Assignment 7
Time Flies

15-414: Bug Catching: Automated Program Verification

Due 23:59pm, Friday, April 26, 2024
65 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. You may use the file `asst7-sol.tex` as a template and submit `asst7-sol.pdf`. You can generate this file by running `make` (assuming you have pdflatex in your system).

  Make sure your PDF solution files are up to date before you create the handin file.

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` and a solution template `asst7-sol.tex` in the handout to get you started on this.

1 Solving formulas with DPLL(T) (15 pts)

Consider the following formula:

$$\varphi : f(x) = f(y) \wedge (g(x) \neq g(y) \vee x \neq y)$$

Recall the DPLL(T) algorithm given in Lecture 19, where we can determine the satisfiability of this formula by performing the following procedure:

1. Construct the Boolean abstraction $\mathcal{B}(\varphi)$;
2. If $\mathcal{B}(\varphi)$ is unsatisfiable then φ is unsatisfiable;
3. Otherwise, get an interpretation I for $\mathcal{B}(\varphi)$;
4. Construct $\omega = \bigwedge_{i=1}^{n} P_i \leftrightarrow I(P_i)$;
5. Send $B^{-1}(\omega)$ to the T-solver;
6. If T-solver reports that $B^{-1}(\omega)$ is satisfiable then φ is satisfiable;
7. Otherwise, update $\mathcal{B}(\varphi) := \mathcal{B}(\varphi) \wedge \neg\omega$ and return to step 2.

Task 1 (15 pts). Use the DPLL(T) algorithm to show that φ is satisfiable or unsatisfiable.

2 Linear Temporal Logic (LTL) (20 pts)

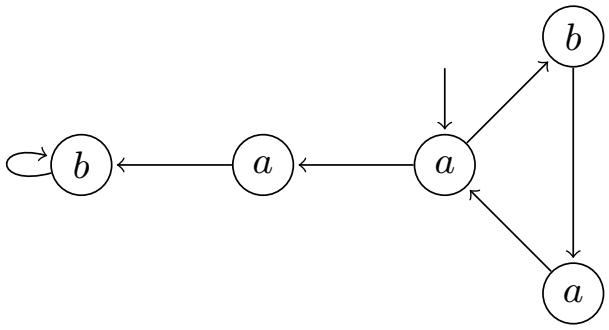
Consider the LTL equivalences that characterize distributive properties of temporal operators in Tasks 2 and 3. First, identify which of those equivalences are correct and which are not. Then use the semantics of LTL given in Lecture 21 to justify your answer with a proof. For the formulas that are not correct, describe an infinite trace that satisfies one side of the equivalence but not the other, i.e., provide a counterexample.

Task 2 (10 pts). $\Diamond(P \wedge Q) \leftrightarrow \Diamond P \wedge \Diamond Q$

Task 3 (10 pts). $\Box(P \wedge Q) \leftrightarrow \Box P \wedge \Box Q$

3 Computation Tree Logic (CTL) (15 pts)

Consider the computation structure given below. Determine whether it satisfies the CTL formulas in Tasks 4, 5 and 6. If it does satisfy the formula provide a brief justification, if it does not satisfy the formula provide a counterexample path to demonstrate the inconsistency.



Task 4 (5 pts). $\mathbf{A}\Diamond(a \wedge \mathbf{AX}a)$

Task 5 (5 pts). $\mathbf{A}\Box(\mathbf{A}a\mathbf{U}b)$

Task 6 (5 pts). $\mathbf{A}\Diamond(\mathbf{E}\Box b)$

4 LTL vs. CTL (15 pts)

LTL formulas P and Q are equivalent when for any path σ, $\sigma \models P$ whenever $\sigma \models Q$. Likewise, CTL formulas P and Q are equivalent whenever for any state s in any Kripke structure K, $s \models P$ whenever $s \models Q$. Recall that unlike with LTL, CTL formulas contain path quantifiers which denote that a temporal property either holds on some path starting at a state (**E**), or on all paths starting at a state (**A**).

Task 7 (15 pts). Show that the following pair of CTL and LTL formulas are not equivalent:

$$\mathbf{A}\Box(\mathbf{E}\Diamond a) \qquad \Box\Diamond a$$

To do so, write down a computation structure that satisfies one but not the other. Show that this is the case by providing a counterexample path for the non-statisfied formula, and explaining why the other is modeled by your system.