University of Southampton

COMP2212 Programming Language Concepts

Simulations

Dr Julian Rathke

University of Southampton

Simulations, intuitively

- In the previous lecture we considered trace equivalence as a possible way of comparing concurrent behaviour.
  - Branching points in behaviour are not captured by traces
- Can we define a notion of equivalence that accounts for branching points?
- This is the idea behind **simulation** :
- A state y in a labelled transition system can simulate a state x if
  - whenever x can do some action, becoming x' in the process, y can reply with the same action, becoming y' in the process
  - moreover, now y' can still do everything that x' can do… and so forth
- This latter point is how we address the branching - any observations possible of x' must still be possible in the matching y' state.
- This is closely related to the notion of **refinement** in formal systems such as Event B

Simulations, formally

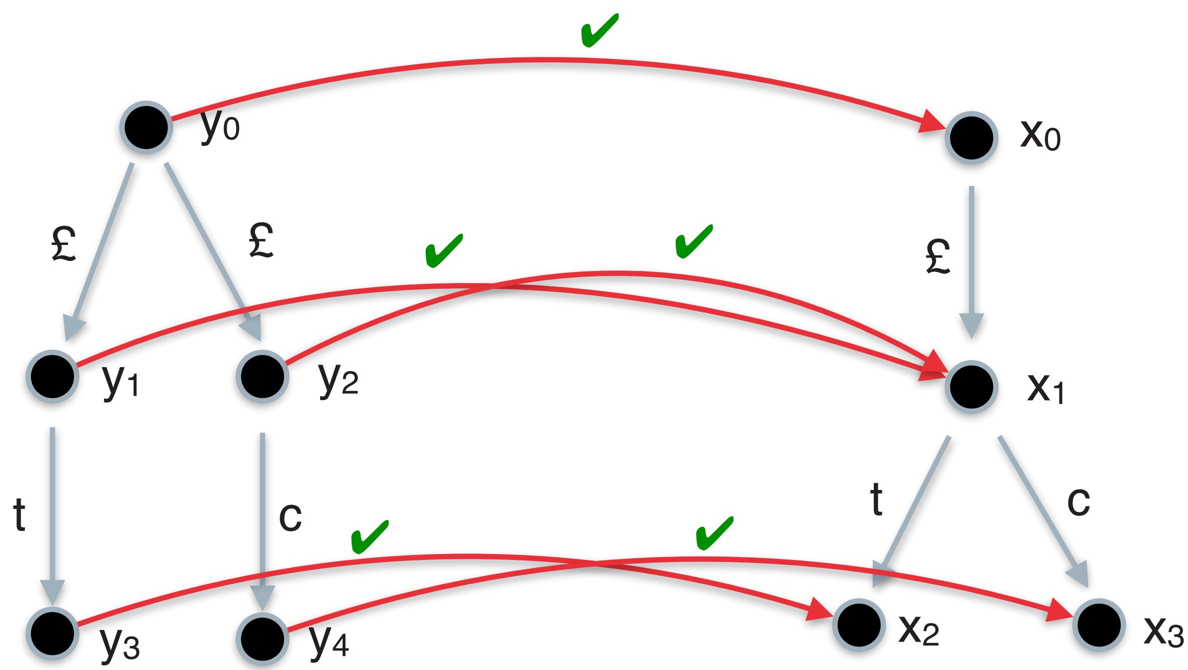
- Recall from Foundations of CS that a binary relation on a set X is simply a subset $R \subseteq X \times X$
- Suppose that **L** = (X, Σ, L) is a labelled transition system (LTS)
- A binary **relation** R on states of **L** is called a **simulation** on **L** if R satisfies the following condition:

> Whenever x R y and $x \xrightarrow{a} x'$ for some state x' then
>
> there exists a state y' such that $y \xrightarrow{a} y'$ and $(x',y') \in R$

- For example, the identity relation Id = { (x,x) | x in X } is always a simulation on **L**
- We define a simulation between two labelled transition systems simply by considering the above definition on the disjoint union of the two LTSs.

Example

- Let R be the binary relation $\{ (y_0, x_0), (y_1, x_1), (y_2, x_1), (y_3, x_2), (y_4, x_3) \}$ defined on the (union of) the labelled transition systems below.
- We can check that this is a simulation!



