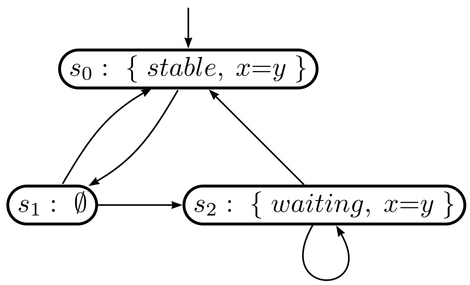
Exercises PV 09/10

Wishnu Prasetya

April 18, 2012

1 LTL Model Checking

1. Consider the Kripke structure K depicted below. The states are $\{ s_0, s_1, s_2 \}$, with s_0 as the innitial state. We use $Prop = \{ stable, waiting, x{=}y \}$. Which propositions hold (and otherwise) at each state can be seen below.



   (a) Consider the property ϕ given as: $\Box(waiting \rightarrow (waiting\ \mathbf{W}\ stable))$. What does it say?

   Answer:

   Whenever *waiting* holds, then either it holds forever, or at some point it stops to hold; but then *stable* has to hold.

   (b) What is its negation?

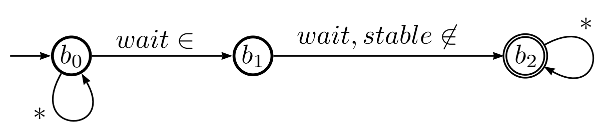
   Answer:

   It can be violated if at some point *waiting* holds, and at the next state both *waiting* and *stable* fails to hold. Formally:

   $$\Diamond(waiting \wedge \mathbf{X}\ \neg waiting \wedge \neg stable)$$

   (c) Give a Buchi automaton $A_\neg$ that represent this negation.

   Answer:

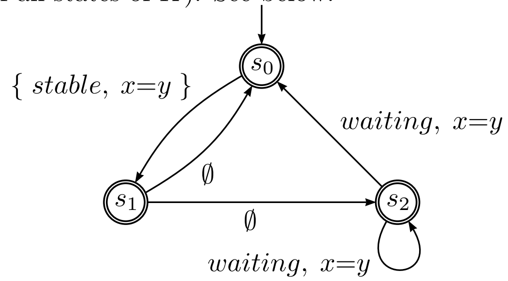


   Notice that this automaton is non-deterministic.

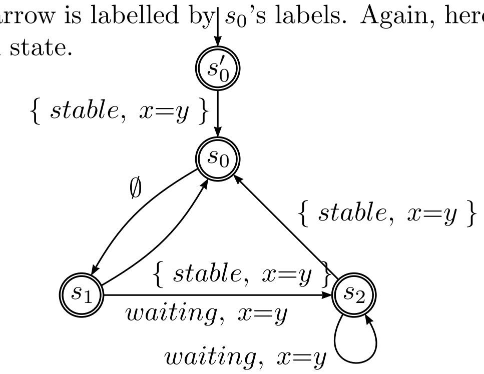
   (d) Construct the automaton $K \cap A_\neg$.

   Answer:

   We'll first convert the Kripke K to its Buchi equivalent. We will have to move the labels from the states to the arrows. We can do this by decorating an arrow $s \rightarrow t$ with the labels of s; assuming no state in K is terminal (we want it to be a non-terminating automaton anyway), the resulting automaton generates the same sentences. Furthermore all states will be made accepting (to be precise, a single accepting set, consisting of all states of K). See below:



Alternatively, we can label $s \to t$ with the labels of t, but we also need to add a dummy initial state $s_0' \to s_0$ that goes to K's original initial state; this arrow is labelled by s_0's labels. Again, here we assume K has no terminal state.



I will now use the first version. Let $\Sigma_K = \{ s_0, s_1, s_2 \}$ be K's set of states. Similarly let $\Sigma_n eg = \{ b_0, b_1, b_2 \}$ be $A_\neg$'s set of states.

Let's call $K \cap A_\neg$ the automaton I, because it is shorter.

The states of I will be drawn from $\Sigma_K \times \Sigma_\neg$.

I would start from (s_0, b_0); that is, the combined starting states of both K and $A_\neg$.

