Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Digital Systems and Design Automation Group

HS 2022

Prof. L. Josipović / J. Xu
based on Prof. L. Thiele's material

Discrete Event Systems

Solution to Exercise Sheet 11

1 Specifying Formal Properties Using Temporal Logic

Notation: w_valid is the valid signal of the write data channel; aw_valid is the valid signal of the write address channel.

a) $\mathbf{AG}\ (valid \rightarrow \mathbf{AF}\ ready)$

b) $\mathbf{AG}\ \mathbf{AF}\ (ready)$

c) $\mathbf{AG}\ ((valid \wedge \neg ready) \rightarrow \mathbf{AX}\ valid)$

d) $\mathbf{AG}\ ((valid \wedge \neg ready) \rightarrow (data \leftrightarrow \mathbf{AX}\ data))$

2 Temporal Logic

a) (i) $Q = \{0, 1, 2, 3\}$

(ii) $Q = \{0, 3\}$

(iii) (AX a) holds for $\{2, 3\}$, thus $Q = \{1, 2\}$

(iv) (a `AND` EX `NOT`(a)) is *true* for states where a is *true* and there exists a direct successor for which it is not. Only state 0 satisfy this (from it you can transition to 1, where a does not hold). Moreover, state 0 is reachable for all states in this automaton ("from all states there exists a path going through 0 at some point"). Hence $Q = \{0, 1, 2, 3\}$

b) (i) $\neg \mathrm{AF}\, Z \equiv \mathrm{EG}\, \neg Z$

(ii) We will first compute the function $Q_k \models \mathrm{EG}\, \neg Z$, which we can compute quite easily (following the procedure given in the lecture), and take the negation in the end.

$$\begin{aligned} Q_0 &= S \backslash Z \\ Q_{i+1} &= Q_i \cap Pre(Q_i, f) \\ k &= \min\{i \mid Q_{i+1} = Q_i\} \\ Q_{\mathrm{AF}\ Z} &= Z \backslash Q_k \end{aligned}$$

The main idea is that we start with the states that are not in Z. Then, at each iteration, we create an intersection between the current set of states, and all predecessors from which we can reach one of the states in the set. By doing this, we will remove any states from which there exists some future, in which Z does not hold. We stop the iteration once nothing changes anymore (we define k to be the first index for which the set of states remains the same). Hence, we express have $Q_k \models \mathrm{EG}\, \neg Z$. What is left to do is to negate the final set (every state which is not present in Q_k).

(iii) We translate the procedure above directly into an algorithm:

Require: ψ_Z, ψ_f

$\psi_{cur} \leftarrow \neg \psi_Z$
$\psi_{next} \leftarrow \psi_{cur} \wedge \psi_{Pre(\psi_{cur}, f)}$
while $\psi_{cur} \neq \psi_{next}$ **do**
  $\psi_{cur} \leftarrow \psi_{next}$
  $\psi_{next} \leftarrow \psi_{cur} \wedge \psi_{Pre(\psi_{cur}, f)}$
end while
return $\psi_{\mathrm{AF}\, Z} = \neg \psi_{cur}$

3 Comparison of Finite Automata

a)

$$\psi_A(x_A, x'_A, u) = \neg x_A \neg x'_A \neg u + \neg x_A x'_A u + x_A x'_A u + x_A \neg x'_A \neg u$$
$$\psi_B(x_B, x'_B, u) = \neg x_B \neg x'_B \neg u + \neg x_B x'_B u + x_B x'_B \neg u + x_B \neg x'_B u$$

b)

$$\begin{aligned}\psi_f(x_A, x'_A, x_B, x'_B) =&(\neg x_A x'_A + x_A x'_A) \cdot (\neg x_B x'_B + x_B \neg x'_B)\\ &+(\neg x_A \neg x'_A + x_A \neg x'_A) \cdot (\neg x_B \neg x'_B + x_B x'_B)\\ &=\neg x_A x'_A \neg x_B x'_B + \neg x_A x'_A x_B \neg x'_B + x_A x'_A \neg x_B x'_B + x_A x'_A x_B \neg x'_B\\ &+\neg x_A \neg x'_A \neg x_B \neg x'_B + \neg x_A \neg x'_A x_B x'_B + x_A \neg x'_A \neg x_B \neg x'_B + x_A \neg x'_A x_B x'_B\end{aligned}$$

c) Computation of the reachable states is performed incrementally. Starts with the initial state of the system $\psi_{X_0}(x_A, x_B) = \neg x_A x_B$ and then add the successors until reaching a fix-point,

$$\begin{aligned}\psi_{X_1}(x'_A, x'_B) =&\psi_{X_0}(x'_A, x'_B) + (\exists(x_A, x_B) : \psi_{X_0}(x_A, x_B) \cdot \psi_f(x_A, x'_A, x_B, x'_B))\\ &= \neg x'_A x'_B + \neg x'_A x'_B + x'_A \neg x'_B\\ &= \neg x'_A x'_B + x'_A \neg x'_B\\ \psi_{X_2}(x'_A, x'_B) =&\neg x'_A x'_B + x'_A \neg x'_B + x'_A x'_B + \neg x'_A \neg x'_B\\ \psi_{X_3}(x'_A, x'_B) =&\neg x'_A x'_B + x'_A \neg x'_B + x'_A x'_B + \neg x'_A \neg x'_B = \psi_{X_2} \quad \rightarrow \text{ the fix-point is reached!}\end{aligned}$$

$$\Rightarrow \quad \boxed{\psi_X(x_A, x_B) = \neg x_A x_B + x_A \neg x_B + x_A x_B + \neg x_A \neg x_B}$$

d) Here you first need to express the output function of each automaton, that is the feasible combinations of states and outputs, $\psi_{g_A} = \neg x_A \neg y_A + x_A y_A$ and $\psi_{g_B} = \neg x_B y_B + x_B \neg y_B$. Then the reachable outputs are the combination of the reachable states and the outputs functions, that is,

$$\begin{aligned}\psi_Y(y_A, y_B) =&(\exists(x_A, x_B) : \psi_X \cdot \psi_{g_A} \cdot \psi_{g_B})\\ &= y_A y_B + \neg y_A \neg y_B + \neg y_A y_B + y_A \neg y_B\end{aligned}$$

e) From the reachable output function, we see that these automata are not equivalent. Indeed, there exists a reachable output admissible ($\psi_Y((y_A, y_B) = (0, 1)) = 1$) for which $y_A \neq y_B$.

Another way of looking at it: $\psi_Y \cdot (y_A \neq y_B) \neq 0$ where $(y_A \neq y_B) = \neg y_A y_B + y_A \neg y_B$.