Assignment 2: Loops and Proofs Revision 1
15-424/15-624/15-824 Foundations of Cyber-Physical Systems

TAs: Nathan Fulton (nathanfu@cs), Anastassia Kornilova (akornilo@andrew)

Due: **Beginning of class**, Thursday 2/11/16
Total Points: 60

1. **Searching for validation!**
Replace α with appropriate parts of the hybrid program that will make the following d$\mathcal{L}$ formulas valid. If such a program can not exist, briefly explain why. Oh, and it looks like your ':' key is broken, so your HP can not use assignments (i.e $x := 5$)

   (a) $[\alpha]false$
   (b) $[(\alpha)^*]false$
   (c) $x = 5 \rightarrow [\alpha \ \& \ x \leqslant 5]false$
   (d) $\langle \alpha \ \& \ x \geqslant 10\rangle x \geqslant 10 \leftrightarrow \langle x' = v, v' = 1; ?x \geqslant 10\rangle x \geqslant 10$
   (e) $[\alpha]x > 1 \leftrightarrow [\alpha]x > 2$
   (f) $[t := 50; t' = 1 \ \& \ t \leqslant 10; \alpha]t = 10$
   (g) $[y := 1]\phi \leftrightarrow [z' = 1]\phi$ (For this question replace ϕ with a formula)

2. **Semantics: What does it actually mean?**

There are at least two styles if giving a meaning to a logical formula. One way is to inductively define a satisfaction relation $\models$ that holds between a state ω and a d$\mathcal{L}$ formula ϕ, written $\omega \models \phi$, whenever the formula ϕ is true in the state ω. Its definition will include, among other cases, the following:

$$\begin{array}{lll} \omega \models P \wedge Q & \text{iff} & \omega \models P \text{ and } \omega \models Q \\ \omega \models \langle\alpha\rangle P & \text{iff} & \nu \models P \text{ for some state } \nu \text{ such that } (\omega,\nu) \in [\![\alpha]\!] \\ \omega \models [\alpha]P & \text{iff} & \nu \models P \text{ for all states } \nu \text{ such that } (\omega,\nu) \in [\![\alpha]\!] \end{array}$$

The full defnінition of these semantics was given in the slides of leacture 5[1].

The other way is to inductively define, for each d$\mathcal{L}$ formula ϕ, the set of states, written $[\![\phi]\!]$, in which ϕ is true. Its definition will include, among other cases, the following:

$$\begin{array}{lll} [\![P \wedge Q]\!] & = & [\![P]\!] \cap [\![Q]\!] \\ [\![\langle\alpha\rangle P]\!] & = & [\![\alpha]\!] \circ [\![P]\!] = \{\omega \ : \ \nu \in [\![P]\!] \text{ for some state } \nu \text{ such that } (\omega,\nu) \in [\![\alpha]\!] \\ [\![[\alpha]P]\!] & = & [\![\neg[\alpha]\neg P]\!] = \{\omega \ : \ \nu \in [\![P]\!] \text{ for all states } \nu \text{ such that } (\omega,\nu) \in [\![\alpha]\!] \end{array}$$

The full definition in this style was given in the lecture notes for lecture 4[2].

   (a) Prove that both styles of defining the semantics are equivalent. That is $\nu \models P$ iff $\nu \in [\![P]\!]$ for all states ν and all d$\mathcal{L}$ formulas P.

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

In recitation, you helped us define the semantics as a function $R(\alpha) : S \rightarrow 2^S$ 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!

[1]http://symbolaris.com/course/fcps16/05-dynax-slides.pdf
[2]http://symbolaris.com/course/fcps16/04-contracts.pdf

(b) 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 $[\![\alpha]\!]$ or R semantics to define the ζ semantics.

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

3. **The sound of soundness proofs** Give a proof of soundness for the following axioms. Focus on using the definitions of the semantics of hybrid programs and formulas - and use the original $[\![\alpha]\!]$ only.