I In will contains only transitions that would be allowed by both K and $A_\neg$. So, there is a transition :

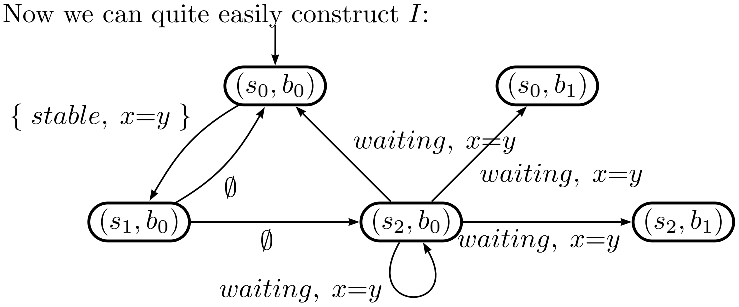
$$(s, b) \ \stackrel{A}{\to} \ (t, c)$$

only if $s \stackrel{A}{\to} t$ and $b \stackrel{A}{\to} c$.

Do keep in mind that the notation like $b \stackrel{p\in}{\to} c$ that we used in the picture of $A_\neg$ represents a family of arrows from b to c, each is labelled by a subset A of *Prop* such that $p \in A$.

The accepting states of I would be all those states (s, b) where s is accepting in (the Buchi version of) K *and* b is accepting in $A_\neg$. However, since all states of (Buchi) K, only b determines if (s, b) would be accepting.

Now we can quite easily construct I:



(e) So, does K satisfies the property ϕ?

Answer:

The set F of the accepting states of I contains all (s, b_3), for any $s \in \Sigma_K$. However, no such state is reachable in I as you can see above. Therefore, $L(I) = \emptyset$.

This implies that no counter example exists for ϕ. Thefore K does satisfy ϕ.

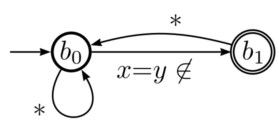
2. Verify whether in the K above eventually $x{=}y$ will remain to hold.

Answer:

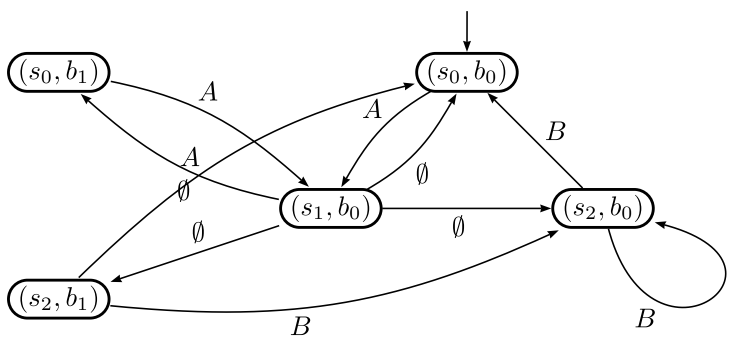
Well, formally what we want to verify is $\phi = \Diamond\Box(x = y)$.

Its negation is: $\Box\Diamond\neg(x = y)$. That is, if $\neg(x = y)$ holds infinitely many often.

The corresponding Buchi automaton $A_\neg$ is:



The automaton $J = K \cap A_\neg$ is then:



Where $A = \{\ stable,\ x{=}y\ \}$ and $B = \{\ waiting,\ x{=}y\ \}$.

The states $\{\ (s_0, b_1), (s_2, b_1)\ \}$ above are accepting.

So for example run $(s_0, b_0), (s_1, b_0), (s_0, b_1), (s_1, b_0), (s_0, b_1), \ldots$ is an accepting run. And therefore $L(J)$ cannot be empty. Furthermore, this run is also a counter example for the property $\Diamond\Box(x = y)$.

That is, if you project the run to the states of K:

$$s_0, s_1, s_0, s_1, \ldots$$

This runs violates $\Diamond\Box(x = y)$.

3. Verify whether the K from No. 1 satisfies the following properties:

(a) $\Box\Diamond(x = y)$

(b) $\neg waiting$ **U** $(waiting \wedge x{=}y)$