Instructor: Edmund M. Clarke TAs: Soonho Kong, David Henriques

Due date: 10/26/2011 cmu15414ta@gmail.com

Assignment 4

1 Monotonicity, Continuity, and Fixed Points

(a) Show that a monotonic function $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ is $\bigcup$-continuous if S is a finite set.

(b) Show that a monotonic function $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ has the greatest fixed point

$$\textbf{gfp}\ Z\ [\tau(Z)] = \cup\{Z \mid \tau(Z) = Z\}.$$

(c) **Extra Credit**: Show an infinite set S and a monotonic function $\tau : \mathcal{P}(S) \to \mathcal{P}(S)$ which is not $\bigcup$-continuous.

2 Semantics of Temporal Logic

(a) The "weak until" operator **W** has similar semantics to the "strong until" operator **U** except that $g_1\,\mathbf{W}\,g_2$ doesn't require g_2 to ever hold true if g_1 is globally true:

$$\pi \models g_1\,\mathbf{W}\,g_2 \quad \Leftrightarrow \quad \text{for all } j \geq 0, \text{ if } \pi^j \models \neg g_1 \text{ then there exists a } k \leq j \text{ such } \pi^k \models g_2$$

Show that **W** can be expressed in terms of the existing temporal operators:

1. Express $\pi \models g_1\,\mathbf{W}\,g_2$ in terms of the existing temporal operators (**F**, **G**, **U**). You may check your answer by consulting the Wikipedia page on Linear Temporal Logic.
2. Prove your expression in part (a) means the same as $\pi \models g_1\,\mathbf{W}\,g_2$ provided $\pi \models \mathbf{F}\,g_2$.
3. Prove your expression in part (a) means the same as $\pi \models g_1\,\mathbf{W}\,g_2$ provided $\pi \models \neg\,\mathbf{F}\,g_2$.

(b) Write an expression equivalent to $\mathbf{AG\,EF}\,p$ without using any of the operators $\{\mathbf{A}, \mathbf{F}, \mathbf{G}\}$. (Hint: use the identities on page 9 of lecture 11 slides)

(c) Prove that $\mathbf{A\,G\,F}\,p$ is equivalent to $\mathbf{AG\,AF}\,p$.

(d) Show that $\mathbf{A}[f\ \mathbf{U}\ g] \equiv \neg\mathbf{E}[\neg g\ \ \mathbf{U}\ \ \neg f \wedge \neg g] \wedge \neg\mathbf{EG}\ \neg g$

Hints:

1. Note that $\neg\,\mathbf{E}\,x = \mathbf{A}\,\neg x$ and $\mathbf{A}(x \wedge y) = (\mathbf{A}\,x) \wedge (\mathbf{A}\,y)$.
2. Rewrite $[f\ \mathbf{U}\ g]$ as $[(f\ \mathbf{W}\ g) \wedge \mathbf{F}\ g]$.
3. Show that $(\pi \models \neg(f\,\mathbf{W}\,g)) \equiv (\pi \models \neg g\ \mathbf{U}\ \neg f \wedge \neg g)$.

(e) This is a problem given in Huth and Ryan's book. Consider the model M in Figure 1. Check whether $M, s0 \models \phi$ and $M, s2 \models \phi$ hold for the CTL formula ϕ:

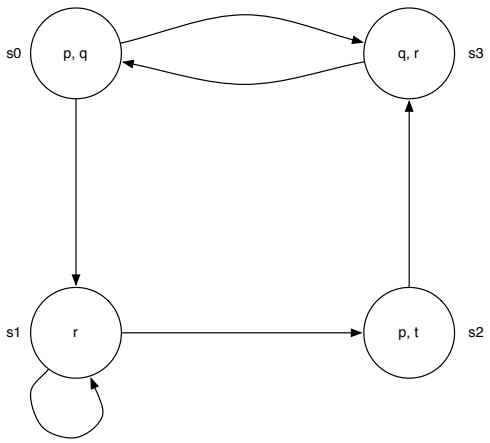


Figure 1: A model with four states

1. **AF** q
2. **AG**(**EF** $(p \vee r)$)
3. **EX**(**EX** r)
4. **AG**$(AF\ q)$

3 CTL Model Checking and Fixed Points

Remark: In class, some of the operators were introduced as abbreviations for sequences of other operators. In this exercise, for the sake of brevity, we will use the semantics presented in the book, page 29, which gives semantics directly to all operators. It is a good (non-mandatory) exercise to show the equivalence between these direct semantics and the semantics inherited from abbreviation.

