COMP 2212 : Semantics of Concurrency Exercise Sheet

1. Consider the labelled transition system below

$$\begin{tikzcd}
x_0 \arrow[r, "c"] \arrow[d, "a"] & x_3 \arrow[d, "a"] \\
x_1 \arrow[r, "c"] & x_2
\end{tikzcd}$$

   Give the set of traces from x_0, x_1, x_2, x_3.

   Solution. Traces from x_2: $\{\epsilon\}$. Traces from x_3: $\{\epsilon, a\}$. Traces from x_1: $\{\epsilon, c\}$. Traces from x_0: $\{\epsilon, a, c, ac, ca\}$.

2. Consider the labelled transition system below.

$$\begin{tikzcd}
y_0 \arrow[r, bend left, "a"] \arrow[r, bend right, "c"'] & y_1 \arrow[r, bend left, "a"] \arrow[r, bend right, "c"'] & y_2
\end{tikzcd}$$

   (a) Are x_0 (from the previous question) and y_0 trace equivalent?
   Solution. Traces from y_0: $\{\epsilon, a, c, aa, ac, ca, cc\}$. Since $\{\epsilon, a, c, ac, ca\} \neq \{\epsilon, a, c, aa, ac, ca, cc\}$, x_0 and y_0 are *not* trace equivalent.

   (b) Show that y_0 simulates x_0.
   Solution. We show that $R = \{(x_0, y_0), (x_1, y_1), (x_3, y_1), (x_2, y_2)\}$ is a simulation. We need to check that each pair satisfies the defining property of simulations.

   From (x_0, y_0) there are two possible moves for x_0. First $x_0 \xrightarrow{a} x_1$. Here y_0 can respond with $y_0 \xrightarrow{a} y_1$ and we know that $x_1 R y_1$. Also $x_0 \xrightarrow{c} x_3$. Now $y_0 \xrightarrow{c} y_1$ and $x_3 R y_1$.

   From (x_1, y_1), $x_1 \xrightarrow{c} x_2$. But $y_1 \xrightarrow{c} y_2$ and $x_2 R y_2$.

   From (x_3, y_1), $x_3 \xrightarrow{a} x_2$. Here $y_1 \xrightarrow{a} y_2$ and $x_2 R y_2$.

   From (x_2, y_2) there is nothing to check, since there are no moves from x_2.

   (c) Play the simulation game to show that x_0 does not simulate y_0.
   Solution. We start in position (y_0, x_0) and the demon chooses $y_0 \xrightarrow{a} y_1$. We have to respond with $x_0 \xrightarrow{a} x_1$. The game now

continues from (y_1, x_1). The demon now chooses $y_1 \xrightarrow{a} y_2$, and we are stuck since we cannot play an a move from x_1. The demon wins – note that this is a winning strategy since we had no choices to make, the demon is guaranteed to win every time, if he follows this strategy. Thus x_0 does not simulate y_0.

3. Consider the two labelled transition system below.

$$
\begin{tikzcd}
x_0 \arrow[r, "c"] \arrow[d, "a"] & x_3 \arrow[d, "b"] \\
x_1 \arrow[r, "b"] \arrow[loop below, "a"] & x_2
\end{tikzcd}
\qquad
\begin{tikzcd}
y_0 \arrow[d, "a"] \arrow[dd, bend right, "c"'] & \\
y_1 \arrow[r, "b"] \arrow[d, "a"] & y_2 \\
y_3 \arrow[r, "b"] \arrow[loop below, "a"] & y_4
\end{tikzcd}
$$

   (a) Show that there is a simulation relation on this LTS containing the pair (x_0, y_0).

   Solution. From (x_0, y_0): If $x_0 \xrightarrow{a} x_1$, then $y_0 \xrightarrow{a} y_1$ and $x_1 R y_1$. If $x_0 \xrightarrow{c} x_3$, then $y_0 \xrightarrow{c} y_3$ and $x_3 R y_3$.

   From (x_1, y_1): If $x_1 \xrightarrow{a} x_1$, then $y_1 \xrightarrow{a} y_3$ and $x_1 R y_3$. If $x_1 \xrightarrow{b} x_2$ then $y_1 \xrightarrow{b} y_2$ and $x_2 R y_2$.

   From (x_1, y_3): If $x_1 \xrightarrow{a} x_1$ then $y_3 \xrightarrow{a} y_3$ and $x_1 R y_3$. If $x_1 \xrightarrow{b} x_2$ then $y_3 \xrightarrow{b} y_4$ and $x_2 R y_4$.

   From (x_3, y_3): If $x_3 \xrightarrow{b} x_2$ then $y_3 \xrightarrow{b} y_4$ and $x_2 R y_4$.

   From (x_2, y_2) and (x_2, y_4) there is nothing to check.

   (b) Play the bisimulation game to show that x_0 and y_0 are *not* bisimilar.

   Solution. We start in the state (x_0, y_0). The demon chooses to play at x_0, choosing $x_0 \xrightarrow{x} x_3$. We have to match with $y_0 \xrightarrow{c} y_3$.

