CSC 410F: Assignment 6

Due on December 4, 2017

Harman Sran **Ryan Downes** **Michael Li** **Steve Lu**

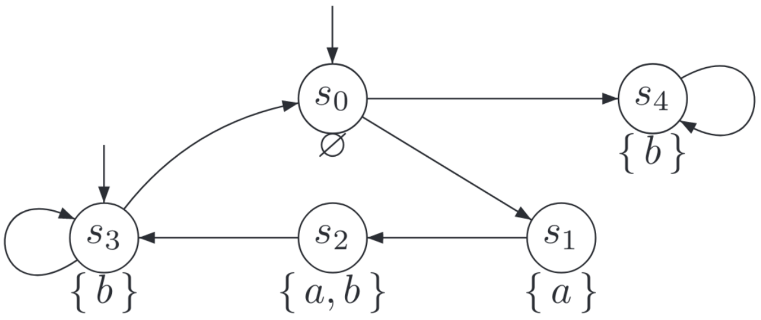
December 4, 2017

Problem 1

The following problems are related to CTL model checking algorithms.

(a)

Consider the transition system TS outlined below:



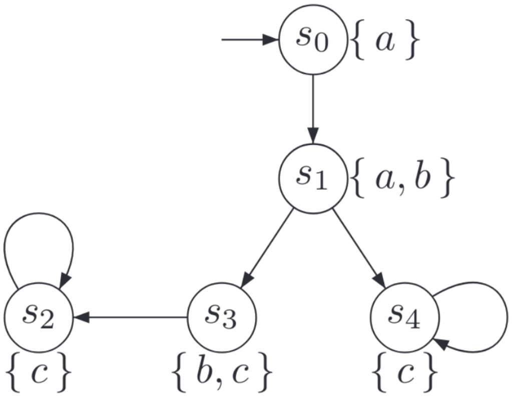
For each of the following CTL formulas, determine the set $Sat(\Phi_i)$ and whether $TS \models \Phi_i$.

- $\Phi_1 = \forall(a \cup b) \vee \exists \bigcirc (\forall\Box b)$
  - $Sat(\Phi_1) = \{s_0, s_1, s_2, s_3, s_4\}$
  - $TS \models \Phi_1$; since $s_0 \in Sat(\Phi_1) \wedge s_3 \in Sat(\Phi_1)$
- $\Phi_2 = \forall\Box\forall(a \cup b)$
  - $Sat(\Phi_2) = \{s_4\}$
  - $TS \not\models \Phi_2$; since $s_0 \notin Sat(\Phi_2) \wedge s_3 \notin Sat(\Phi_2)$
- $\Phi_2 = \forall\Box\exists\Diamond\Phi_1$
  - $Sat(\Phi_3) = \{s_0, s_1, s_2, s_3, s_4\}$
  - $TS \models \Phi_3$; since $s_0 \in Sat(\Phi_3) \wedge s_3 \in Sat(\Phi_3)$

Problem 1 continued on next page. . .

(b)

Consider the two CTL formulas $\Phi_1 = \exists\lozenge\forall\Box c$ and $\Phi_2 = \forall(a \cup \forall\lozenge c)$ and the transition system TS outlined below:



Decide whether $TS \models \Phi_i$ (for $i = 1, 2$); sketch the main steps (sub formulas and Sat sets).

i) We claim that $TS \models \Phi_1 = \exists\lozenge\forall\Box c$

Proof. We begin by converting to Existential Normal Form:

$$\begin{aligned} \Phi_1 &= \exists\lozenge\forall\Box c \qquad (1)\\ &= \exists\lozenge(\neg\neg(\forall\Box(\neg\neg c))) \qquad (2)\\ &= \exists\lozenge(\neg(\neg\forall\Box\neg(\neg c))) \qquad (3)\\ &= \exists\lozenge(\neg\exists\lozenge(\neg c)) \qquad (4) \end{aligned}$$

From this we derive the sub formulas:

$$\begin{aligned} \Phi_1' &= \exists\lozenge(\neg c) \qquad (5)\\ \Phi_1'' &= \exists\lozenge(\neg\Phi_1') \qquad (6) \end{aligned}$$

We use the sub formulas to compute the intermediate and final Sat sets:

$$\begin{aligned} Sat(\Phi_1') &= \{s_0, s_1\} \qquad (7)\\ Sat(\Phi_1) = Sat(\Phi_1'') &= \{s_0, s_1, s_2, s_3, s_4\} \qquad (8) \end{aligned}$$