Alternative representations

Two formulae are said to be equivalent in CTL if they have the same semantics. For example, in class, we have made the following claim, which we now prove

$$\mathbf{EF}\, p \equiv p \vee \mathbf{EX}\, \mathbf{EF}\, p$$

Proof: ($\Rightarrow$) Assume $M, s \models \mathbf{EF}\, p$, then there is a path $\pi = \pi_0 \cdot \pi_1 \cdot \pi_2 ...$ s.t. $\pi_0 = s$, which means there exists a $k \geq 0$ s.t. $M, \pi_k \models p$ which means $p \in L(\pi_k)$. If several k have this property, take the smaller one. There are two cases:

$k = 0$ Then, since $\pi_0 = s$, $M, s \models p \vee \mathbf{EX}\, \mathbf{EF}\, p$ holds since $M, s \models p$ because $p \in L(s)$

$k > 0$ Then we have to show $M, s \models \mathbf{EX}\, \mathbf{EF}\, p$, which means we have to find a path $\dot{\pi}$ s.t. $\dot{\pi}_0 = s$ and $M, \dot{\pi}_1 \models \mathbf{EF}\, p$ with. Take $\dot{\pi}$ to be π. We now have to check if $M, \pi_1 \models \mathbf{EF}\, p$, which means we have to find a path $\ddot{\pi}$ s.t. $\ddot{\pi}_0 = \pi_1$ and s. t. there is $\ddot{k} \geq 0$ for which $M, \ddot{\pi}_{\ddot{k}} \models p$. Take $\ddot{\pi}$ to be $\pi_1 \cdot \pi_2 ...$, which is π except for the first state. Notice that $\pi_i = \ddot{\pi}_{i-1}$. Obviously $\ddot{\pi}_0 = \pi_1$ and we know $M, \pi_k \models p$, therefore $M, \ddot{\pi}_{\ddot{k}} \models p$ for $\ddot{k} = k - 1 \geq 0$.

($\Leftarrow$) Assume $M, s \models p \vee \mathbf{EX}\,\mathbf{EF}\,p$. Then either $M, s \models p$ or $M, s \models \mathbf{EX}\,\mathbf{EF}\,p$.

In the former case, $M, s \models \mathbf{EF}\,p$ by taking any path starting with s and taking $k = 0$.

In the latter case, there is a path π s.t. $\pi_0 = s$ and s.t. $M, \pi_1 \models \mathbf{EF}\,p$. This, in turn, means that there is a path $\dot{\pi}$ s.t. $\dot{\pi}_0 = \pi_1$ and $\dot{k} \geq 0$ s.t $M, \dot{\pi}_k \models p$. Consider the path $\ddot{\pi} = \pi_0 \cdot \dot{\pi}_0 \cdot \dot{\pi}_1 ...$, which is $\dot{\pi}$ prefixed by π_0. $\ddot{\pi}_0 = s$ and for $\ddot{k} = \dot{k} + 1$, $M, \ddot{\pi}_{\ddot{k}} \models p$, which means $M, s \models \mathbf{EF}\,p$.

(a) Prove the following equivalencies

$$\mathbf{EG}\,p \equiv p \wedge \mathbf{EX}\,\mathbf{EG}\,p$$

$$\mathbf{AF}\,p \equiv p \vee \mathbf{AX}\,\mathbf{AF}\,p$$

Hint: You may want to use contrapositive reasoning for the second proof.

Remark: Other CTL constructs can be similarly represented in this way.

Formulae as sets of states

The problem of model checking for CTL can be stated as "given a Kripke structure M, a state s and a CTL formula φ, is it true that $M, s \models \varphi$?". An equivalent formulation is "given a Kripke structure M, and a CTL formula φ, determine the set $\{s \in S : M, s \models \varphi\}$" (think for a moment why these formulations are equivalent). In order to address the latter question, it will be useful to think of temporal formulae as functions from states to states.

