## Assignment 2: Loops and Proofs
## 15-424/15-624/15-824 Foundations of Cyber-Physical Systems
## TA: Brandon Bohrer (bbohrer@cs)

Due: **11:59pm**, Thursday 2/16/17
Total Points: 60

1. **Searching for validation!**
Replace $\alpha$ with a hybrid program that makes each the following d$\mathcal{L}$ formulas valid (in part (c), replace it with an ODE instead of an arbitrary hybrid program). If such a program cannot 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) $x = 5 \rightarrow [\alpha \mathrel{\&} x \leqslant 5]false$

   (d) $\langle \alpha \mathrel{\&} x \geqslant 10 \rangle x \geqslant 10 \leftrightarrow \langle x' = v, v' = 1; ?x \geqslant 10 \rangle x \geqslant 10$

   (e) $[\alpha]x > 1 \leftrightarrow [\alpha]x > 2$

   (f) $[t := 50; t' = 1 \mathrel{\&} t \leqslant 10; \alpha]t = 10$

   (g) $[y := 1]\phi \leftrightarrow [z' = 1]\phi$ (For this question replace $\phi$ with a formula)

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

There are several equivalent ways to define the meaning of a logical formula. One way is to inductively define truth as a binary relation $R(\omega, \phi)$ over a state $\omega$ and $\phi$ which is true exactly when $\phi$ is true in state $\omega$. In Lecture 4, we used the notation $\omega \in [[\phi]]$ for this relation. In Recitation 2, we also used the notation $\omega \models \phi$ for the same relation, which emphasizes that the Lecture 4 semantics do not compute the set of true states $[[\phi]]$, but instead they take in a state $\omega$ and formula $\phi$ and tell us whether $\phi$ is true in $\omega$. For this assignment, you too should use the $\omega \models \phi$ notation, to avoid confusion with the Lecture 5 semantics below.

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 definition of these semantics was given in the slides of Lecture 4[1].

The second way is to inductively define, for each d$\mathcal{L}$ formula $\phi$, the *set* of states, written $[\![\phi]\!]$, in which $\phi$ is true. Whereas the previous definition was a binary relation, this is a unary function from formulas to sets of states.

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 5[2].

[1] http://symbolaris.com/course/fcps17/04-contracts.pdf
[2] http://symbolaris.com/course/fcps17/05-dynax-slides.pdf

(a) Prove that both styles of defining the semantics are equivalent. That is $\nu \models P$ iff $\nu \in [\![P]\!]$ for all states $\nu$ and all d$\mathcal{L}$ formulas $P$. You may simplify your proof by combining symmetric cases, but please be explicit when you do this and justify informally why the cases are symmetric. The notational similarities might have you confused about the difference between these two version of semantics. The important part of the proof will be to show that the *set* operators used in Lecture 5 are equivalent to the *logical* statements in Lecture 4.

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 $R(\alpha) : S \to \wp(S)$ of the program $\alpha$ and of an initial state $\nu$. It would return the set of states that could be reached from $\nu$ through $\alpha$. 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!

(b) Define the semantics of hybrid programs as a function $\zeta(\alpha) : \wp(S) \to \wp(S)$, which takes in a set of input states and gives back the set of all output states than are reachable from some (i.e. any) input. Your definition of $\zeta$ must be inductive, i.e. you should not define it in terms of the $[\![\alpha]\!]$ or $R$ semantics. Rather it should be defined in the same style as those 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. In the third axiom, you should assume that $y$ is a *global* solution to $x' = \theta$.

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

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

(c) $([']') : \; [x' = \theta \& Q]\phi \leftarrow (\forall t \; [x := y(t)]\phi)$

*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 $\nu$ be an arbitrary state, and assume it satisfies the formula $\phi_1$, 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 $\phi_2$, and use the semantics in the opposite direction, adding back syntax until you obtain $\nu \models \phi_2$.

(d) The following formulas are different versions of the ODE solve axiom $[']$. Which, if any, are valid? In these axioms, $y$ stands for a global solution to the ODE $x' = \theta$, i.e. a solution which holds for all $t \in [0, \infty)$. Explain your answer (intuitively – a proof is not required!).

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

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

4. **Inevitability of invariants (IT'S FULL OF STARS!)**

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 $T$.

$$vel > 0 \;\wedge\; pos < station \;\wedge\; T > 0 \wedge acc = \_\_\_\_\_ \;\to$$
$$\left[\big(t := 0; pos' = vel, vel' = acc, t' = 1 \;\&\; vel \geqslant 0 \wedge t < T\big)^*\right] vel = 0 \to pos = station$$

(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 $\alpha$ execute. If it holds before $\alpha$ executes, it holds after $\alpha$ 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

- $Pre \equiv vel > 0 \;\wedge\; pos < station \;\wedge\; T > 0 \wedge acc = \underline{\hspace{1.2cm}}$
- $\alpha \equiv t := 0; pos' = vel, vel' = acc, t' = 1 \;\&\; v \geqslant 0 \wedge t < T$

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

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

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 (i.e. the outermost operator in the formula), not to smaller sub-formulas.

5. **Practice makes for perfect proofs.** In each of the following subproblems, you are given a valid sequent. We want you to finish the first step of a proof for that sequent. In some cases we will tell you which rule to use, but the rule takes an argument (e.g. an invariant). In this case make sure to specify what the argument is. In other cases we will not say which rule to use, so make sure to include the name of the rule. 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.

**Spring '17 Note:** For (d) and (e), the variable y was until recently named upper-case X. If you already did the problem with its old name, you do not need to redo the proof - I will know what you mean.

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

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

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

$$\texttt{loop invariant}\;\dfrac{\textit{(PART F)} \qquad \textit{(PART G)} \qquad \textit{(PART H)}}{x = 0 \vdash [(x := x + 1)^*]x \neq -1}$$

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

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. When teaching them to your friend, assume you are using the basic `loop` rule given in lecture (i.e. you do not have any extra magic that might be provided by KeYmaera X):

$$\texttt{loop invariant}\ \frac{\Gamma \vdash J, \Delta \qquad J \vdash [\alpha]J \qquad J \vdash P, \Delta}{\Gamma \vdash [\alpha]P, \Delta}$$

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) After applying the loop rule with this invariant, the proof will eventually break down. Which part breaks down and what do you have to do to 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) After applying the loop rule with this invariant, the proof will eventually break down. Which part breaks down and what do you have to do to fix it?

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

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

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.