   (a) $([*]) : \quad [(\alpha)^*]\phi \leftrightarrow \phi \wedge [\alpha][(\alpha)^*]\phi$

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

   (c) $\quad [x' = \theta \& Q]P \leftrightarrow \forall t \geqslant 0(([x := y(0)]Q \wedge [x := y(t)]Q) \to [x := y(t)]P)$

   Hint: Use the semantics of hybrid programs, e.g. statements of the form $(\nu, \omega) \in [\![\alpha]\!]$ and the semantics of d$\mathcal{L}$ formulas, 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 $\nu \models \phi_2$.

   (d) Are the following formulas equivalent? Explain your answer (intuitively – a proof is not required!)

   $$[x' = \theta \& Q]P \leftrightarrow \forall t \geqslant 0(([x := y(0)]Q \wedge [x := y(t)]Q) \to [x := y(t)]P)$$

   $$[x' = \theta \& Q]P \leftrightarrow \forall t \geqslant 0((\forall 0 \leqslant s \leqslant t[x := y(s)]Q) \to [x := y(t)]P)$$

4. **Inevitability of 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.

   $$vel > 0 \;\wedge\; pos < station \;\wedge\; T > 0 \wedge acc = \_\_\_\_\_ \;\to$$

   $$\left[\left(t := 0; pos' = vel, vel' = acc \;\&\; v \geqslant 0 \wedge t < T\right)^*\right] vel = 0 \to pos = station$$

   (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!

   (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 $\alpha*$ 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

   - $Pre \equiv vel > 0 \;\wedge\; pos < station \;\wedge\; T > 0 \wedge acc = \_\_\_\_\_$
   - $\alpha \equiv t := 0; pos' = vel, vel' = acc \;\&\; v \geqslant 0 \wedge t < T$

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

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

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.

5. **Practice makes for perfect proofs.** 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.

$$\textit{(PART A)}\ \dfrac{(x^2y \geqslant 0 \land x \geqslant 0 \land z \geqslant x) \vdash y \geqslant 0 \qquad \textit{(PART B)}}{(x^2y \geqslant 0 \land x \geqslant 0 \land z \geqslant x) \vdash [x := 2x][y := 2y]xy \geqslant 0}$$

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

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

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

$$\texttt{loop invariant}\ \dfrac{\textit{(PART I)} \qquad \textit{(PART J)} \qquad \textit{(PART K)}}{B > 0 \land T > 0 \land a < A \land t > 0 \land \textit{(PART L)} \vdash [((a := B \cup a := 0); x' = v, v' = a \;\&\; t \leqslant T)^*](v \geqslant 0)}$$

6. **Loopholes in loop invariants** After getting through the first few questions on this homework, you decided that you love loop-invariants so much, you will teach them to your friend. You start of with a simple system:

$$x \geqslant 0 \vdash [(x := 5 \cup x := 3)^*]x > 0$$

Your friend exclaims, the invariant is just $x > 0$!

(a) What is wrong with this invariant? How would you fix it?

You decide to challenge your friend with a more complicated system:

$$x > 0 \land a > 0 \land A > 0 \land v \geqslant 0 \vdash [((a := 0 \cup a := A); x' = v, v' = a)^*]x \geqslant 0$$

Your friend, ever so quick to jump to conclusions, suggests the invariant $x \geqslant 0$.

(b) What is wrong with this invariant? How would you change it?

(c) To show off your skills, you decide to prove the whole formula: (Starting template is availible)

$$(\textit{ind'})\ \frac{\textit{Your Proof Goes Here}}{x \geqslant 0 \land a > 0 \land A > 0 \land v \geqslant 0 \vdash [((a := 0 \cup a := A); x' = v, v' = a)^*]x \geqslant 0}$$

7. **The one where robots go back and forth** In the previous lab, our robot moved to the charging station and stopped. What if instead we had a robot that moves back and forth between two walls (say one is at 0 and one is at W)? How would we model this system?

   First, let's consider what we want to prove about this model:

   (a) What safety and efficiency conditions would you use?

   (b) What 'continuous dynamics' will you use? (How do you model the motion)

   (c) What controls would the robot use? (Should it be able to accelerate or brake?)

   (d) Using the first three parts of the question, write-out a complete d$\mathcal{L}$ formula to model this situation.

   Note: This question is designed to be a little open-ended, so feel free to make design decisions that you see appropriate. At the same time, accurate models are a crucial part of CPS, so make sure that your formula sets out to prove meaningful properties.