For example, suppose your formula φ is $\mathbf{EX}\,\psi$ for some (possibly complicated) formula ψ. Assuming $\psi(S)$ returns the set of states that satisfy ψ, we can easily compute the set of states that have a predecessor in $\psi[S]$. We are essentially looking at $\mathbf{EX}$ as a function $\mathbf{EX} : 2^S \to 2^S$ s.t. $\mathbf{EX}[Q] = \{s \in Q : \text{state } s \text{ has a transition to at least one state in } Q\}$. Given state Q, this set is easy to compute.

More formally, if φ is:

p	$p(Q) = \{s \in S : p \in L(s)\}$;
$\neg\psi$	$(\neg\psi)[Q] = S - \psi[Q]$;
$\psi_1 \vee \psi_2$	$(\psi_1 \vee \psi_2)[Q] = \psi_1[Q] \cup \psi_2[Q]$;
$\psi_1 \wedge \psi_2$	$(\psi_1 \wedge \psi_2)[Q] = \psi_1[Q] \cap \psi_2[Q]$;
$\mathbf{EX}\,\psi$	$\mathbf{EX}\,\psi[Q] = \{s \in S : \text{state } s \text{ has a transition to at least one state in } \psi[Q]\}$;
$\mathbf{AX}\,\psi$	$\mathbf{AX}\,\psi[Q] = \{s \in S : \text{all transitions form state } s \text{ lead to states in } \psi[Q]\}$;
$\mathbf{EF}\,\psi$	$\mathbf{EF}\,\psi[Q] = \{s \in S : \text{there is a path departing from } s \text{ that eventually reaches a state in } \psi[Q]\}$;
$\mathbf{AF}\,\psi$	$\mathbf{AF}\,\psi[Q] = \{s \in S : \text{all paths departing from } s \text{ eventually reach a state in } \psi[Q]\}$;
$\mathbf{EG}\,\psi$	$\mathbf{EG}\,\psi[Q] = \{s \in F : \text{there is a path departing from } s \text{ that never leaves states in } \psi[Q]\}$;
$\mathbf{AG}\,\psi$	$\mathbf{AF}\,\psi[Q] = \{s \in S : \text{all states in all paths departing from } s \text{ never leave } \psi[Q]\}$;

If we have efficient ways of computing all these functions, we have a recursive procedure for model checking (just compute $\varphi[S]$). Notice that p does not care about its input because in practice, it will always recieve S as input.

It is easy to find efficient implementations for the first six connectives. Unfortunately, naive algorithms for the remaining connectives are not very efficient. We will see in the next section how to

use the fixed point theorems we learned in class to solve this problem.

(b) Assume φ is a CTL formula and that you are explicitly given a Kripke structure as a transition graph with N states $\{1, ..., N\}$ and a transition matrix[1] M.

Write down the pseudocode for $\mathbf{EX}\,\varphi[S]$ assuming that you have a function isPhiSatisfied s.t. isPhiSatisfied$(i) = 1$ if φ is satisfied in state i and isPhiSatisfied$(i) = 0$ otherwise.

Write down the pseudocode for $\mathbf{AX}\,\varphi[S]$ assuming that you have a function isPhiSatisfied s.t. isPhiSatisfied$(i) = 1$ if φ is satisfied in state i and isPhiSatisfied$(i) = 0$ otherwise.

For this problem, you may use standard mathematical notation for sets and set operations ($\{x, y, ..\}, \cup, \cap, -$, etc...). Both functions should have roughly the same efficiency.

Connectives and fixed points

We will now find a way to efficiently implement $\mathbf{EF}, \mathbf{AF}, \mathbf{EG}$ and $\mathbf{AG}$. In order to do that, we will use the alternative representation from the first part of this exercise. This may seem counterintuitive because the alternative expression seems more complicated than the usual representation, but as we will see, it will prove to be very useful. We will use $\mathbf{EF}$ as a case study.

