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.

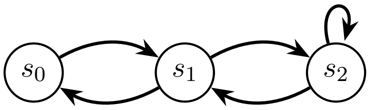
Homework 1 (CTL): ComS/CprE/SE 412, ComS 512

Due-date: Feb 7 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⟩-hw1.⟨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. Consider the following Kripke structure, with $p \in L(s_0) \cap L(s_2)$ and $q \in L(s_2)$.



Identify the set of states that satisfy each of the following:

(a) $\texttt{EX}(q)$
(b) $\texttt{AX}(p)$
(c) $\texttt{AX}(q)$
(d) $\texttt{AG}(p)$
(e) $\texttt{EG}(p)$
(f) $\texttt{AF}(p)$
(g) $\texttt{AG}(\texttt{EX}(p))$
(h) $\texttt{AG}(\texttt{AF}(p))$

(16 pts)

Answer: The tree of computation is -

(a) $\{s_1, s_2\}$
(b) $\{s_1\}$
(c) $\{\}$
(d) $\{\}$
(e) $\{s_2\}$
(f) $\{s_0, s_1, s_2\}$

(g) $\{\}$

(h) $\{s_0, s_1, s_2\}$

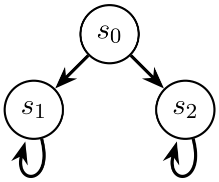
2. Express the following statements as CTL formula: (4+4+6 pts)

   (a) Along all paths **withdraw-money** is never true after **invalid-login**.
   Answer: AG (**invalid-login** $\Rightarrow$ AXAG $\neg$ **withdraw-money**)

   (b) Along all execution sequences of an elevator behavior, if the elevator door is **open** then the door remains **open** until a **request-to-move** is sent to the elevator.
   Answer: A ((**open** $\Rightarrow$ AG **open**) U **request-to-move**)

   (c) Whenever proposition **request-site-update** is true in a state, it is followed in zero or more steps by a state where proposition **updating-site** is true, which in turn is followed in one or more steps by a state where **update-complete** is true.
   Answer: AG ((**request-site-update** $\Rightarrow$AF **updating-site**) $\Rightarrow$ AXAF **update-complete**)

3. To disprove that two CTL formula are equivalent, you are required to draw a Kripke structure and identify a state in that structure, which satisfies one of the formula and does not satisfy the other. For instance, in order to disprove that $\mathtt{EX}(p) \wedge \mathtt{EX}(q)$ and $\mathtt{EX}(p \wedge q)$ are equivalent, we can draw the following the Kripke structure:



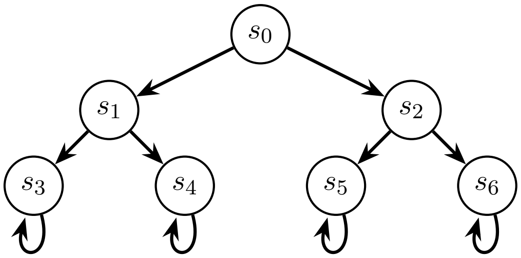
$L(s_1) = \{p\}, L(s_2) = \{q\}$

We present the labels for the relevant states in the Kripke structure and state that s_0 satisfies $\mathtt{EX}(p) \wedge \mathtt{EX}(q)$ and s_0 does not satisfy $\mathtt{EX}(p \wedge q)$. This is because there exists a path from s_0 where in the very next state s_1 the proposition p is true; thus conforming that s_0 satisfies $\mathtt{EX}(p)$ (similarly, one can justify the satisfiability of $\mathtt{EX}(q)$ at state s_0). On the other hand, there exists no path from s_0, where in the very next state both p and q are satisfied.

Disprove that the two CTL formula $\mathtt{AF}(p \wedge q)$ and $\mathtt{AF}(p) \wedge \mathtt{AF}(q)$ are equivalent. (5 pts)

Answer:

$\mathtt{AF}(p \wedge q)$ implies that $p \wedge q$ is true sometimes in all possible future states from starting state. However, $\mathtt{AF}(p) \wedge \mathtt{AF}(q)$ says that "In all possible future states, p must be true, and in all possible future states, q must be true". Therefore, p and q do not necessarily true in one state to satisfy $\mathtt{AF}(p) \wedge \mathtt{AF}(q)$, but $\mathtt{AF}(p \wedge q)$ demands p and q to be true in the same state.



$L(s_1) = \{p\}, L(s_3) = \{q\}, L(s_4) = \{q\}$
$L(s_2) = \{q\}, L(s_5) = \{p\}, L(s_6) = \{p\}$

The above Kripke structure satisfies $\mathtt{AF}(p) \wedge \mathtt{AF}(q)$, but it does not satisfy $\mathtt{AF}(p \wedge q)$. Therefore, $\mathtt{AF}(p \wedge q)$ and $\mathtt{AF}(p) \wedge \mathtt{AF}(q)$ are not equivalent.

