Simulations, intuitively

- A state $y$ 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

- Closely related to the notion of refinement
Simulations, formally

- Recall from COMP1215 that a relation on a set \( X \) is a subset \( R \subseteq X \times X \).

- Suppose that \( (X, \Sigma, L) \) is a labelled transition system.

  - A simulation is a relation on states that satisfies the following condition:
    - Whenever \( xRy \) and \( x \xrightarrow{a} x' \) then there exists \( y' \) such that \( y \xrightarrow{a} y' \) and \( (x', y') \in R \).

- If \( (x, y) \in R \) we say that \( y \) simulates \( x \).

- We sometimes write \( x \leq y \) if there exists a simulation that contains \( (x, y) \).
Example

- Let \( R=\{ (y_0, x_0), (y_1, x_1), (y_2, x_1), (y_3, x_2), (y_4, x_3) \} \)

- check that this is a simulation!
Properties of simulations

- **Theorem.** Unions of simulations are simulations.
- There exists the largest simulation — it is the union of all simulations.
  - The largest simulation, written ≤, is called **similarity**.
- Simulation is an instance of something called **coinduction** in mathematics. More on this next week!
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? This sounds tedious…
The simulation game

- We are playing against a demon. The game starts at position \((x, y)\).
  1. The demon picks a move \(x \xrightarrow{a} x'\)
  2. We must choose a \(y'\) such that \(y \xrightarrow{a} y'\)
  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, we win
- If the Demon has a **winning strategy**, then \(y\) does not simulate \(x\)
- If we have a winning strategy then \(x \leq y\)
We know that $x_0$ simulates $y_0$. Now let us play the game and check that $y_0$ does not simulate $x_0$. 
We give a winning strategy for the demon, when starting in position \((x_0, y_0)\).

- The demon picks the £ move to \(x_1\)
- we have to reply with one of the two available moves, moving either to \(y_1\) or to \(y_2\)
- if we picked \(y_1\) then the demon can play the c move and we are stuck (we lose)
- if we picked \(y_2\) then the demon can play the t move and we are stuck (we lose)
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
Example

❖ We know that $x_0$ and $y_0$ are not simulation equivalent. Indeed, we have shown:

❖ $y_0 < x_0$, by constructing a simulation

❖ but not $x_0 < y_0$, by playing the simulation game
Simulation equivalence and trace equivalence

- **Theorem.** If \( x \) and \( y \) are simulation equivalent then they are also trace equivalent.

- We have seen that the converse is not true: there are trace equivalent states \((x_0 \text{ and } y_0)\) that are not simulation equivalent.

- Sometimes we say that simulation equivalence is a **finer** equivalence (it distinguishes more) than trace equivalence. Conversely, trace equivalence is **coarser** than simulation equivalence.
Example - an evil coffee machine

- $z_0$ can act like $x_0$, by going to $z_1$ but sometimes (maybe a race condition in the implementation?) it may go to $z_2$ or $z_3$

- $x_0$ and $z_0$ are simulation equivalent (why?). Should we be able to distinguish them?