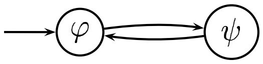
CSC2108: Automated Verification
Assignment 1 - Solutions

1. Solve the following problem: Use the definition of $\models$ between states and CTL formulas to explain why $s \models AGAF\varphi$ means that φ is true infinitely often along every path starting at s.

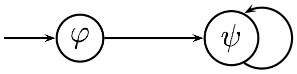
   Assume there is a path $s = s_0, s_1, \ldots$ on which φ is true only finite # of times. Then exists s_i on this path where $s_i \models \varphi$ and $\forall s_j \cdot j > i \Rightarrow \neg(s_j \models \varphi)$. For the successor of this node, s_{i+1}, $\neg(s_{i+1} \models AF\varphi)$. Since s_{i+1} is reachable from s and $\neg(s_{i+1} \models AF\varphi)$, then $\neg(s \models AGAF\varphi)$. Contradiction.

2. Which of the following pairs of CTL formulas are equivalent? For those which are not, exhibit a model of one of the pair which is not a model of the other. Otherwise, use the definition of CTL formulas and relationships between them to show that the formulas are equivalent.

   (a) (EG φ) $\vee$ (EG ψ) and EG ($\varphi \vee \psi$)

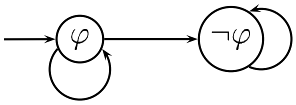
   (b) (AF φ) $\wedge$ (AF ψ) and AF ($\varphi \ \wedge \ \psi$)

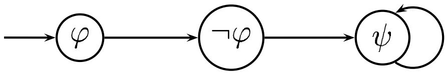
   (c) $\top$ and (AG φ) $\rightarrow$ (EG φ)

   These are equivalent, provided the transition relation is total.

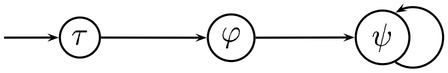
   (d) $\top$ and (EG φ) $\rightarrow$ (AG φ)

   (e) A [φ U ψ] and $\varphi \wedge$ AF(ψ)

   (f) A [φ U ψ] $\vee$ A [τ U ψ] and A[($\tau \vee \varphi$) U ψ]

   (g) A[φ U A[ψ U τ]] and A[A[φ U ψ] U τ]

   TODO

3. Express the following properties in CTL. You may invent any boolean variables necessary to express your atomic propositions. Do not use the pattern system.

(a) Whenever we get into that situation, we will sometimes be able to get out of it.
$AG(x \Rightarrow EF\neg x)$

(b) Both of those things may happen, but not at the same time.
$\neg EF(x \wedge y) \wedge (EFx) \wedge (EFy)$

(c) If that ever happens, it won't keep happening forever.
$AG(x \Rightarrow AF\neg x)$

(d) When this situation happens, it may persist for a while, but not forever, and it is always followed immediately by that situation.
$AG(x \Rightarrow A[x \; U \; (\neg x \wedge y)])$

(e) One should be able to get candy and drinks from the vending machine, but not both at the same time.
$AG(EF\text{candy} \wedge EF\text{drink} \wedge \neg(\text{candy} \wedge \text{drink}))$

(f) An elevator should keep its doors open until there is a request to use it.
$AG(\text{open} \Rightarrow A[\text{open} \; U \; \text{request}])$

(g) After p, q is never true. Express it so that the constraint is meant to apply on all computation paths.
$AG(p \Rightarrow AXAG\neg q)$

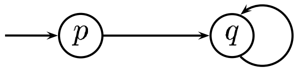
(h) Transitions to states satisfying p occur at most twice on all computation paths.
Let "transition to state satisfying p" be indicated as $\neg p \wedge EXp$. Then, the answer is

$$\neg EF(\neg p \wedge EX(p \wedge EF(\neg p \wedge EX(p \wedge EF(\neg p \wedge EXp)))))$$

(i) p precedes s and t on all computation paths. (Hint: try the negation of this specification first.)
$\neg E[\neg p \; U \; s] \wedge \neg E[\neg p \; U \; t]$

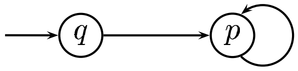
4. Which of the following pairs of LTL formulas are equivalent? For the equivalent formulas, show the equivalence. For those that are not,exhibit a model of one of the pair that is not the model of the other:

(a) $\diamond(p \wedge q)$ and $\diamond p \ \wedge \ \diamond q$



(b) $\Box \diamond (p \vee q)$ and $\Box \diamond p \vee \Box \diamond q$
These are equivalent. Intuitively, the first formula means that $p \vee q$ must occur infinitely often. However, this is possible only if either p or q occurs infinitely often.

(c) $\circ(p\ U\ q)$ and $(p\ U\ \circ q) \vee q$



(d) $\circ \diamond p$ and $\diamond \circ p$

These are equivalent. The proof follows trivially from their definitions.

