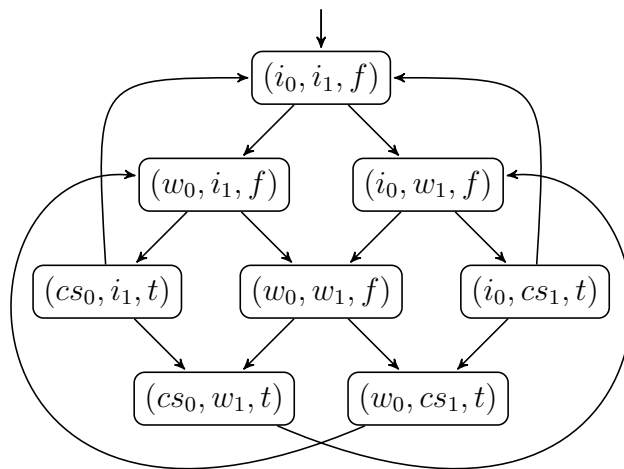


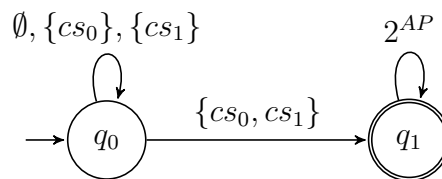
Model Checking – Sample Solution 6

Exercise 6.1

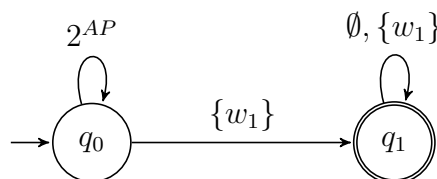
- (i) $\mathcal{T} = (S, \rightarrow, r)$, where $S = \{i_0, w_0, cs_0\} \times \{i_1, w_1, cs_1\} \times \{t, f\}$ for modeling three locations in m and two possible values of s .



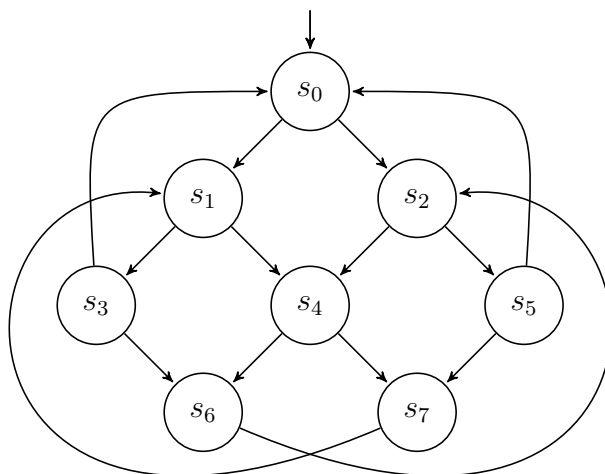
- (ii) $\phi_a = \neg \mathbf{F}(cs_0 \wedge cs_1)$, where $AP_a = \{cs_0, cs_1\}$ and
 $\phi_b = \mathbf{G}(w_0 \rightarrow \mathbf{F} cs_0)$, where $AP_b = \{w_0, cs_0\}$
- (iii) $\neg \phi_a = \mathbf{F}(cs_1 \wedge cs_2)$. So, $\mathcal{B}_{\neg \phi_a}$ can be constructed as follows:



$\neg \phi_b = \mathbf{F}(w_0 \wedge \mathbf{G} \neg cs_0)$. So, $\mathcal{B}_{\neg \phi_b}$ can be constructed as follows:



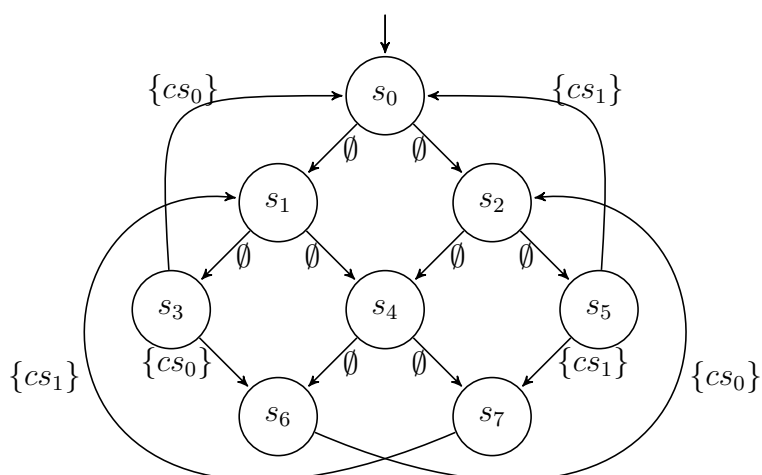
(iv) Let rename the states to $S = \{s_0, \dots, s_7\}$.