Then we have $TS \models \Phi_1$; since $s_0 \in Sat(\Phi_1)$. □

ii) We claim that $TS \models \Phi_2 = \forall(a \cup \forall\Diamond c)$

Proof. We begin by converting to Existential Normal Form:

$$\begin{aligned}
\Phi_2 &= \forall(a \cup (\forall\Diamond c)) && (9)\\
&= \forall(a \cup (\neg\exists\Box\neg c)) && (10)\\
&= \neg\exists(\neg\neg\exists\Box\neg c \cup (\neg a \wedge \neg\neg\exists\Box\neg c)) \wedge \neg\exists\Box\neg\neg\exists\Box\neg c && (11)\\
&= \neg\exists((\exists\Box\neg c) \cup ((\neg a) \wedge (\exists\Box\neg c))) \wedge \neg\exists\Box\exists\Box\neg c && (12)
\end{aligned}$$

From this we derive the sub formulas:

$$\begin{aligned}
\Phi_2' &= \exists\Box\neg c && (13)\\
\Phi_2'' &= \neg\exists\Box(\Phi_2') && (14)\\
\Phi_2''' &= \neg a && (15)\\
\Phi_2'''' &= \neg\exists(\Phi_2' \cup (\Phi_2''' \wedge \Phi_2')) && (16)
\end{aligned}$$

For each sub formula, we compute the Sat set:

$$\begin{aligned}
Sat(\Phi_2') &= \emptyset && (17)\\
Sat(\Phi_2'') &= \{s_0, s_1, s_2, s_3, s_4\} && (18)\\
Sat(\Phi_2''') &= \{s_2, s_3, s_4\} && (19)\\
Sat(\Phi_2'''') &= \{s_0, s_1, s_2, s_3, s_4\} && (20)\\
& && (21)
\end{aligned}$$

Finally we compute the Sat set for $\Phi_2 = \Phi_2'''' \wedge \Phi_2''$:

$$\begin{aligned}
Sat(\Phi_2) &= Sat(\Phi_2'''') \cap Sat(\Phi_2'') && (22)\\
&= \{s_0, s_1, s_2, s_3, s_4\} && (23)
\end{aligned}$$

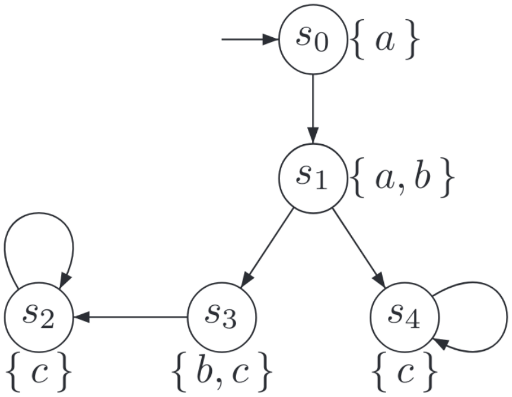
Then we have $TS \models \Phi_2$; since $s_0 \in Sat(\Phi_2)$. □

Problem 2

Let TS be a finite transition system (over AP) without terminal states, and Φ and Ψ be CTL state formula (over AP).

Prove or disprove $TS \models \exists(\Phi \cup \Psi) \iff TS' \models \exists\Diamond\Psi$ where TS' is TS with all outgoing transitions from states s such that $s \models \Psi \vee \neg\Phi$ removed.

We will provide a counter example. let $\Phi = a$ an let $\Psi = \exists \bigcirc b$ then in the below TS we have $TS \models \exists(\Phi \cup \Psi)$



Namely any path from the state S_0 models $\exists\bigcirc b$. After eliminating all outgoing transitions from states S_0 we have the TS consisting of only S_0. In this new TS S_0 has no next node and hence no path from S_0 models $\exists\Diamond\exists \bigcirc b$ and so it is not the case that $TS \models \exists(\Phi \cup \Psi) \iff TS' \models \exists\Diamond\Psi$ where TS' is TS with all outgoing transitions from states s such that $s \models \Psi \vee \neg\Phi$ removed.

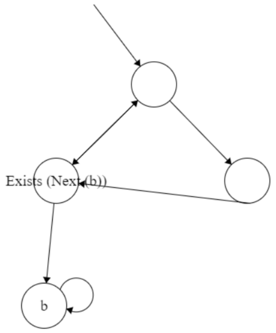
Problem 3

Prove the following equivalences (formally prove one) or provide counter examples.

