Discrete Event Systems
Solution to Exercise Sheet 12

1 Comparison of Finite Automata

Here are two simple finite automata:

![Finite Automata Diagram]

For each, we have a one bit encoding for the states \(x_A\) and \(x_B\), one binary output \(y_A\) and \(y_B\), and one common binary input \(u\). We want to verify whether or not these two automata are equivalent. This can be done through the following steps:

a) Express the characteristic function of the transition relation for both automaton, \(\psi_r(x, x', u)\).

b) Express the joint transition function, \(\psi_f\).
   
   Reminder: \(\psi_f(x_A, x'_A, x_B, x'_B) = (\exists u : \psi_A(x_A, x'_A, u) \cdot \psi_B(x_B, x'_B, u))\).

c) Express the characteristic function of the reachable states, \(\psi_X(x_A, x_B)\).

d) Express the characteristic function of the reachable output, \(\psi_Y(y_A, y_B)\).

e) Are the automata equivalent? Justify with a simple calculus.

\[\psi_A(x_A, x'_A, u) = x_A x'_A u + x_A x'_A u + x_A x'_A u + x_A x'_A u\]
\[\psi_B(x_B, x'_B, u) = x_A x'_A u + x_B x'_B u + x_B x'_B u + x_B x'_B u\]

\[\psi_f(x_A, x'_A, x_B, x'_B) = (x_A x'_A + x_A x'_A) \cdot (x_B x'_B + x_B x'_B) + (x_A x'_A + x_A x'_A) \cdot (x_B x'_B + x_B x'_B)\]
\[= x_A x'_A x_B x'_B + x_A x'_A x_B x'_B + x_A x'_A x_B x'_B + x_A x'_A x_B x'_B + x_A x'_A x_B x'_B + x_A x'_A x_B x'_B + x_A x'_A x_B x'_B + x_A x'_A x_B x'_B\]

\[\psi_X(x_A, x_B) = \psi_X x_A + (3(x_A, x_B) : \psi_X x_A + \psi_f(x_A, x'_A, x_B, x'_B))\]
\[= x_A x_B + x_A x_B + x_A x_B = \psi_X\]
\(\Rightarrow \psi_X = x_A x_B + x_A x_B + x_A x_B\)
d) Here you first need to express the output function of each automaton, that is the feasible combinations of states and outputs,
\[ \psi_{g_A} = x_A y_A + x_A y_A \]  and  \[ \psi_{g_B} = x_B y_B + x_B y_B \]
Then the reachable outputs are the combination of the reachable states and the outputs functions, that is,
\[ \psi_Y(y_A, y_B) = (\exists (x_A, x_B) : \psi_X \cdot \psi_{g_A} \cdot \psi_{g_B}) = y_A y_B + \overline{y_A} y_B + y_A \overline{y_B} \]

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 \(y_A \neq y_B\).
One other way of saying this: \(\psi_Y \cdot (y_A \neq y_B) \neq 0\), where \((y_A \neq y_B) = y_A y_B + y_A y_B\).

2 Temporal Logic

a) We consider the following automaton. The property \(a\) is true on states 0 and 3.

For each of the following CTL formula, list all the states for which it holds true.

(i) \(EF \  \ a\)
(ii) \(EX \ AX \ a\)
(iii) \(EF \ (\ a \ AND \ EX \ NOT(\ a) \ )\)

(i) \(Q = \{0, 1, 2, 3\}\)
(ii) \((AX \ a)\) holds for \(\{2, 3\}\), thus \(Q = \{1, 2\}\)
(iii) \((a \ AND \ EX \ NOT(\ a))\) 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) Given the transition function \(\psi_f(x, x')\) and the characteristic function \(\psi_Z(x)\) for a set \(Z\), write a small pseudo-code which returns the characteristic function of \(\psi_{AF} Z(x)\). It can be expressed as symbolic boolean functions, like \(x_A x_B x' A' B' + x_A x_B x' A' B'\).

Hint: To do this, simply use the classic boolean operators \(\ AND, \ OR, \ NOT\) and \(\! =\). You can also use an existence selector \(EXISTS(a)\). For a given argument \(a\), it returns the set \(\{x : \exists x', \ a(x, x')\ \text{is true}\}\).

Hint: It can be useful to reformulate \(AFZ\) as another CTL formula.

Here the trick is to remember that \(AFZ \equiv NOT(EG NOT(Z))\). Hence, one can compute the function for \(EG NOT(Z)\) quite easily (following the procedure given in the lecture) and take the negation in the end. A possible pseudo-code doing this is the following,
Require: $\psi_Z, \psi_f$

current = NOT($\psi_Z$);
next = current AND (EXISTS($\psi_f$ AND current));
while next != current do
    current = next;
    next = current AND (EXISTS($\psi_f$ AND current));
end while
return $\psi_{AF} Z =$NOT(current);

▷ Equivalence in term of sets:

▷ $X_0$

▷ $X_1 = X_0 \cap Pre(X_0, f)$

▷ $X_1 ! = X_{i-1}$

▷ $X_i = X_{i-1} \cap Pre(X_{i-1}, f)$

▷ $X_f \models EG NOT(Z)$

▷ $X_f \models AF Z = NOT(EG NOT(Z))$