Principles of Model Checking
Solutions to exercise class 4

Computation tree logic

Prof. Dr. Joost-Pieter Katoen, Dr. Taolue Chen,
Dr. Jeroen Ketema, and Ir. Mark Timmer

October, 5, 2012

Problem 1

1. To say that there exists a path on which eventually a holds and in the next state after that, $\neg a$ holds, we could not simply write $\exists\Diamond(a \wedge \bigcirc\neg a)$. After all, $\bigcirc\neg a$ is a path formula, while the $\wedge$ operator expects two state formulas. Hence, we need to prefix $\bigcirc\neg a$ by either $\exists$ or $\forall$. Since there only has to be one path such that at some point a and then $\neg a$, we use $\exists$ and get $\exists\Diamond(a \wedge \exists\bigcirc \neg a)$.

   Since we need to express the property that says that there exists a path on which for every state s the above holds, we obtain

$$\exists\Box\exists\Diamond(a \wedge \exists\bigcirc \neg a)$$

2. We can express that c holds as long as b does not hold by the formula $c\mathsf{W}b$. Note that $c\mathsf{U}b$ would require b to eventually hold; this is something stronger than what we want.

   To say that a is true and all paths satisfy the above, we easily write $a \wedge \forall(c\mathsf{W}b)$. Finally, since we only need one state in which this holds, we can existentially range over all paths and require the above to eventually hold in some state:

$$\exists\Diamond(a \wedge \forall(c\mathsf{W}b))$$

Problem 2

For each of the CTL state formulas Φ_i, we have to compute

$$Sat(\Phi_i) = \{s \in S \mid s \models \Phi_i\}$$

From this, we can decide $TS \models \Phi_i$ by checking $I \subseteq Sat(\Phi_i)$.

- $\Phi_1 = \forall(a \mathsf{U} b) \vee \exists\bigcirc(\forall\Box b)$

  We follow the bottom-up construction of the satisfaction sets:

  * $Sat(b) = \{s_2, s_3, s_4\}$
  * $Sat(\forall\Box b) = \{s_4\}$
  * $Sat(\exists\bigcirc(\forall\Box b)) = \{s_0, s_4\}$
  * $Sat(a) = \{s_1, s_2\}$
  * $Sat(\forall(a \mathsf{U} b)) = \{s_1, s_2, s_3, s_4\}$
  * $Sat(\forall(a\mathsf{U}b) \vee \exists\bigcirc(\forall\Box b)) = \{s_1, s_2, s_3, s_4\} \cup \{s_0, s_4\} = \{s_0, s_1, s_2, s_3, s_4\}$

  Since all initial states are in $Sat(\Phi_1)$, indeed $TS \models \Phi_i$.

  Alternatively, we could for instance argue directly that $s_0 \models \Phi_1$ since it has a path $\pi = s_0 s_4^\omega$ that satisfies $\bigcirc(\forall\Box b)$, and that $s_3 \models \Phi_1$, since all paths from s_3 start with b and hence satisfy $(a \mathsf{U} b)$.

- $\Phi_2 = \forall\Box\forall(a \mathsf{U} b)$

  First note that

$$\begin{aligned} s \models \Phi_2 &\iff \forall \pi \in Paths(s).\ \pi \models \Box\forall(a \mathsf{U} b) \\ &\iff \forall \pi \in Paths(s).\ \forall i \geq 0.\ \pi[i] \models \forall(a \mathsf{U} b) \\ &\iff \forall \pi \in Paths(s).\ \forall i \geq 0.\ \forall \pi' \in Paths(\pi[i]).\ \pi' \models a \mathsf{U} b \end{aligned}$$

  We consider the state s_0 and the path $\pi'' = s_0 s_4^\omega$. According to the equivalence above, for $s_0 \models \Phi_2$ to hold, all the suffixes of π'' should satisfy $a \mathsf{U} b$. Choose $i = 0$, and take $\pi' = \pi''$. Clearly, $\pi' \not\models a \mathsf{U} b$, and therefore $s_0 \not\models \Phi_2$. So, $s_0 \notin Sat(\Phi_2)$.

  Since we do have $s_0 \in I$, we find that $I \not\subseteq Sat(\Phi_2)$, and thus that $TS \not\models \Phi_2$.

  (As all states except for s_4 can reach s_0, it can be seen that they are also not in $Sat(\Phi_2)$. Hence, $Sat(\Phi_2) = \{s_4\}$.)

Problem 3

We prove that there is no equivalent LTL-formula for the CTL-formula

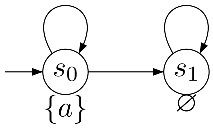
$$\Phi = \forall\Diamond(a \wedge \exists\bigcirc a)$$

We omit all path quantifiers from Φ, obtaining the LTL-formula

$$\varphi = \Diamond(a \wedge \bigcirc a)$$

We now prove that $\Phi \not\equiv \varphi$. Then, by Theorem 6.18 indeed there is no equivalent LTL-formula for Φ.

To see that $\Phi \not\equiv \varphi$, consider the following transition system TS:



We have $TS \not\models_{\text{LTL}} \varphi$, due to the path $s_0 s_1^\omega$. It never sees two consecutive a-states.

On the other hand, we do have $TS \models_{\text{CTL}} \Phi$. After all, there are only two types of paths to consider: s_0^ω and $s_0^+ s_1^\omega$. Both eventually reach a state such that $a \wedge \exists\bigcirc a$, namely s_0. To see why s_0 satisfies $a \wedge \exists\bigcirc a$, note that indeed $a \in L(s_0)$ and there is a path $s_0 s_1^\omega$ from s_0 that satisfies $\bigcirc a$.