Whenever y R x and

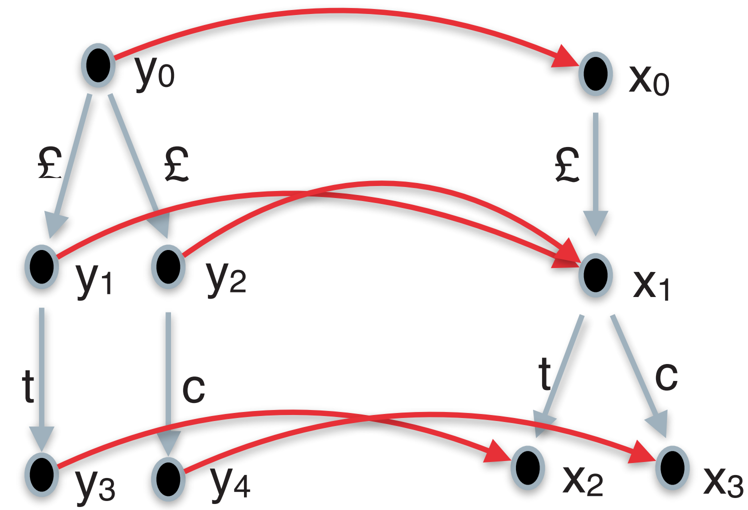
$y \xrightarrow{a} y'$ for some state y' then there exists a state x' such that

$x \xrightarrow{a} x'$ and $(y', x') \in R$

if $(x,y) \in R$ we say that **y simulates x**

Example, more formally

- Let R be the binary relation $\{ (y_0, x_0), (y_1, x_1), (y_2, x_1), (y_3, x_2), (y_4, x_3) \}$ defined on the (union of) the labelled transition systems below.
- We can check that this is a simulation!



(y0,x0) $\in$ R and y0 $\xrightarrow{£}$ y1 is matched by

x0 $\xrightarrow{£}$ x1 with (y1,x1) $\in$ R

(y0,x0) $\in$ R and y0 $\xrightarrow{£}$ y2 is matched by

x0 $\xrightarrow{£}$ x1 with (y2,x1) $\in$ R

(y1,x1) $\in$ R and y1 $\xrightarrow{t}$ y3 is matched by

x1 $\xrightarrow{t}$ x2 with (y3,x2) $\in$ R

(y2,x1) $\in$ R and y2 $\xrightarrow{c}$ y4 is matched by

x1 $\xrightarrow{c}$ x3 with (y4,x3) $\in$ R

(y3,x2) $\in$ R and (y4,x3) $\in$ R have no outgoing transitions

Similarity

Theorem: the union of two simulation relations R1, R2 on an LTS **L** is a simulation relation on **L**

Proof : to show this we take any $(x,y) \in R1 \cup R2$, then (x,y) is either in R1 or R2.

Suppose, wlog that it is in R1. Then suppose $x \xrightarrow{a} x'$ for some a and some x'. We know that R1 is a simulation so there exists a $y \xrightarrow{a} y'$ such that $(x',y') \in R1$. And hence $(x',y') \in R1 \cup R2$ also.

Theorem : for any given LTS L there is a largest simulation on L

Proof : the largest simulation on L is the union of all simulations on L

We write $\leq$ to denote the largest simulation on an LTS - we call this relation **similarity**

Coinduction

- Similarity is an instance of something called a **coinductively defined relation** in mathematics.
  - It is the **largest** relation such that ...

cf. inductively defined relations are the **smallest** such that ...

- This gives us an easy proof technique for showing that something is in the similarity relation:
- To show that $(x,y) \in \leq$ (written $x \leq y$) for some states x,y of an LTS, we simply need to show that $(x,y) \in R$ for **some** relation R that is a simulation.
- Why?
- If $(x,y) \in R$ and R is a simulation, then because $\leq$ is the largest simulation then $R \subseteq \leq$ and hence $x \leq y$ also.
- This is known as the coinduction principle and using it gives us a coinductive proof technique.

Using simulations

- **Q**. How to show that y simulates x?
  - **A**. Construct a simulation that contains the pair (x,y)

- **Q**. How to show that y **does not** simulate x?
  - **A**. Show that all relations that contain (x,y) are not simulations?
  - Ahh! This is trickier - we need to consider **all** relations that contain (x,y)
  - There are potentially very many of them!
  - We need another proof technique for this.

The simulation game

Imagine a game in which two players must pick matching moves in an LTS trying force each other to fail to match the next move. We can use this idea as a proof technique!