(a)

$$\forall \bigcirc \forall \Diamond \Phi \equiv \forall \Diamond \forall \bigcirc \Phi$$

let $\Phi = \exists \bigcirc b$ then in the below TS we have $\forall \bigcirc \forall \Diamond \Phi$ but it is not the case that the below TS models $\forall \Diamond \forall \bigcirc \Phi$



Note empty nodes mean no atomic statements are true and in one empty node we have written $\exists \bigcirc b$ for clarification.

(b)

$$\exists \bigcirc \exists \Diamond \Phi \equiv \exists \Diamond \exists \bigcirc \Phi$$

Assume $s \models \exists \bigcirc \exists \Diamond \Phi$ then

$$\exists \pi \in Paths(s).\pi[1] \models \bigcirc \exists \Diamond \Phi$$

hence we have that

$$\exists s'.\forall \pi' \in Paths(s')$$

so that we have

$$\pi' = \pi[1] \in Paths(s) \wedge s' \models \exists \Diamond \Phi$$

so there is some

$$\pi[1] \in Paths(s).\pi' \models \Diamond \Phi$$

thus there is some j such that

$$j \geq 0.\pi'[j] \models \Phi$$

thus there is some j + 1 such that

$$.\pi[j+1] \models \Phi$$

so there is some

$$\pi[j] \in Paths(s).\pi[j] \models \bigcirc \Phi$$

hence we have that

$$\exists s''.\forall \pi'' \in Paths(s)$$

such that

$$\pi'' = \pi[j] \in Paths(s)$$

hence we have that

$$s'' \models \exists \bigcirc \Phi$$

so there is some path π''' in s such that

$$\pi''' \models \Diamond \exists \bigcirc \Phi$$

then

$$\exists \pi \in Paths(s).\pi \models \Diamond \exists \bigcirc \Phi$$

hence

$$s \models \exists \Diamond \exists \bigcirc \Phi$$

thus there is some j + 1 such that

$$.\pi[j+1] \models \Phi$$

Problem 3 continued on next page. . .

(b)

Assume $s \models \exists\Diamond\exists\bigcirc\Phi$ then

$$\exists\pi \in Paths(s).\pi \models \bigcirc\exists\Diamond\Phi$$

so there is some path π in s such that

$$\pi \models \Diamond\exists\bigcirc\Phi$$

hence we have that

$$\exists s''.\forall\pi'' \in Paths(s)$$

we have that

$$\pi'' = \pi[j] \in Paths(s) \wedge s'' \models \exists\bigcirc\Phi$$

hence we have that

$$s'' \models \exists\bigcirc\Phi$$

so there is some

$$\pi[j] \in Paths(s).\pi[j] \models \bigcirc\Phi$$

thus there is some j + 1 such that

$$.\pi[j+1] \models \Phi$$

so there is some π such that

$$\pi[1] \models \Diamond\Phi$$

hence we have that

$$\exists s'.\forall\pi' \in Paths(s')$$

so that we have

$$\pi' = \pi[1] \in Paths(s)$$

$$s' \models \exists\Diamond\Phi$$

then

$$\exists\pi \in Paths(s).\pi[1] \models \bigcirc\exists\Diamond\Phi$$

thus showing that $s \models \exists\bigcirc\exists\Diamond\Phi$ this completes the proof.

(c)

$$\forall \bigcirc \forall \Box \Phi \equiv \forall \Box \forall \bigcirc \Phi$$

Let's consider the statement $\forall \bigcirc \forall \Box \Phi$ we know that it is equivalent to