5. Do $\circ$ and U form the adequate set for LTL? If so, prove it.

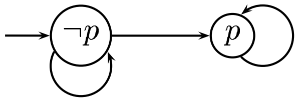
We need to show how to express $\Box$ and $\diamond$ using these two operators.

$$\begin{aligned} \Box p &= \neg \diamond \neg p \\ \diamond p &= \text{true}\ U\ p \end{aligned}$$

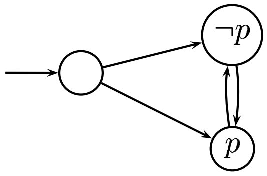
6. Find a transition system which distinguishes the following pairs of formulas (CTL and LTL). That is, show that they are *not* equivalent:

(a) $\Box \diamond p$ and AG AF p DO NOT DO THIS.

(b) $\Box \diamond p$ and AG EF p



(c) $\circ p \vee (\circ \circ p)$ and AX $p \vee$ AX AX p



7. Show that the following LTL formulas are *valid*, i.e., true in any state of any model

(a) $\neg q$ U $(\neg p\ \wedge\ \neg q) \rightarrow \neg \Box p$

Notice that $\neg \Box p = \diamond \neg p$, the result then follows.

(b) $\Box \neg q\ \wedge\ \diamond \neg p \rightarrow \neg q$ U $(\neg p\ \wedge\ \neg q)$

(c) Expand $\neg((p$ U $q) \vee \Box p)$ using de Morgan rules and the LTL equivalence $\neg(a$ U $b) \equiv (\neg b$ U $(\neg a\ \wedge\ \neg b)) \vee \neg \diamond b$. Using this expansion and the facts above, show that $\neg((p$ U $q) \vee \Box p) \rightarrow \neg q$ U $(\neg p\ \wedge\ \neg q)$

Expand as suggested, and then use part (b).

8. Using the pattern hierarchy, express the following properties in CTL and LTL:

(a) When a client A makes a method call to a server B, it will eventually receive the results of its call if the server is OK.

(b) It is always the case that when the req-search-state is not enabled, then the req-close-state shall not be closed and will remain not closed until the req-search-state is enabled.

Pattern: “absence” pattern with “after-until” scope.

$$\Box(\neg\text{req\_search\_state\_enabled} \Rightarrow (\neg\text{req\_close\_state\_closed}\ W\ \text{req\_search\_state\_enabled}))$$

(c) After opening a network connection, an error message will pop up in response to a network error.

Pattern: “response” pattern with “after” scope.

$$\Box(\text{OpeningNetworkConn} \Rightarrow \Box(\text{NetworkError} \Rightarrow \diamond\text{ErrorMessage}))$$

(d) Every time the form is parton-view, it must have been preceded by a corresponding request-view.

Pattern: “precedence” pattern with “global” scope.

$$\neg\text{form} = \text{patron\_view}\ W\ \text{form} = \text{request\_view}$$

(e) Checkout is 0 until the status of the book is charged or on hold

Pattern: “precedence” pattern with “global” scope.

$$\text{checkOut} = 0\ W\ (\text{status} = \text{charged} \vee \text{status} = \text{hold})$$

9. Consider the property (using the property patterns): P is universal between S and T.

In LTL: $\Box((S \wedge \neg T \wedge \diamond T) \Rightarrow (P\ U\ T))$.

In CTL: $AG(S \wedge \neg T \Rightarrow A[(P \vee AG(\neg T))\ W\ T])$

(a) Is the interval open/closed on the left/right? (Use the LTL and CTL mappings to answer this question).

The interval is closed on the left and is open on the right.

(b) According to the mappings, when the specification is satisfied, must P occur when T first becomes true?

No, that what open on the left means.

(c) According to the mappings, when the specification is satisfied, can S occur without T?

Yes. For example, in the LTL mapping if S occurs without T then the antecedent of the implication is false, which makes the formula true vacuously.

(d) According to the mappings, when the specification is satisfied, if there is no matching T for an S, is P required to hold after the S with no matching T?

No. See previous answer for explanation.

(e) For each combination of open/closed on left/right, create the LTL universality pattern.

open left, open right

$$\Box((S \wedge \neg T \wedge \Diamond T) \Rightarrow \circ(P \; U \; T))$$

open left, closed right

$$\Box((S \wedge \neg T \wedge \Diamond T) \Rightarrow \circ(P \; U \; T \wedge P))$$

closed left, closed right

$$\Box((S \wedge \neg T \wedge \Diamond T) \Rightarrow (P \; U \; T \wedge P))$$

10. Repeat parts (a)-(d) of the previous problem for the specification P is universal after S until T.

In LTL: $\Box(S \wedge \neg T \Rightarrow (P \; W \; T))$

In CTL: $AG(S \wedge \neg T \Rightarrow A[P \; W \; T])$

(a) Is the interval open/closed on the left/right? (Use the LTL and CTL mappings to answer this question).

The interval is closed on the left and is open on the right.

(b) According to the mappings, when the specification is satisfied, must P occur when T first becomes true?

No, that what open on the left means.

(c) According to the mappings, when the specification is satisfied, can S occur without T?

Yes.

(d) According to the mappings, when the specification is satisfied, if there is no matching T for an S, is P required to hold after the S with no matching T?

Yes.