SYS-6581 Transition Systems and Linear Temporal Logic

Alan Wang

December 16, 2018

6 Transition Systems and Linear Temporal Logic

6.1 Problem 1

Assuming question is asking for states transitioned to in a valid path:

(a) Next is a: $\{s_1, s_2, s_3, s_4\}$

(b) Next, next, next proposition is a: $\{s_1, s_2, s_3, s_4\}$

(c) Always b. No states fits this requirement: $\{\}$

(d) Always eventually a: $\{s_1, s_2, s_3, s_4\}$

(e) Always b until a: $\{\}$

(f) Eventually a until b. Since starting from s_4 would make the trace invalid with starting $\{b\}$, the answer is: $\{s_1, s_2, s_3\}$

6.2 Problem 2

(a) Eventually always c. **False**. Trace: $a, (b, c)^+$

(b) Always eventually c. **True**. Trace: $a, (b, c)^+$

(c) Next not c implies next, next is c. **True**. Next not c implies s_4, and next, next c equivalent to neighboring states of s_4, which always has c.

(d) Always a. **False**. Example: $a, (b, c)^+$

(e) a until always (b or c). **False**. Example trace: $c, (b, \{b, c\})^+$

(f) Next, next b until (b or c). **True**.

6.3 Problem 3

(a) Only one user can use the printer at a time:

$$\Box\neg(Peter.use \wedge Betsy.use)$$

(b) Users can only print for a finite amount of time; or, users cannot print for an infinite amount of time:

$$\neg\Box Peter.use \wedge \neg\Box Betsy.use$$

(c) If a user wants to print something, he is eventually able to do so:

$$(Peter.request \to \Diamond Peter.use) \wedge (Betsy.request \to \Diamond Betsy.use)$$

(d) User can always request to use the printer:

$$\Box\Diamond(Peter.request \vee Betsy.request)$$

(e) Assuming that there could be times when neither people use the printer, for either user their use implies that the printer will remain unused until the end of time or be used by the other person:

$$\begin{aligned}&\Box(Peter.release \to \times(\neg Peter.request \vee Betsy.request)\\ &\quad \wedge Betsy.release \to \times(\neg Betsy.request \vee Peter.request))\end{aligned} \tag{1}$$

6.4 Problem 4

Let N= 4, and n= $\{0, 1, 2, 3\}$:

(a) $\Box\neg(Door_n.open \wedge Elevator_n)$

(b) $\Box\Diamond Request_n \to (Elevator_n \wedge Door_n \wedge Indicator_n)$

(c) $\Box\Diamond Elevator_0$

(d) $Request_{N-1} \to \times(Elevator_{N-1} \wedge Door_{N-1} \wedge Indicator_{N-1})$

6.5 Problem 5

(a) **True**. $\varphi \bigcup \psi$, since ψ or $\neg\phi$ is equivalent to ψ, and $\Box\phi \to \Diamond\psi$

(b) **False**. Simplified: $\Box(\psi \bigcup \phi)$ not equivalent to $\Box(\Diamond True)$

(c) **True**. The equivalence seems to be, for the right side, you can never have ψ without a φ. For the left side, you always have ψ until not ϕ. The total two equivalent states for both equations are:

$$\begin{gathered}\psi\\ \psi, \phi+\\ phi...\psi, \phi+\end{gathered} \tag{2}$$

(d) **False** by non-identity. Counter-Example: $a+$