$$\forall \bigcirc \forall \Box \neg\neg\Phi$$

which is the same as

$$\neg\exists \bigcirc \exists \Diamond \neg\Phi$$

by part b we have

$$\neg\Diamond \bigcirc \exists \bigcirc \neg\Phi$$

which is equivalent to

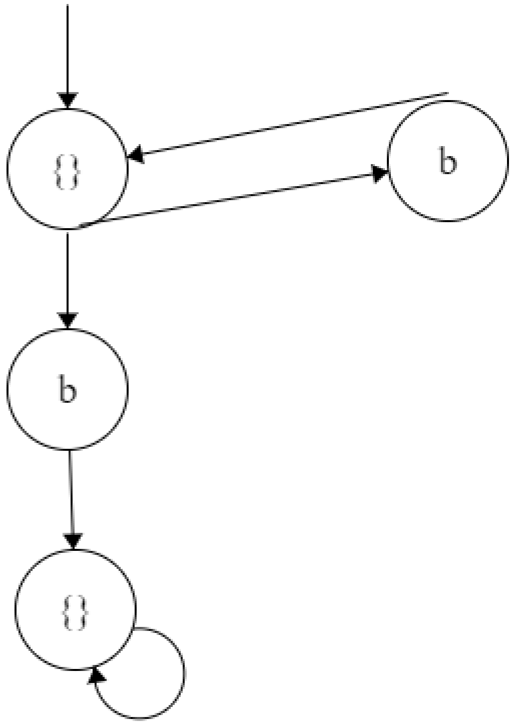
$$\forall \Box \forall \bigcirc \neg\neg\Phi$$

and hence is true if the statement in b is true since we have proven b c follows.

(d)

$$\exists \bigcirc \exists \Box \Phi \equiv \exists \Box \exists \bigcirc \Phi$$

Let $Phi = b$. Then in the below TS $\Box \exists \bigcirc b$ but it is not the case that $\exists \bigcirc \exists \Box b$.



Problem 4

(a)

Since the program will return in that case that any if statement becomes true we will make a linear path and on each node of the symbolic execution tree will note the feasibility of the statement being true or the statement being not true.
13.

$$a \mapsto A, b \mapsto B, c \mapsto C$$

Transition from 13 to 15

$$A > 0, B > 0, C > 0$$

in addition $A <= 0, B <= 0, C <= 0$ is a feasible path leading to the execution of line 16

Translation from 15 to 18

$$A! = B \land B! = C$$

in addition $A == 0 \lor B == C$ is a feasible path leading to the execution of line 19

Translation from 18 to 21

$$A < (B + C) \land C < (B + A) \land B < (A + C)$$

in addition

$$A >= (B + C) \lor C >= (B + A) \lor B >= (A + C)$$

is a feasible path leading to the execution of line 22
Translation from 21 to 24

$$A! = B \lor B == C \land A == B \lor C! = A \land C! = B \lor C == A$$

in addition

$$A == B \land B! = C \lor A! = B \land C == A \lor C == B \land C! = A$$

is a feasible path leading to the execution of line 22
lastly the execution of line 27 is feasible

Problem 4 continued on next page...

(b)

If we begin with

$$a \mapsto 0, b \mapsto 0, c \mapsto 0$$

then we have the execution path 15,16.

The it is necessary to have all values be greater then 0 to ensure line 15 is not satisfied so let

$$a \mapsto 1, b \mapsto 1, c \mapsto 1$$

then the program will execute the lines 15, 18, 19.
To ensure the failure of the condition on lines 15 and 18 we need it to be the case that a,b,c are greater then 0 and at lease 1 is not the same as the others.The Assignment

$$a \mapsto 1, b \mapsto 1, c \mapsto 2$$

achieves this requirement. Preforming the program execution 15,18,21,22.
To achieve the execution to line 25 we can use the assignment

$$a \mapsto 3, b \mapsto 3, c \mapsto 5$$

ensures that the conditions on lines 15,18,21 and that the execution of the would be 15,18,21,24,25
For an execution of line 27 the assignment must not satisfy lines 15,18,21,24. If

$$a \mapsto 3, b \mapsto 4, c \mapsto 5$$

then the execution would be 15, 18,21,24,27.

(c)

Table 1: JPF Test cases for c symbolic

x	return value
-10	-1
1	-1
4	1
2	2
3	1
7	-1

Problem 4 [(c)] continued on next page…

The JPF test cases cover every possible branch of execution, which we can conclude by following the multiple inputs through the execution tree in a)

(d)

Table 2: JPF Test cases for a b c symbolic

a	b	c	return value
1	1	-10	-1
1	1	1	0
2	2	1	1
1	1	2	-1
1	2	1	-1
1	2	2	1
4	3	2	2
2	1	2	1
1	2	3	-1
2	1	1	-1

Table 3: Additional test cases for MCDC

a	b	c	return value
-10	1	1	-1
1	-10	1	-1

The JPF test cases with the additional test cases achieve MC/DC. This is because of complete branch coverage, and the fact each condition is shown to independently affect the decision. When testing the triangle inequality with $z \geq x+y$, JPF tests with an invalid z in each variable position a, b, and c. This also occurs when checking for a valid isosceles triangle $(x == y)$ && $(y \neq z)$ and our added test cases for invalid negative integers
$(x \leq 0 \,||\, y \leq 0 \,||\, z \leq 0)$