Exam-2 Program Verification 2019/2020

RUPPERT-C, 5th Nov. 2019, 17:00 - 20:00

Lecturer: Wishnu Prasetya

1. ... [1.5 pt].

2. **Theory** [1 pt].

   Let $M = (S, \{s_0\}, R, Prop, V)$ be a Kripke structure where S is a finite set of states, $s_0 \in S$ is M's initial state, $R : S \rightarrow \mathbf{Pow}(S)$ describes the transitions between the states, $Prop$ is a set of state predicates, and $V : S \rightarrow \mathbf{Pow}(Prop)$ describes the labelling of the states with members from $Prop$.

   Consider the following variation of LTL called $\text{LTL}^{\text{bounded}}$. Let ϕ and ψ represent $\text{LTL}^{\text{bounded}}$ formulas.

$$
\begin{array}{lll}
\phi & ::= & p \quad \text{, where } p \in Prop \text{ (so, } p \text{ is a state predicate)} \\
 & | & \neg\phi \\
 & | & \phi \wedge \psi \\
 & | & \mathbf{X}\phi \\
 & | & \phi\ \mathbf{U}^*\ \psi \\
 & | & \phi\ \mathbf{U}^{\leq k}\ \psi \quad \text{, where } k \text{ is a concrete non-negative integer}
\end{array}
$$

   - The meaning of $\phi\ \mathbf{U}^{\leq k}\ \psi$ is the same as $\phi\ \mathbf{U}\ \psi$ in the standard LTL, except that the future ψ should happen within at most k steps.
   - The meaning of $\phi\ \mathbf{U}^*\ \psi$ is the same as $\phi\ \mathbf{U}^{\leq k}\ \psi$ with $k = \infty$.
   - The meaning of other constructs is the same as in the standard LTL.

   Your task: give a formal definition of what $\sigma, i \models \phi$ means for all the above constructs of $\text{LTL}^{\text{bounded}}$.

3. **Buchi** [1.5 pt].

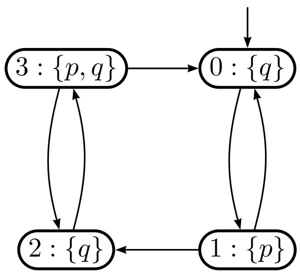
   Give for each LTL formula below, a Buchi automaton that equivalently describes the formula. Do not forget to specify what are the initial and accepting states are. Also, indicate whether you use a standard or generalized Buchi.

   Assume $Prop = \{p, q\}$.

   (a) $\mathbf{X}(p\ \mathbf{U}\ q)$
   (b) $\neg\mathbf{X}(p\ \mathbf{U}\ q)$
   (c) $(\Box\Diamond p) \wedge (\Box\Diamond(q \wedge \neg p))$

4. **LTL model checking** [1.5 pt].

   Consider the following Kripke structure K that you can think as modelling some program. $Prop = \{p, q\}$. The initial state is 0.



   Describe the steps of LTL model checking to verify whether the following LTL property ϕ holds on K:

   $$(p \vee q)\ \mathsf{U}\ (p \wedge q)$$

   Do **provide your resulting intersection automaton**.

5. **SPIN/Promela** [1 pt].

   Consider a Promela system consisting of **three** processes: `P(0)`, `P(1)`, and `P(2)`. They are defined below. Let's call the system of these three processes Sys.

```
#define NEXT(i) i+1
#define FREE 127
byte a[4] ;
byte lock[4] ;

proctype P(byte i) {
  byte tmp ;
  do
  :: { atomic { (lock[i]==FREE  && lock[NEXT(i)]==FREE) ;
                lock[i] = i ;
                lock[NEXT(i)] = i } ;
       if
       :: (a[i]>a[NEXT(i)]) ;
           /* swap a[i] and a[i+1]: */
           tmp = a[i] ; a[i] = a[NEXT(i)] ; a[NEXT(i)] = tmp
       :: else -> skip
       fi ;
       lock[i] = FREE ;
       lock[NETXT(i)] = FREE
     }
  od
}
```

We claim the system Sys as defined above will concurrently sort the array **a** ascendingly.

Questions:

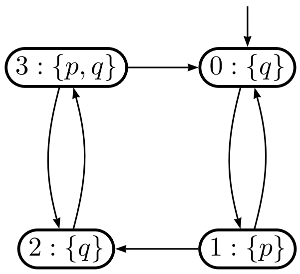
(a) Give an LTL specification that would fully capture the correctness of Sys.

(b) Consider the progress property $\Diamond(\mathtt{a[0]} \leq \mathtt{a[1]})$. You might expect this property to be valid on Sys, but unfortunately it is not, unless you insist on some fairness assumption. Please explain: what kind of fairness assumption would we need here?

(c) The model above is unrealistic because it allows each process `P(i)` to acquire multiple locks in a single atomic statement. Can you remedy this? You can assume that Promela statements of the form $(e_x); x{=}d_x$ can be implemented atomically, provided e_x and d_x are expressions that only mention x as variables.

6. **CTL model checking** [1.5 pt].

Consider again the Kripke structure K from question No 4. So:



Describe the steps of CTL model checking in order to verify if the CTL property:

$$\phi \;=\; \mathbf{E}((p \vee q)\ \mathbf{U}\ ((p \wedge q) \wedge \mathbf{AX}q))$$

holds on K.

Do show in your explanation how you calculate $W_{\mathbf{E}((p\vee q)\ \mathbf{U}\ ((p\wedge q)\wedge \mathbf{AX}q))}$, and what W_ϕ finally is.

(THERE ARE MORE QUESTIONS; next page)

7. **Symbolic model checking** [1.5 pt].

   Let the Boolean formula:

   $$R(x, y, x', y') \;=\; \bar{x}yx' \vee x\bar{y}\bar{x'}$$

   encode the transitions of a program K. In this formula, the values of x, y represent the source states of a given transition, and x', y' represent the transition's destination states.

   xy means the conjunction $x \wedge y$, and $\bar{x}$ means the negation $\neg x$.

   (a) Suppose the formula xy describes all states of K where q holds. Give the Boolean formula that represents **EX** q.

   (b) As above, but now give the Boolean formula that represents **AX** q.

   (c) Construct the reduced OBDD representing R, using x', y', x, y as the ordering.

8. **Challenge** [0.5 pt].

   ....