Assignment 7
Time Flies

15-414: Bug Catching: Automated Program Verification

Due Friday, April 25, 2025
70 pts

This assignment is due on the above date and it must be submitted electronically on Gradescope. Please carefully read the policies on collaboration and credit on the course web pages at http://www.cs.cmu.edu/~15414/assignments.html.

What To Hand In

You should hand in the following files on Gradescope:

- Submit a PDF containing your answers to the written questions to Assignment 7 (Written). You may use the file `asst7.tex` as a template and submit `asst7.pdf`.

Using LaTeX

We prefer the answer to your written questions to be typeset in LaTeX, but as long as you hand in a readable PDF with your solutions it is not a requirement. We package the assignment source `asst7.tex` with handout to get you started on this.

1 Linear Time Logic

Task 1 (15 pts). Your job is to design a specification for a basic elevator system that services four floors. There is a door at each floor, with a call button and an indicator light that indicates whether the elevator has been called to that floor. Describe a set of atomic propositions and LTL formulas to specify the following properties of the elevator.

1. On each floor, a door will open eventually.
2. A door never opens if the elevator is not present at the corresponding floor.
3. Pushing a call button results in the elevator eventually servicing the corresponding floor.
4. The elevator returns to floor 0 infinitely often.
5. When the call button on floor 4 is pressed, the elevator serves it without stopping at any other floors along the way.

You should use the following atomic propositions to describe the elevator system:

- d_i: the door at floor i is open.
- c_i: the call button at floor i is pressed.
- e_i: the elevator is at floor i.
- l_i: the light at floor i is on.

"Serving" a floor means moving to the floor, and then opening the door.

2 Computation Tree Logic

Task 2 (15 pts). Draw a Kripke structure that satisfies the formula $\mathbf{A}[a \,\mathbf{U}\, \mathbf{AF}\, b] \wedge \mathbf{EX}\, \neg b$.

Task 3 (15 pts). For each state in your answer to Task 2, label which of the formulas $\mathbf{AF}\, b$, $\mathbf{EX}\, \neg b$, and $\mathbf{A}[a \,\mathbf{U}\, \mathbf{AF}\, b]$ are satisfied. You may refer to them as P, Q, and R, respectively.

3 Weakly Until

Task 4 (15 pts). Consider an LTL operator with the following semantics:

$$\sigma \models P\mathbf{W}Q \text{ iff, for all } i \geq 0, \text{ if } \sigma^i \models \neg P, \text{ then there exists } k \leq i \text{ such that } \sigma^k \models Q$$

This is a weaker version of the normal until operator, in that it doesn't require Q to eventually hold as long as P always does. Show that $\mathbf{W}$ can be expressed in terms of existing LTL operators by writing an equivalence. I.e., find an LTL formula R built from the standard operators covered in lecture for which the following holds:

$$P\mathbf{W}Q \leftrightarrow R$$

Use the semantics of LTL to justify why your equivalence is correct.

4 Strange Computations

Task 5 (10 pts). Recall that a computation structure $K = (W, W_0, \curvearrowright, v)$ with initial states $W_0 \subseteq W$ satisfies a CTL formula P if and only if each initial state $s \in W_0$ satisfies P:

$$K \models P \text{ if and only if for all } s_0 \in W_0 . s_0 \models P$$

This definition has a strange property, where it is possible that a given structure K there exists a formula P where $K \not\models P$ and $K \not\models \neg P$. Find a CTL formula and transition system for which this is the case. *Hint:* Strive for simplicity. There are many correct answers to this problem, and some of them are very simple. Start by thinking of very simple formulas, and then try to find a small computation structure over which this is true.