Assignment 2: Loops and Proofs
15-424/15-624 Foundations of Cyber-Physical Systems
Course TAs: João Martins (jmartins@cs), Annika Peterson (apeterso@andrew)

Due: **Beginning of class**, Wednesday 9/24/14
Total Points: 60

1. **Valid, Satisfiable, or Unsatisfiable.** For each of the following d$\mathcal{L}$ formulas, determine if they are valid, satisfiable, and/or unsatisfiable. **Very briefly justify**. Recall that the diamond modality $\langle\alpha\rangle\phi$ holds if there exists at least one run of HP α such that ϕ holds.

   (a) $[?true]true$

   (b) $[?true]false$

   (c) $[?false]true$

   (d) $[?false]false$

   (e) $[(?false)^*]false$

   (f) $[x' = 1 \ \&\ false]false$

   (g) $[(x' = 1 \ \&\ false)^*]false$

   (h) $x > 0 \rightarrow (y > 0 \rightarrow x > 0)$

   (i) $(x > 0 \rightarrow y > 0) \rightarrow x > 0$

   (j) $x = 5 \rightarrow [x' = 1 \ \&\ x \leq 5]false$

   (k) $[(x := -1 \cup x := 1); ?x > 0]x > 0$

   (l) $[?x \geq 1; x' = 1 \ \&\ true]x > 1 \leftrightarrow [?x \geq 1; x' = 1]x > 1$

   (m) $[x' = 1]x > 1 \leftrightarrow [x' = 1]x > 2$

   (n) $[x' = 1 \ \&\ x > 0]x > 1 \leftrightarrow [?x > 0]x > 1$

   (o) $\langle x' = v, v' = 1 \ \&\ x \geq 10\rangle x \geq 10 \leftrightarrow \langle x' = v, v' = 1; ?x \geq 10\rangle x \geq 10$

   (p) $[t := 0;\ t' = 1 \ \&\ t \leq 10; ?t = 10]t = 10$

   (q) $[t := 50; t' = 1 \ \&\ t \leq 10; ?t = 10]t = 10$

   (r) $[y := 1]x > 0 \leftrightarrow [z' = 1]x > 0$

2. **Practice with hybrid programs.**

   (a) Nondeterministic Evolution: Rewrite the d$\mathcal{L}$ formula $[x' = 0 \ \&\ H]\phi$ as an equivalent formula which does not use nondeterministic evolution through differential equations.

   (b) While Loop: The C0 programming language, introduced in 15-122 Principles of Imperative Computation, allows programmers to define contracts which must be satisfied. The `@requires` contract defines initial conditions which must be met, while the `@ensures` contract defines a final condition the program must satisfy. The semantics of C0 are as expected and can be found online[1]. Translate this while loop in C0 to a hybrid program α that has an equivalent set of traces, i.e. that executes the same transitions. Then create a d$\mathcal{L}$ formula using α that proves the contracts are satisfied. Since HPs don't have return statements, just ensure that the value of variable `res` is correct at the end of the HP. *Be careful with non-determinism!*

[1] `http://c0.typesafety.net`

```
int sum (int k, int n)
//@requires n >= 0;
//@ensures \result >= 1;
//@ensures \result = k*n;
{ int res = 0; int i = 0;
  while (i < n) {
    res = res + k;
    i = i + 1;
  }
  return res;
}
```

3. **Transition relation.** In class, we gave you the semantics of hybrid programs defined as a transition relation $\rho(\alpha) \subseteq S \times S$. We used statements of the form $(\nu, \omega) \in \rho(\alpha)$ to talk about these semantics.

   In recitation, you helped us define the semantics as a function $R(\alpha) : S \to 2^S$ of the program α and of an initial state ν. It would return the set of states that could be reached from ν through α. We would use statements of the form $\omega \in R(\alpha)(\nu)$ to talk about these semantics.

   In our ongoing efforts to transfer all of our work on semantics over to you, you are now going to define a new semantics on your own!

   (a) Define the semantics of hybrid programs as a function $\zeta(\alpha) : 2^S \to 2^S$, which gives you the set of states reachable from any of the initial input states. You cannot use the ρ or R semantics to define the ζ semantics.

   (b) Do you see any pros/cons of using this definition? Very briefly describe them.

4. **Soundness.** Give a proof of soundness for the following axioms. Focus on using the definitions of the semantics of hybrid programs and formulas - and use original $\rho(\alpha)$ only.

   (a) $([;]) : \quad [\alpha;\beta]\phi \leftrightarrow [\alpha][\beta]\phi$

   (b) $([?]) : \quad [?H]\phi \leftrightarrow (H \to \phi)$

   Hint: Use the semantics of hybrid programs[2], e.g. statements of the form $(\nu, \omega) \in \rho(\alpha)$ and the semantics of $\mathsf{d}\mathcal{L}$ formulas[3], e.g. statements of the form $\nu \models \phi$. You don't need *that* much text.