Rules of the Game:

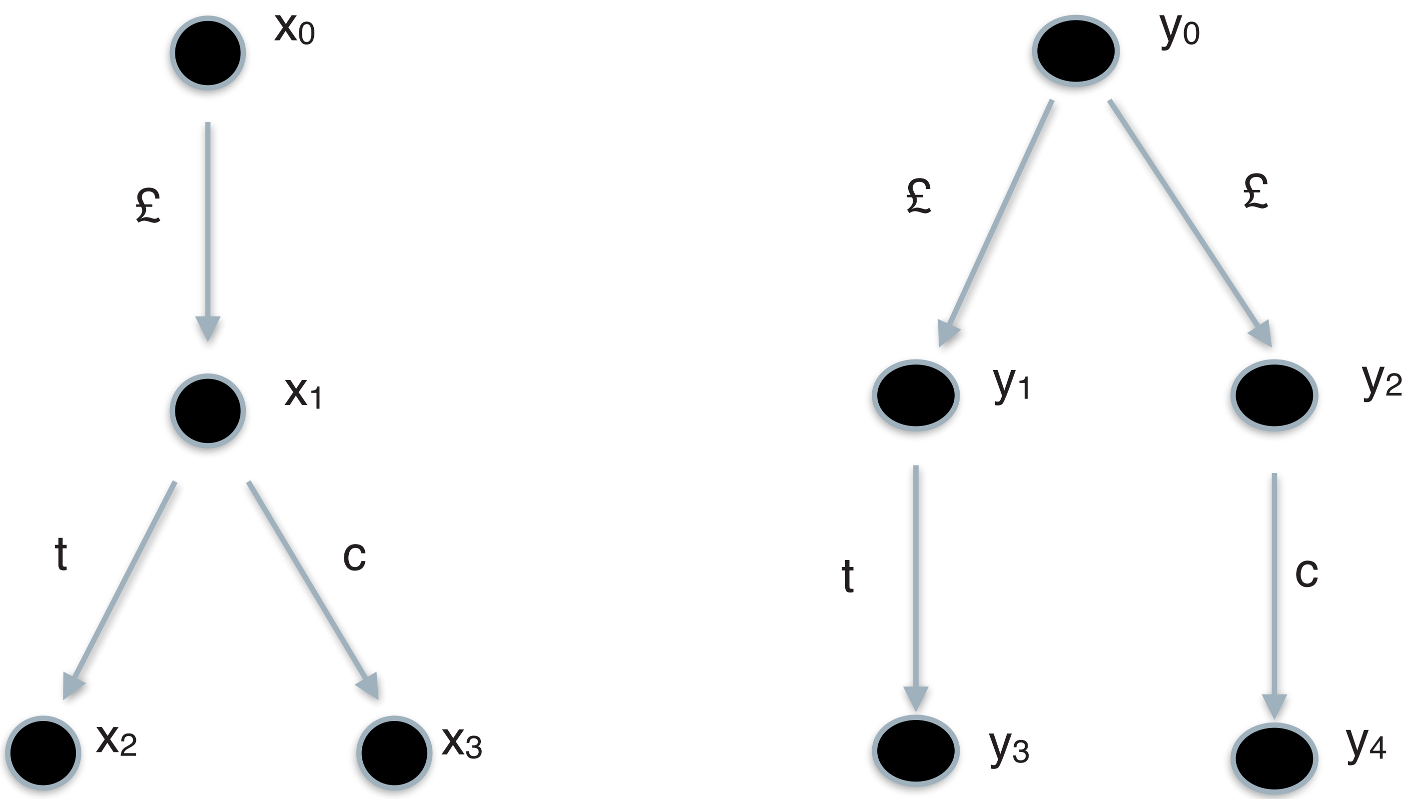
- You are playing against a demon 😈 . The game starts at position (x,y).

  1. The demon picks a move starting from x, they choose $x \xrightarrow{a} x'$ say.
  2. You must start from y and choose a matching move to y' say, so $y \xrightarrow{a} y'$
  3. The game goes back to Step 1, changing the position to (x',y')

- If at any point a player cannot make a move, that player loses
- If the game goes on forever, you win.
- nb. The demon always plays moves reachable from x and we always play moves reachable from y.

Let's Play!

I'll be the demon



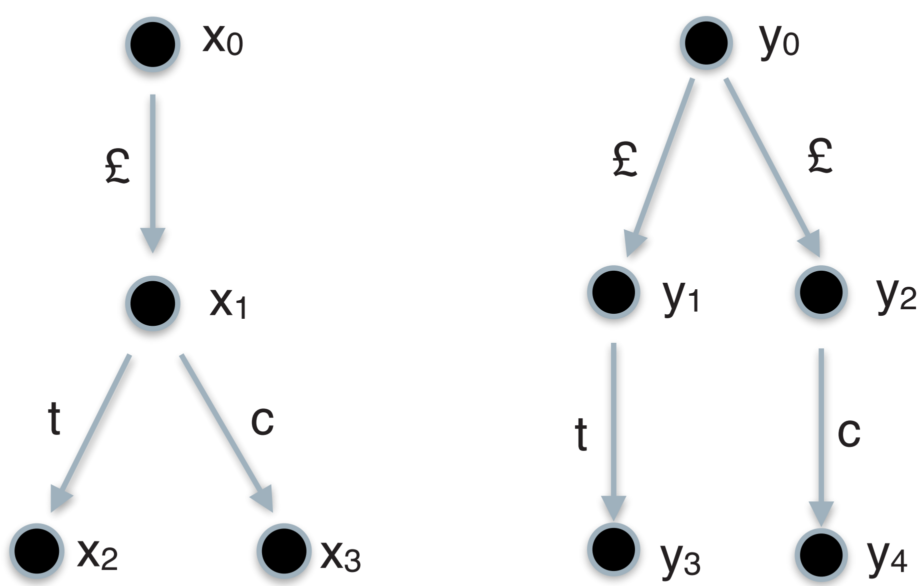
Anyone reckon they can beat me?