   The game continues from (x_3, y_3). The demon now chooses to play at y_3, and chooses $y_3 \xrightarrow{a} y_3$. We cannot respond, since there is no a move from x_3 — we lose! This is a winning strategy for demon as the player had no choice in their responses so the demon will win however player responds.

4. In this question we'll consider a variation of trace equivalence. In order to define this we will need to make some preliminary defintions.

   - We define a *failure trace* as a sequence of pairs of the form
   $$(a_1, F_1)(a_2, F_2)\dots(a_n, F_n)$$
   where the F_i are refusal sets.
   - A refusal set for a state x in an LTS with alphabet Σ is defined as
   $$\mathsf{Ref}(x) = \{b \in \Sigma \mid x \stackrel{b}{\not\to} x' \text{ for any } x'\}$$
   - The failure traces of a state x is defined as
   $$\mathsf{FailTr}(x) = \{(a_1, F_1)ft \mid x \xrightarrow{a_1} x', F_1 = \mathsf{Ref}(x'),\ \text{and } ft \in FailTr(x')\}$$

   Two states x and y of an LTS are said to be *failure trace equivalent* iff
   $$\mathsf{FailTr}(x) = \mathsf{FailTr}(y)$$

   (a) Give an argument that failure trace equivalence implies trace equivalence.

   Solution. Suppose x and y are failure trace equivalent and suppose there is a trace t in the traces of x where t is $x \xrightarrow{a_1} x_1 \dots \xrightarrow{a_n} x_n$. We can make a failure trace ft of x from t by simply calculating the refusal sets $F_i = Ref(x_i)$. Note that ft is therefore also a failure trace of y. By simply forgetting the refusal sets in ft we see that t is also a trace of y. Because this is true for any trace t of x and a symmteric argument shows the same for traces from y, we have that x and y must be trace equivalent.

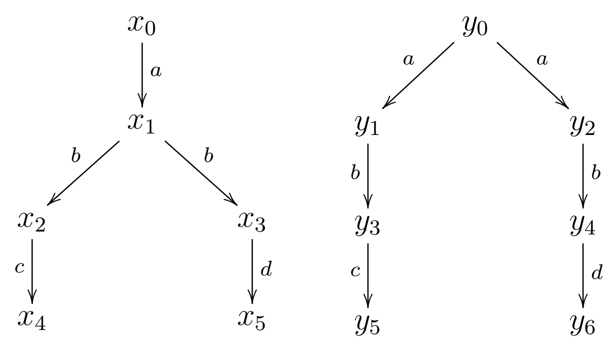
   (b) Give a counter-example that shows that trace equivalence does not imply failure trace equivalence.

   Solution. Consider

$$\begin{tikzcd}
 & x_0 \arrow[d, "a"] & & & & y_0 \arrow[dl, "a"'] \arrow[dr, "a"] & \\
 & x_1 \arrow[dl, "b"'] \arrow[dr, "c"] & & & y_1 \arrow[d, "b"'] & & y_2 \arrow[d, "c"] \\
x_2 & & x_3 & & y_3 & & y_4
\end{tikzcd}$$

Then x_0 and y_0 are easily seen to be trace equivalence but they are not failure trace equivalent because y_0 has failure trace $(a, \{a, c\})$ but x_0 does not have this same failure trace as the refusal set after a from x_0 is just $\{a\}$.

(c) Consider the following two LTSs defined over the alphabet $\Sigma = \{a, b, c, d\}$.



Are the states x_0 and y_0 failure trace equivalent?

Solution. Yes, the failure traces are as follows

$$\begin{aligned}&\mathsf{FailTr}(x_0) = \{(\epsilon, \{b, c, d\}), (a, \{a, c, d\}),\\ &(a, \{a, c, d\})(b, \{a, b, d\}), (a, \{a, c, d\})(b, \{a, b, c\}),\\ &(a, \{a, c, d\})(b, \{a, b, d\})(c, \{a, b, c, d\}), (a, \{a, c, d\})(b, \{a, b, c\})(d, \{a, b, c, d\})\}\end{aligned}$$

and the failure traces from y_0 are exactly the same.

(d) Are the states x_0 and y_0 bisimilar?

Solution. No. A winning strategy for demon in the bisimulation game is to begin play in y_0 and select the a move to y_1. The player must respond by playing to x_1. The demon now switches to play from x_1 and chooses the b move to x_3. The player must respond by playing the b move to y_3. The demon now plays the d move to x_5 and the player cannot respond. Hence the demon always wins.

(e) Use your answers to these to describe the relationship between failure trace equivalence and bisimilarity.

Solution. Bisimilarity is strictly finer than failure trace equivalence. The counter-example in Part (d) shows that failure trace equivalence does not imply bisimilarity. For the converse, note that bisimiarity between x and y implies that the refusal sets at

x and y are equal. Hence, as bisimilarity also implies trace equivalence, we see that it implies failure trace equivalence also.