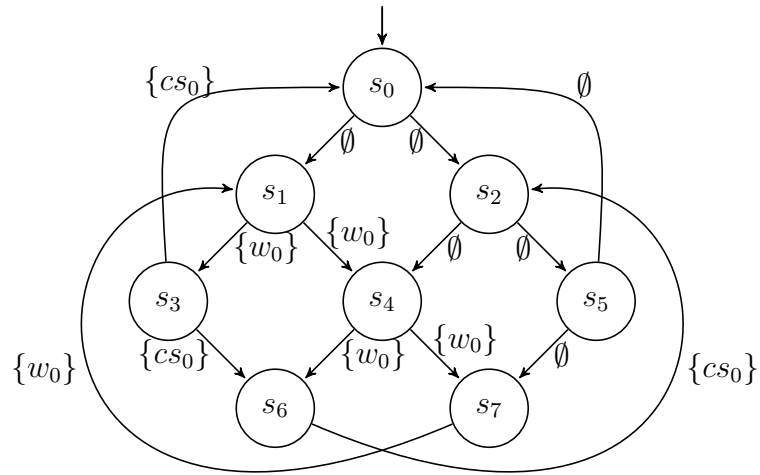
$$\mathcal{K}_a = (S, \rightarrow, r, AP_a, \nu_a), \text{ where } \nu_a(s) = \begin{cases} \{cs_0\}, & \text{if } s \in \{s_3, s_6\} \\ \{cs_1\}, & \text{if } s \in \{s_5, s_7\} \\ \emptyset, & \text{otherwise} \end{cases}$$

$$\mathcal{K}_b = (S, \rightarrow, r, AP_b, \nu_b), \text{ where } \nu_b(s) = \begin{cases} \{w_0\}, & \text{if } s \in \{s_1, s_4, s_7\} \\ \{cs_0\}, & \text{if } s \in \{s_3, s_6\} \\ \emptyset, & \text{otherwise} \end{cases}$$

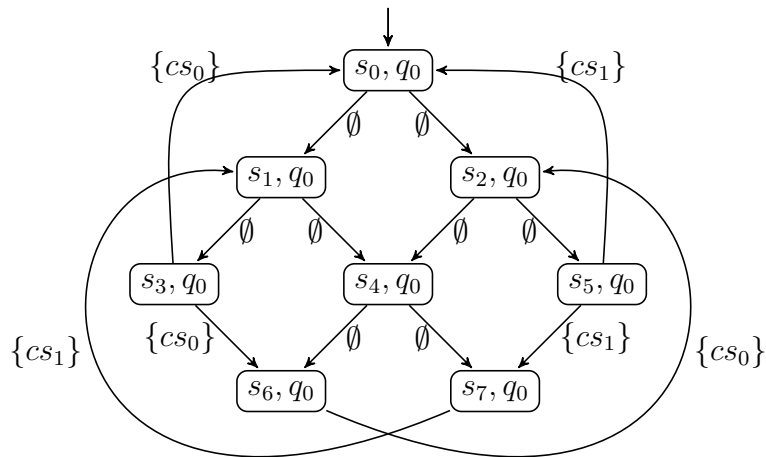
The Büchi automaton $\mathcal{B}_{\mathcal{K}_a} = (2^{AP_a}, S, r, \Delta_{\mathcal{K}_a}, S)$, where $\Delta_{\mathcal{K}_a} = \{(s, \nu_a(s), t) \mid s \rightarrow t\}$:



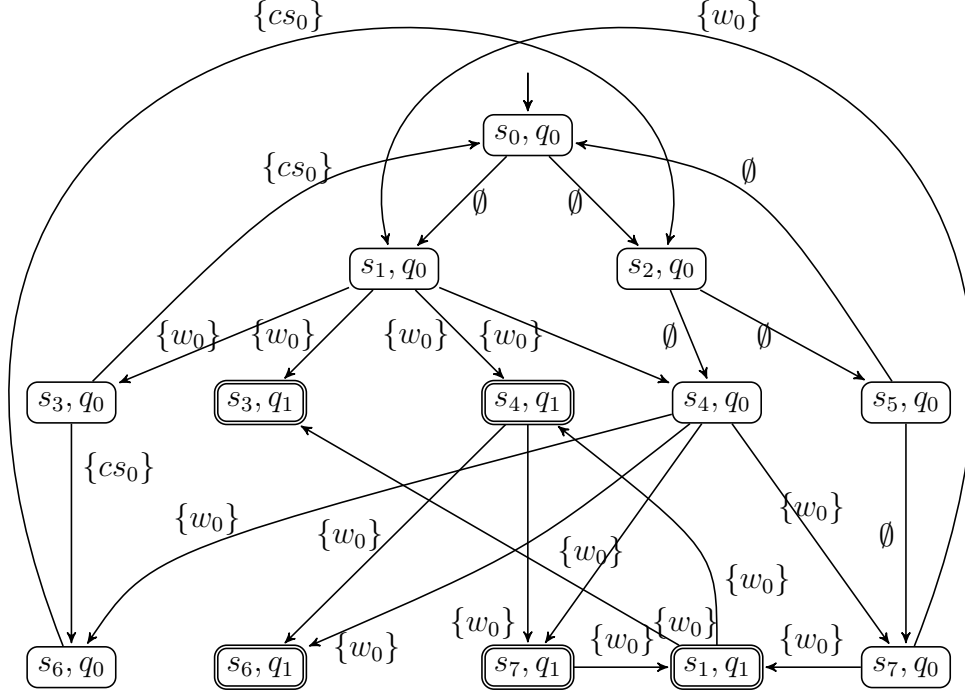
The Büchi automaton $\mathcal{B}_{K_b} = (2^{AP_b}, S, r, \Delta_{K_b}, S)$, where $\Delta_{K_b} = \{(s, \nu_b(s), t) \mid s \rightarrow t\}$:



(v) $\mathcal{B}_a = (2^{AP_a}, S \times \{q_0, q_1\}, (s_0, q_0), \Delta_a, S \times \{q_1\})$



$$\mathcal{B}_b = (2^{AP_b}, S \times \{q_0, q_1\}, (s_0, q_0), \Delta_b, S \times \{q_1\})$$



- (vi) For \mathcal{B}_a , the algorithm terminates without reporting a counterexample. By the time the algorithm terminates, it finds out that every state forms an SCC, but without an accepting state.

For \mathcal{B}_b , assuming that the algorithm always searches the automaton above top-to-bottom and left-to-right, then it first finds the SCC $\{(s_0, q_0), (s_1, q_0), (s_3, q_0)\}$ when it reaches (s_3, q_0) . Later, it finds the SCC $\{(s_6, q_0), (s_2, q_0), (s_4, q_0)\}$ when it reaches (s_4, q_0) . Finally, it reaches (s_4, q_1) and finds out that its successor (s_7, q_1) is still active. Since (s_4, q_1) is an accepting state, the algorithm stops. Notice that the last SCC it discovers is $\{(s_7, q_1), (s_1, q_1), (s_4, q_1)\}$ with (s_7, q_1) as the root.

The counterexample run in \mathcal{B}_b found by the algorithm is

$$(s_0, q_0), (s_1, q_0), (s_3, q_0), (s_6, q_0), (s_2, q_0), (s_4, q_0), ((s_7, q_1), (s_1, q_1), (s_4, q_1))^\omega$$

The corresponding path in \mathcal{K}_b can be obtained by projecting the run on the first component. The counterexample valuation sequence is as follows:

$$\{\}\{w_0\}\{cs_0\}\{cs_0\}\{\}\{w_0\}(\{w_0\}\{w_0\}\{w_0\})^\omega$$

- (vii) Check the formulas: $!\langle \rangle(m[0]@cs \ \&\& \ m[1]@cs)$ and $[\](m@wait \ \rightarrow \ (\langle \rangle \ m@cs))$