   Follow the process we used in recitation. The axiom is of the form $\phi_1 \leftrightarrow \phi_2$. Let ν be an arbitrary state, and assume it satisfies the formula ϕ_1, so $\nu \models \phi_1$. Now, simply apply the definitions of the semantics until you've gotten rid of most syntax. Then, reason why this is similar to the meaning of ϕ_2, and use the semantics in the opposite direction, adding back syntax until you obtain $\nu \models \phi_2$.

5. **Loops and invariants**

   Remember Lab 1? Wasn't that fun? Let's add stars to make it even *more* fun! In real cyber-physical systems, control isn't executing all the time. CPS controllers typically poll sensors and decide on what to do at regular intervals. As a step in this direction, in this exercise, we allow continuous evolution to happen for at most time T.

$$vel > 0 \ \wedge \ pos < station \ \wedge \ T > 0 \wedge acc = \underline{\qquad} \ \to$$

$$\big[(t := 0; pos' = vel, vel' = acc \ \& \ v \geq 0 \wedge t < T)^* \, ; ?(vel = 0)\big] \, pos = station$$

   (a) Find the value of *acc* for which the robot will stop at exactly the station. If your robot was efficient, you already solved this for Lab 1!

[2] page L3.11 of `http://symbolaris.com/course/fcps14/03-choicecontrol.pdf`

[3] page L2.13 of `http://symbolaris.com/course/fcps14/02-diffeq.pdf`

(b) There is no guarantee that the robot will stop within the first T time units, so multiple loops of the program might be required before the car does actually stop. But we have no clue how many! What do we do? *Invariants to the rescue!* Recall that an invariant of α^* is true no matter how many iterations of the α execute. If it holds before α executes, it holds after α executes. Invariants will generally relate the different state variables in a way that isn't altered by the dynamics.

Find an invariant for this system that is able to prove the property. *Hint*: it is tied to the physical dynamics.

(c) To simplify, let

- $Pre \equiv vel > 0 \ \wedge \ pos < station \ \wedge \ T > 0 \wedge acc = \underline{\quad\quad}$
- $\alpha \equiv t := 0; pos' = vel, vel' = acc \;\&\; v \geq 0 \wedge t < T$

Rewriting the above formula, we obtain $Pre \to \big[(\alpha)^* \,; ?(vel = 0)\big]\, pos = station$, which we will try to prove.

$$\to_R\ \dfrac{[;]\ \dfrac{?\ \dfrac{?}{Pre \vdash \big[(\alpha)^*\big]\,[?(vel = 0)]\,pos = station}}{Pre \vdash \big[(\alpha)^* \,; ?(vel = 0)\big]\,pos = station}}{\vdash Pre \to \big[(\alpha)^* \,; ?(vel = 0)\big]\,pos = station}$$

Which rule would you apply next? Give a brief explanation of why each resulting branch is valid (you do not need to show us the proof).

Hint: Recall that rules can only be applied to the main formula, not to smaller sub-formulas. For example, you can't really apply the test rule to the subformula $[?(vel = 0)]\,pos = station$.

6. **Practice with Proof Rules.** In each of the following, fill in the missing parts to give an instantiation of a given proof rule. In some cases, the name of the most appropriate proof rule to use is not given, and is left to you to fill it in. In each case, make sure that your instantiation is not only syntactically correct, but that the instantiation you chose makes it possible to prove the property.

$$\texttt{cut}\ \dfrac{(PART\ A) \qquad (PART\ B)}{(x^2y \geq 0 \wedge x \geq 0 \wedge z \geq x) \vdash [x := 2x][y := 2y]xy \geq 0}$$

$$\texttt{hide left (aka Weakening or Wl)}\ \dfrac{(PART\ C)}{x^2y \geq 0, x \geq 0, z \geq x \vdash [x := 2x][y := 2y]xy \geq 0}$$

$$(PART\ D)\ \dfrac{(PART\ E) \vdash v = 0}{(\forall x. x^2 = X^2 + 2v) \vdash v = 0}$$

$$\texttt{loop invariant}\ \dfrac{(PART\ F) \qquad (PART\ G) \qquad (PART\ H)}{F \vdash [\alpha^*]G}$$

$$\texttt{loop invariant}\ \dfrac{(PART\ I) \qquad (PART\ J) \qquad (PART\ K)}{\begin{array}{c} B > 0 \wedge b > 0 \wedge T > 0 \wedge a < A \wedge t > 0 \wedge (PART\ L) \\ \vdash [((a := B \cup a := b); x' = v, v' = a \;\&\; t \leq T)^*](v \geq 0) \end{array}}$$

7. **Write a Proof.** Using the proof rules you learned in class, construct a full proof for the following d$\mathcal{L}$ formulas. For your convenience, you can download a tex template for one of them[4], with the first rule application already filled in for you.

   (a) $\forall x.((\forall y.[?(y > 0)]yx > 0) \rightarrow [?(xz > 0)]x > 0)$

   (b) $x \geq 1 \rightarrow [((v := 1 \cup v := 2); x' = v \ \& \ x \geq -1)^*](x \geq 0)$

[4] `http://symbolaris.com/course/fcps14/asst2_template.tex`