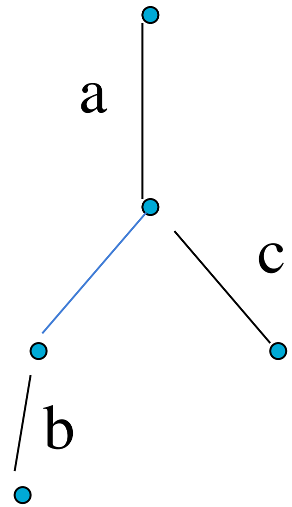


Bisimulations

- weak

A process can go through (non equivalent, non consecutive) states with invisible moves Trying to simulate the other one.





Bisimulations

- branching

A process can go through different (equivalent) states with invisible moves while the other does not move, but has the same possibilities.