Since $TS \not\models_{\text{LTL}} \varphi$ and $TS \models_{\text{CTL}} \Phi$, indeed $\Phi \not\equiv \varphi$.

Problem 4

(a) Determine $Sat(\Phi_1)$ and $Sat(\Psi_1)$ (without fairness).

We compute $Sat(\Phi_1) = Sat(b \wedge \neg a)$ using Algorithm 14 on page 348, i.e., by recursion on the subformulas. We thus first obtain

$$Sat(a) = \{s_0, s_5\} \qquad Sat(b) = \{s_0, s_2, s_3\}$$

Next, applying the rule for negation, we obtain

$$Sat(\neg a) = S \setminus Sat(a) = S \setminus \{s_0, s_5\} = \{s_1, s_2, s_3, s_4\}$$

Finally,

$$\begin{aligned} Sat(\Phi_1) = Sat(b \wedge \neg a) &= Sat(b) \cap Sat(\neg a) \\ &= \{s_0, s_2, s_3\} \cap \{s_1, s_2, s_3, s_4\} = \{s_2, s_3\} \end{aligned}$$

Next, we compute $\mathit{Sat}(\Psi_1) = \mathit{Sat}(\exists(b \, \mathsf{U} \, (a \wedge \neg b)))$. First, using $\mathit{Sat}(a)$ and $\mathit{Sat}(b)$ from above, we find

$$\mathit{Sat}(\neg b) = S \setminus \mathit{Sat}(b) = S \setminus \{s_0, s_2, s_3\} = \{s_1, s_4, s_5\}$$
$$\mathit{Sat}(a \wedge \neg b) = \mathit{Sat}(a) \cap \mathit{Sat}(\neg b) = \{s_0, s_5\} \cap \{s_1, s_4, s_5\} = \{s_5\}$$

Finally, using a smallest fixed point computation, we obtain

$$\mathit{Sat}(\Psi_1) = \mathit{Sat}(\exists(b \, \mathsf{U} \, (a \wedge \neg b))) = \{s_0, s_2, s_5\}$$

(b) Determine $\mathit{Sat}_{\mathit{sfair}}(\exists\Box \, \mathrm{true})$.

To compute $\mathit{Sat}_{\mathit{sfair}}(\exists\Box \, \mathrm{true})$, we need to establish for each state $s \in S$ if there is a fair path starting from s. By Lemma 6.40, this means we need to check if there is a cycle through s that either visits no states from $\mathit{Sat}(\Phi_1)$ or visits at least one state from $\mathit{Sat}(\Psi_1)$.

First note that cycles through s_3 always visit s_3 and possibly also s_4. Since s_3 is in $\mathit{Sat}(\Phi_1)$ and neither s_3 nor s_4 is in $\mathit{Sat}(\Psi_1)$, such cycles cannot be fair. A similar argument can be given for cycles through s_4. Hence, s_3 and s_4 have no outgoing fair paths, and thus they are not in $\mathit{Sat}_{\mathit{sfair}}(\exists\Box \, \mathrm{true})$.

From s_5 there is a cycle $(s_5 s_2)^\omega$, which contains at least one state from $\mathit{Sat}(\Psi_1)$. From s_2, we can use the cycle $(s_2 s_5)^\omega$. From s_1, the cycle $(s_1 s_0)^\omega$ contains a state from $\mathit{Sat}(\Psi_1)$, and from s_0 we can use the cycle $(s_0 s_1)^\omega$. Hence, all these states have a fair path, and thus are in $\mathit{Sat}_{\mathit{sfair}}(\exists\Box \, \mathrm{true})$

In conclusion, $\mathit{Sat}_{\mathit{sfair}}(\exists\Box \, \mathrm{true}) = \{s_0, s_1, s_2, s_5\}$.

(c) Determine $\mathit{Sat}_{\mathit{sfair}}(\Phi)$.

To compute $\mathit{Sat}_{\mathit{sfair}}(\Phi) = \mathit{Sat}_{\mathit{sfair}}(\forall\Box\forall\Diamond a)$ using Algorithm 17 on page 364, first note that

$$\forall\Box\forall\Diamond a \equiv \neg\exists\Diamond(\neg\forall\Diamond a) \equiv \neg\exists(\mathrm{true} \, \mathsf{U} \, \neg\forall\Diamond a) \equiv \neg\exists(\mathrm{true} \, \mathsf{U} \, \exists\Box\neg a)\,.$$

We begin by computing $\mathit{Sat}_{\mathit{sfair}}(\exists\Box\neg a)$. To this end we first have to find the strongly connected components in $G[\neg a]$ which realize *sfair*. Note that the component consisting of s_3 and s_4 is the only strongly connected component in $G[\neg a]$. However, this component does not realize *sfair*, as we saw above. Hence, $\mathit{Sat}_{\mathit{sfair}}(\exists\Box\neg a) = \varnothing$ and thus also $\mathit{Sat}_{\mathit{sfair}}(\exists(\mathrm{true} \, \mathsf{U} \, \exists\Box\neg a)) = \varnothing$. Hence, we can conclude that

$$\mathit{Sat}_{\mathit{sfair}}(\Phi) = \mathit{Sat}_{\mathit{sfair}}(\neg\exists(\mathrm{true} \, \mathsf{U} \, \exists\Box\neg a)) = S$$