Strategies

- A strategy for the demon in this game is a mapping from every pair of states (x,y) to a move $x \xrightarrow{a} x'$
- Similarly, a strategy for the non-demon player in this game is a mapping from every (x',y) and label a to a move $y \xrightarrow{a} y'$
- A strategy for a player is called a **winning strategy** if no matter which moves the other player makes, as long as they follow the winning strategy they are guaranteed to win.
- **Theorem** : for any two states, $x \leq y$ if and only if the non-demon player has a winning strategy in the simulation game.
- **Proof** : a little lengthy to show here but in essence you use the winning strategy to continue matching moves until all nodes reachable from x,y have been visited.
- This gives us a proof technique to show that two states are not in the similarity relation - i.e. demon has a winning strategy from (x,y) means $x \not\leq y$

Example continued

- We give a winning strategy for the demon, when starting in position (x_0, y_0).
  - From (x0,y0) the demon picks the £ move to x_1
  - From (x1,y1) the demon picks the c move to x3
  - From (x1,y2) the demon picks the t move to y3
- No matter what the other player chooses, demon will always win.

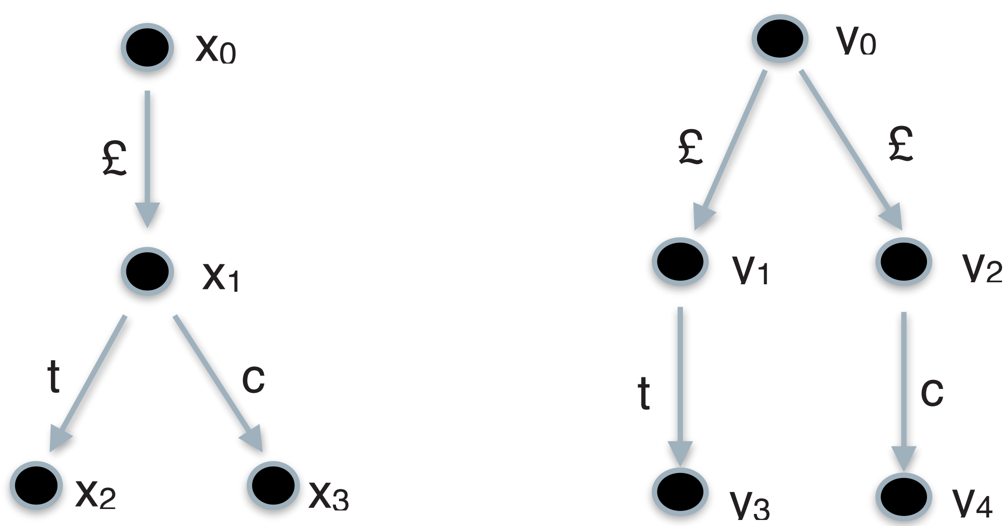


What if demon played in the y states instead?

Would either player have a winning strategy?

Simulation equivalence

- States x and y are said to be **simulation equivalent** if both $x \leq y$ and $y \leq x$
  - thus to check that two states are simulation equivalent, we typically need to construct two simulations - one in each direction.
- e.g. We know that x_0 and y_0 below are **not** simulation equivalent. Indeed, we have shown:
  - $y_0 \leq x_0$, by constructing a simulation
  - but **not** $x_0 \leq y_0$, by playing the simulation game



Simulation and trace equivalence

- **Theorem**. If x and y are simulation equivalent then they are also trace equivalent.
- **Proof** : Lengthy but in essence, any trace of actions a1a2 ... aN starting from a state x can be matched using the simulation relation to give the same trace of actions from y, and v.v.
- We have seen that the converse is not true: there are trace equivalent states (x_0 and y_0) that are not simulation equivalent.

We say that simulation equivalence is a **finer** equivalence (it distinguishes more) than trace equivalence. Conversely, trace equivalence is **coarser** than simulation equivalence.

Next Lecture

Bisimulation