You can review the latex source for this assignment-file to learn and use latex to prepare your homework submission. You will see the use of macros (to write uniformly formatted text), different text-styles (emphasized, bold-font), different environments (figures, enumerations).

It is not required that you use exactly this latex source to prepare your submission.

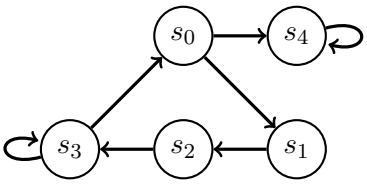
Homework 3 (CTL/LTL/BDD): ComS/CprE/SE 412, ComS 512

Due-date: April 4 at 11:59PM.

Submit online on Canvas two files: the source file in latex format and the pdf file generated from latex. Name your files: ⟨your-net-id⟩-hw3.⟨tex/pdf⟩.

Homework must be individual's original work. Collaborations and discussions of any form with any students or other faculty members or soliciting solutions on online forums are not allowed. Please review the academic dishonesty policy on our syllabus. If you have any questions/doubts/concerns, post your questions/doubts/concerns on Piazza and ask TA/Instructor.

1. For the following Kripke structure with $b \in L(s_3)$, $\{a, b\} \subseteq L(s_2)$, $a \in L(s_1)$ and $b \in L(s_4)$,



We are given a function defined as follows:

$$A_\varphi(Z) = [\![\varphi]\!]_M \ \cap \ R_\forall(R_\forall(Z))$$

where φ is some CTL formula, $[\![\varphi]\!]_M$ is the semantics of φ (in the context of a Kripke structure $M = (S, T, L)$) defined as a set of states of a Kripke structure, and $R_\forall$ is

$$R_\forall(Z) = \{s \mid \forall s'.(s, s') \in T \ \Rightarrow \ s' \in Z\}$$

(a) Compute the greatest fixed point A_b for the given Kripke structure.

Answer:

$R_\forall(Z) = \{s_0, s_1, s_2, s_3, s_4\} = Z$, therefore, $R_\forall(R_\forall(Z)) = Z$ and $[\![b]\!]_M = \{s_2, s_3, s_4\}$

$$A_b(Z) = [\![b]\!]_M \ \cap \ R_\forall(R_\forall(Z)) = \{s_2, s_3, s_4\} \cap \{s_0, s_1, s_2, s_3, s_4\} = \{s_2, s_3, s_4\}$$

$$A_b^2(Z) = [\![b]\!]_M \ \cap \ R_\forall(R_\forall(A_b(\{s_2, s_3, s_4\}))) = \{s_2, s_3, s_4\} \cap \{s_0, s_1, s_4\} = \{s_4\}$$

$$A_b^3(Z) = [\![b]\!]_M \ \cap \ R_\forall(R_\forall(A_b^2(\{s_4\}))) = \{s_2, s_3, s_4\} \cap \{s_4\} = \{s_4\}$$

$$A_b^4(Z) = [\![b]\!]_M \ \cap \ R_\forall(R_\forall(A_b^3(\{s_4\}))) = \{s_2, s_3, s_4\} \cap \{s_4\} = \{s_4\}$$

$$\vdots$$

Therefore, the greatest fixed point of $A_b(Z) = \{s_4\}$.

(b) **512 only** It is claimed that the semantics of the property expressed by the greatest fixed point of A_b cannot be expressed in CTL. Justify the validity of the claim.

Answer:

If we look at $R_\forall(R_\forall(\{s_2, s_3, s_4\}))$ for $A_b^2(Z)$, we get, $R_\forall(\{s_2, s_3, s_4\}) = \{s_1, s_2, s_4\}$ and $R_\forall(R_\forall(\{s_2, s_3, s_4\})) = \{s_0, s_1, s_4\}$. Therefore, $R_\forall(R_\forall(\{Z\}))$ captures next states of next states of current state satisfying the properties given in the question. This implies the greatest fixed points capture all the even states starting from s_4. However, all states starting from s_4 satisfies b, but the fixed points only capture the even states starting from s_4.

Therefore, it cannot be expressed in CTL.

(6+4(Extra Credit))

2. Your company requires that in any collaborative project, code commits must be reviewed by collaborator(s) of the person who commits the code. One fine morning, you wake up to see your collaborator has committed some late night code written in Pseudo-Lang with minimal comments. Your job is to review the commit and provide some constructive comments.

   The only code-comments are as follows:

   - The code utilizes ROBDD implementation where each node of the ROBDD is described using a structure *node* containing three properties: *name* of the decision variable for that node; a pointer to the root of the left-subtree (*left*); and a pointer to the root of the right-subtree (*right*).

     The terminal or leaf nodes are associated with the *name* true or false (and have null pointers for *left* and *right*).
   - The code corresponds to some function with two formal parameters each of type node, and returns a boolean type.

```
function wit(struct node n1, struct node n2) {
   if ((n1.name == true) && (n2.name == false)) return true;
   if ((n2.name == true) && (n1.name == false)) return true;
   if ((n1.left == null) || (n2.left == null)) return false;
   if (n1.name == n2.name) {
      if (wit(n1.left, n2.left))
          return wit(n1.right, n2.right)
      }
   return false;
}
```

(a) What is the functionality of the function *wit*?

Answer: It checks if two ROBDDs having the same set of decision nodes are inverse of each other, which means their terminal nodes are opposite of each other.