4. To prove that one formula (say, φ_1) is "stronger" than another (say, φ_2), you need to prove that whenever in any state in any Kripke structure φ_1 holds, φ_2 holds in that state as well. In other words, φ_1 is "stronger" than φ_2 if and only if $\varphi_1 \Rightarrow \varphi_2$ is a tautology (always evaluates to true). For instance, to prove that $\mathtt{AX}(p)$ is stronger than $\mathtt{EX}(p)$ we can write:

$$\begin{aligned} \forall s.s \in \llbracket \mathtt{AX}(p) \rrbracket &\Leftrightarrow \forall \pi \in \mathit{Path}(s) : \pi[1] \in \llbracket p \rrbracket \\ &\Rightarrow \exists \pi \in \mathit{Path}(s) : \pi[1] \in \llbracket p \rrbracket \\ &\Rightarrow s \in \llbracket \mathtt{EX}(p) \rrbracket \end{aligned}$$

To disprove that $\mathtt{EX}(p)$ is stronger that $\mathtt{AX}(p)$, you will draw a Kripke structure and present a state in the Kripke structure that satisfies $\mathtt{EX}(p)$ but does not satisfy $\mathtt{AX}(p)$ (see the Kripke structure example in the previous problem).

Prove or disprove $\mathtt{AG}(p \ \wedge \ q)$ is stronger than $\mathtt{AG}(p \ \Rightarrow \ \mathtt{AF}(q))$.

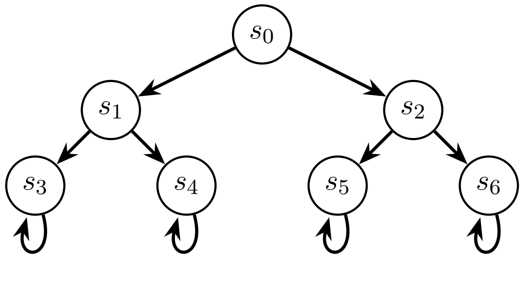
Answer:

Let's simplify $\mathtt{AG}(p \Rightarrow \mathtt{AF}(q))$.

$$\begin{aligned} \forall s.s \in \llbracket \mathtt{AG}(p \ \Rightarrow \ \mathtt{AF}(q)) \rrbracket &\Leftrightarrow \forall \pi \in \mathit{Path}(s) : \pi[0] \in \llbracket p \ \Rightarrow \ \mathtt{AF}(q) \rrbracket \\ &\Rightarrow \forall \pi \in \mathit{Path}(s) : \pi[0] \in \llbracket \neg p \vee \mathtt{AF}(q) \rrbracket \\ &\Rightarrow \forall \pi \in \mathit{Path}(s) : \pi[0] \in \llbracket \mathtt{AF}(q) \rrbracket \\ &\Rightarrow \forall \pi \in \mathit{Path}(s) : \pi[0] \in \llbracket q \rrbracket \end{aligned}$$

When $\mathtt{AG}(p \Rightarrow \mathtt{AF}(q))$ holds in s, it is immediately true that $p \Rightarrow \mathtt{AF}(q)$ holds for all successors of s, including s itself. We can rewrite $p \Rightarrow \mathtt{AF}(q)$ as $\neg p \vee \mathtt{AF}(q)$, which implies $\mathtt{AF}(q)$ true in s and its successors. without loss of generality, we can write q holds in state s. Furthermore, as $\mathtt{AF}(q)$ could be immediately true in the source s; therefore, we can interpret $p \Rightarrow \mathtt{AF}(q)$ as whenever p is true, q is also true.

$\mathtt{AG}(p \wedge q)$ implies $p \wedge q$ holds in all successors of s including s itself. As both p and q holds in state s, which suffices to show that $p \ \Rightarrow \ \mathtt{AF}(q)$ immediately true at that state. The Kripke structure satisfying $\mathtt{AG}(p \wedge q)$ is provided below.



$L(s_0) = \{p, q\}$, $L(s_1) = \{p, q\}$, $L(s_3) = \{p, q\}$, $L(s_4) = \{p, q\}$
$L(s_2) = \{p, q\}$, $L(s_5) = \{p, q\}$, $L(s_6) = \{p, q\}$

So, we can say that the formula $\mathtt{AG}(p \wedge q)$ is stronger than $\mathtt{AG}(p \Rightarrow \mathtt{AF}(q))$.

(5 pts)

5. **Extra Credit for 412; Required problem for 512.** We will define a new operator $A\circ$ as follows. A state s satisfies $\mathtt{A}(\varphi_1 \ \circ \ \varphi_2)$ if and only if for all paths starting from s at least one of the following holds

   - there exists a state where φ_2 is satisfied and before that φ_1 holds in all states in the path
   - all states in the path satisfy φ_1 and not φ_2

   Prove or disprove that

   (a) $A(p \ \circ$ false$)$ can be expressed in CTL.

   Answer: Let's first consider the first condition, which says for all paths starting from s finally satisfies $false$, and before that p holds in all state. In CTL, there is no way to satisfy $false$ formula along all paths. So, $A(p \ \circ$ false$)$ does not hold the first formula.

   The second condition says for all paths starting from s satisfy $p \wedge \neg false \Rightarrow p \wedge true \Rightarrow p$. Which can be written as $\mathtt{AG}(p)$.

   Therefore, $A(p \ \circ$ false$)$ can be expressed using CTL formula $\mathtt{AG}(p)$.

   (b) $A($false $\circ \ p)$ can be expressed in CTL.

   Answer: Let's first try to express the first condition in CTL, first condition says that for all paths there exists a state where p is satisfied, and before that $false$ holds in all states in the path. It can be immediately inferred that we cannot express this in CTL because these paths have negative path quantification, which cannot be expressed in CTL.

   The second condition says that $\mathtt{AG}(false \wedge \neg p) \Rightarrow \mathtt{AG}(false)$. Which is also not supported by CTL.

   Therefore, $A($false $\circ \ p)$ cannot be expressed in CTL.

   (10pts)