Universität
des
Saarlandes
FR Informatik

MPI INFORMATIK

Harald Ganzinger
Uwe Waldmann

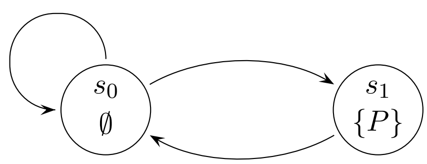
July 12, 2002

Tutorials for "Logic in Computer Science"
Sample solution for exercise sheet 12

Exercise 12.1:

(a) Give an example of a finite time structure $M = (S, R, L)$ and a state $s_0 \in S$ such that $M, s_0 \models \mathsf{AG}(\mathsf{EF}\, P)$, but $M, s_0 \not\models \mathsf{AF}\, P$.

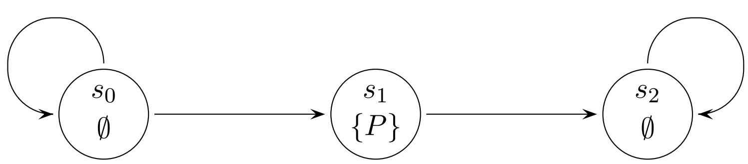
Solution:



For every path starting from s_0 and for every state on that path, there exists a path from that state on which P is sometimes true. However, it is not true that on every path starting from s_0, P is sometimes true (there exists a path that remains in s_0 all the time).

(b) Give an example of a finite time structure $M = (S, R, L)$ and a state $s_0 \in S$ such that $M, s_0 \models \mathsf{EG}(\mathsf{EF}P)$, but $M, s_0 \not\models \mathsf{EGF}P$.

Solution:



There exists a path (namely the one that remains in s_0 all the time), such that from each state on that path there exists a path on which P is sometimes true. However, it is not true that there exists a path, such that for each state on this path there exists some later state on this path in which P is true.

Exercise 12.2:

Express the following statements in CTL:

(a) It is possible that the CD drive of a computer gets into such a state that opening it becomes impossible forever.

Solution:

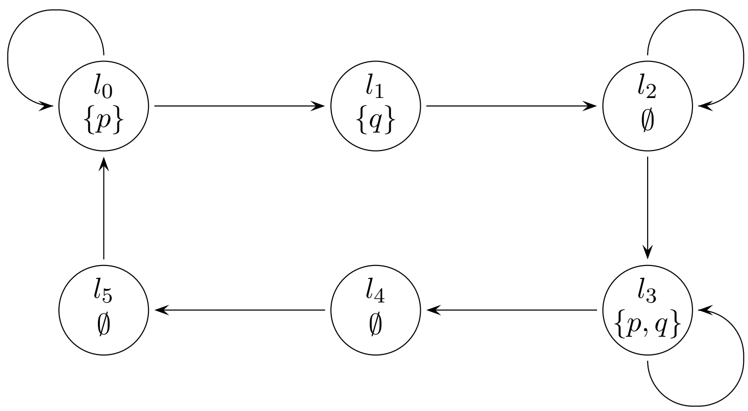
$\mathsf{EF}(\mathsf{AG}\neg open)$

(b) A student cannot take the electrical engineering exam more than two times.

Solution:

$\mathsf{AG}(ee\_exam \rightarrow \mathsf{AXAG}(ee\_exam \rightarrow \mathsf{AXAG}\neg ee\_exam))$

Exercise 12.3:
Let $S = \{l_0, l_1, l_2, l_3, l_4, l_5\}$, let $\Pi = \{p, q\}$, and let $M = (S, R, L)$ be the following time structure (where R is represented by $\rightarrow$):



Compute $[\![\mathsf{AX}\,\mathsf{AX}\,\mathsf{E}((\neg q)\mathsf{U}p)]\!]$.

Solution:

$$
\begin{aligned}
[\![p]\!] &= \{l_0, l_3\} \\
[\![q]\!] &= \{l_1, l_3\} \\
[\![\neg q]\!] &= S \setminus [\![q]\!] \\
&= \{l_0, l_2, l_4, l_5\} \\
[\![\mathsf{E}((\neg q)\mathsf{U}p)]\!] &= \mu\, Z.\, [\![p]\!] \cup ([\![\neg q]\!] \cap R^{-1}(Z)) \\
&= \{l_0, l_2, l_3, l_4, l_5\} \\
&\quad \big[\ \tau^0(\emptyset) = \emptyset \\
&\quad\ \ \tau^1(\emptyset) = \{l_0, l_3\} \\
&\quad\ \ \tau^2(\emptyset) = \{l_0, l_3, l_2, l_5\} \\
&\quad\ \ \tau^3(\emptyset) = \{l_0, l_3, l_2, l_5, l_4\} \\
&\quad\ \ \tau^4(\emptyset) = \{l_0, l_3, l_2, l_5, l_4\} = \tau^3(\emptyset)\ \big]
\end{aligned}
$$

$$\begin{aligned}
[\![\mathsf{AX}\,\mathsf{E}((\neg q)\mathsf{U}p)]\!] &= S \setminus (R^{-1}(S \setminus [\![\mathsf{E}((\neg q)\mathsf{U}p)]\!])) \\
&= \{l_1, l_2, l_3, l_4, l_5\} \\
[\![\mathsf{AX}\,\mathsf{AX}\,\mathsf{E}((\neg q)\mathsf{U}p)]\!] &= S \setminus (R^{-1}(S \setminus [\![\mathsf{AX}\,\mathsf{E}((\neg q)\mathsf{U}p)]\!])) \\
&= \{l_1, l_2, l_3, l_4\}
\end{aligned}$$

Exercise 12.4:
The syntax of CTL given on slide 20 permits state formulas like $\mathsf{EAX}\,P$ (for $P \in \Pi$). The simplified definition given on slide 23 and 34 does not permit formulas of this kind. Why is this difference semantically irrelevant?

Solution:

$\mathsf{EAX}\,P$ is equivalent to $\mathsf{AX}\,P$. And more generally: whenever a state formula is allowed by the definition on slide 20 but excluded by the simplified definition on slide 23, then it contains two or more path quantifiers (E, A) following directly after another. In this case, all but the last quantifier can be deleted without changing the semantics of the formula. The resulting state formula is allowed by the simplified definition on slide 23.

Exercise 12.5:
Give an example of a Boolean function f with three variables x, y, z, such that the minimal OBDD for f has 5 interior nodes for the variable ordering $x < y < z$ and 4 interior nodes for some other variable ordering.

Solution:

The Boolean function $f(x, y, z) = y \vee (x \wedge \neg z) \vee (\neg x \wedge z)$ has an OBDD with 5 interior nodes for the variable ordering $x < y < z$ (i. e., x is the topmost node), and an OBDD with 4 interior nodes for the variable ordering $y < x < z$.

Exercise 12.6:
Let p be an arbitrary mixed CTL formula such that Z is the only explicit set of states occurring in p. Is the function $\tau : Z \mapsto p$ necessarily monotone?

Solution:

No, if Z occurs in p below a negation sign or in the left argument of an implication, then τ need not be monotone. For instance, the function $\tau : Z \mapsto \neg Z$ that maps each set of states to its complement is not monotone.