(b) What is the runtime of the function *wit*?

The function's runtime is equal to the exponential size of ROBDD.

(10)

3. For each of the following requirements, discuss whether LTL and CTL model checking can be used to verify the requirements. For instance, if your answer for some requirement is: it can be verified using LTL model checking but not using CTL model checking, then justify your answer.

   (a) Along all paths, propositions p and q hold true in alternate states.

   Answer: This is expressible in both CTL and LTL.

   CTL : $(p \vee q) \wedge \mathtt{AG}((p \rightarrow \mathtt{AX}(q)) \wedge (q \rightarrow \mathtt{AX}(p)))$

   LTL : $(p \vee q) \wedge \mathtt{G}((p \rightarrow \mathtt{X}(q)) \wedge (q \rightarrow \mathtt{X}(p)))$

   Therefore, it is verifiable in both LTL and CTL.

   (b) Along all paths, eventually a state is reached such that all its next states satisfy the proposition p.

   Answer: This can be expressed in CTL but not in LTL because LTL cannot capture branching behavior after reaching a state. The CTL formula satisfying the requirement is $\mathtt{AF}(\mathtt{AX}(p))$. Therefore, CTL model checking can be used to verify this property.

   (c) There exists no path where p holds true until q holds true.

   Answer: This can also be represented in both LTL and CTL.

   CTL: $\neg\mathtt{A}(p \ \mathtt{U} \ q)$

   LTL: $\neg(p \ \mathtt{U} \ q)$

   So, both LTL and CTL model checking can be used to verify this property.

   (d) There exists a path where p holds in a state and it is followed by a state where q holds.

   Answer:

   CTL representation of this formula: $\mathtt{EF}(p \wedge \mathtt{EX}(q))$.

   The statement's negation becomes "For all paths, p holds in a state, and it is not followed by a state where q holds", or we can also write negation as "For all paths, p does not holds in a state, and it is not followed by a state where q holds" and so on.

   It is evident that even if we negate the expression, the branching behavior remains. Therefore, it is expressible and verifiable in CTL but cannot be expressed and verified in LTL.

(12)

4. Prove or disprove the following claims:

   (a) For any path π in a Kripke structure, $\pi \models (\mathtt{F}(p) \ \mathtt{U} \ q)$ and $\pi[0]$ does not satisfy q implies that there must be some state(s) where p holds true before the state where q holds true.

   Answer: *disproved*, *Explanation*: Assume a path $\pi = s_0 s_1 s_2 s_3 ...$, $\pi[0] \not\models q$, and $\pi \models (\mathtt{F}(p) \ \mathtt{U} \ q)$. This implies $\pi[0] \models \mathtt{F}(p)$, and a path can satisfy $\mathtt{F}(p)$ if p becomes true at anywhere in the path for some $i \geq 0$. Therefore, $\pi \models (\mathtt{F}(p) \ \mathtt{U} \ q)$ does not demand p be true in states before q.

   Let's elaborate it as

   $\pi \models (\mathtt{F}(p) \ \mathtt{U} \ q)$ iff $\exists i.\pi^i \models q \wedge \forall j < i.\pi^j \models \mathtt{F}(p)$

   Now let's consider path starting from π^j , $\pi^j \models \mathtt{F}(p)$ iff $\exists k \geq 0.\pi^k \models p$

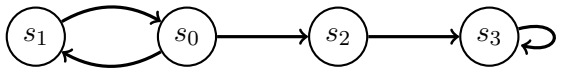
   Here, k value might be greater than i, and still $\pi^j \models \mathtt{F}(p)$ is satisfied.

   Therefore, this statement is incorrect.

(b) For any state in any Kripke structure, the state satisfies $\texttt{AX}(\texttt{AF}(p))$ if and only if the same state satisfies $\texttt{F}(\texttt{X}(p))$.

Answer: *proved*, *Explanation*: This is vacuously true. $\texttt{AX}(\texttt{AF}(p))$ implies for all paths starting from all next states of the current state, it is possible to eventually reach p. The equivalent LTL formula is $\texttt{X}(\texttt{F}(p))$. $\texttt{X}(\texttt{F}(p))$ is equivalent to $\texttt{F}(\texttt{X}(p))$.

For the following Kripke structure TS with $p \in L(s_1)$, $\{p\} \subseteq L(s_3)$ and s_0 as the initial state.



Here, $TS \models \texttt{AX}(\texttt{AF}(p))$ and it also satisfies the LTL formula $TS \models \texttt{X}(\texttt{F}(p))$.

(c) The LTL formula $\texttt{F}(\texttt{G}(p)) \Rightarrow \texttt{G}(\texttt{F}(p))$ is equivalent to propositional constant true.

Answer: *proved*, *Explanation*:

$\texttt{F}(\texttt{G}(p)) \Rightarrow \texttt{G}(\texttt{F}(p))$ can be written as $\neg(\texttt{F}(\texttt{G}(p))) \vee \texttt{G}(\texttt{F}(p))$ = $\texttt{G}(\texttt{F}(\neg p)) \vee \texttt{G}(\texttt{F}(p))$

Which is a tautology. Therefore, $\texttt{F}(\texttt{G}(p)) \Rightarrow \texttt{G}(\texttt{F}(p))$ is equivalent to propositional constant true.

(12)