First notice that, since the formulae are equivalent, the set of states that satisfies them must be the same

$$\mathbf{EF}\,\varphi[S] = (\varphi[S] \vee \mathbf{EX}\,\mathbf{EF}\,\varphi)[S]$$

Now notice that $\mathbf{EF}$ appears on both sides of the equation:

$$\underline{\mathbf{EF}\,\varphi}[S] = (\varphi \vee \mathbf{EX}\,\underline{\mathbf{EF}\,\varphi})[S] = \varphi[S] \cup \mathbf{EX}[\underline{\mathbf{EF}\,\varphi}[S]]$$

This motivates us to write the right hand side as a function $\tau : 2^S \to 2^S$

$$\tau(X) = \varphi[S] \cup \mathbf{EX}[X]$$

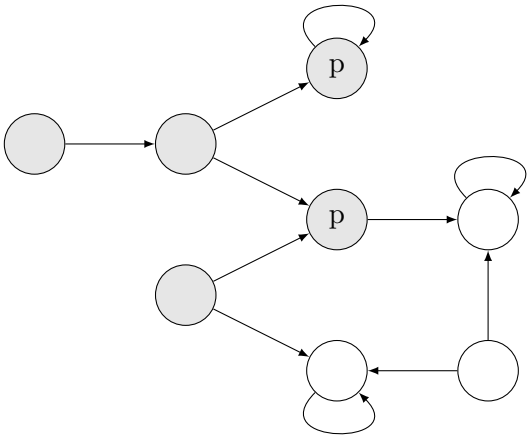
because we now know that $\mathbf{EF}\,\varphi[S]$ is a fixed point of this equation:

$$\tau(\mathbf{EF}\,\varphi[S]) = \varphi[S] \cup \mathbf{EX}[\mathbf{EF}\,\varphi[S]] = \mathbf{EF}\,\varphi[S]$$

Unfortunately, $\mathbf{EF}\,\varphi[S]$ it is not the only fixed point of the equation.

(c) Below is a Kripke strucutre with the states that satisfy $\mathbf{EF}\,p[S]$ shaded. This set of states is a fixed point of $\tau(X) = p[S] \cup \mathbf{EX}[X]$.

[1] A transition matrix for a graph with N vertices is a $N \times N$ matrix s.t. $M[i, j] = 1$ if there is a transition from state i to state j and $M[i, j] = 0$ otherwise.



Find and shade in the states of another fixed point of τ in the same Kripke structure.

The good news, however, is that $\mathbf{EF}\,\varphi[S]$ is the *least* fixed point of τ! (see pages 63, 64 and 65 of the book for a proof of this claim). Why is this good news, you ask?

Well, it is easy to see that τ is monotonic. Since the set of states is finite, by problem 1(a) of this homework, τ is also $\cup$-continuous. We have seen in class an efficient way of computing the least fixed point of τ by using Tarksi's Lemma. We just need to iterate $\tau(\emptyset), \tau(\tau(\emptyset)), ...$ until we reach a fixed point. This procedure allows us to efficiently get the set of sates that satisfies $\mathbf{EF}\,\varphi[S]$ provided we have (recursively) computed $\varphi[S]$.

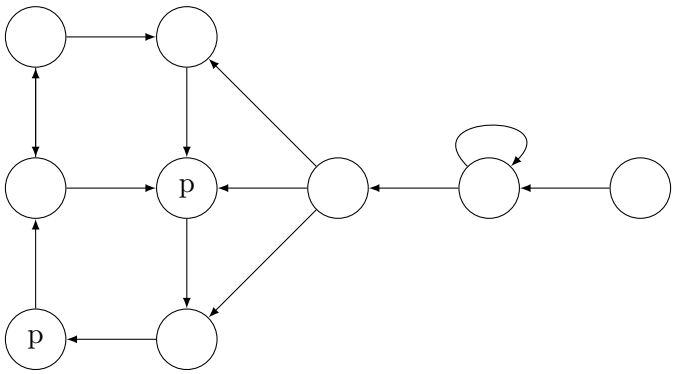
The other connectives also have fixed point representations that can be computed in similar ways; sometimes we have to use greatest fixed points instead of least fixed points, but the underlying principle is the same. Refer to chapter 6 of the book for examples.

Remark: For the sake of brevity, we left the until operators out of this discussion. They work exactly as the others but take longer to explain, that's the only reason why they've been left out. You can find them in the book or discuss them with the TAs if you are curious about them.

And that's it, now you know how to model check efficiently. Congratulations!

(d) Be the model checker!

Consider the following Kripke strucutre. Model check the formula $\mathbf{EX}\,\mathbf{AF}\,p$ in it. Make sure to show all iterations of the fixpoint algorithm!



Hint : remember that $\mathbf{AF}\, p \equiv p \vee \mathbf{AX}\, \mathbf{AF}\, p$.