Assignment 4: Differential Invariants and Nondeterministic Assignment
15-424/15-624 Foundations of Cyber-Physical Systems
TAs: Nathan Fulton (nathanfu@cs), Anastassia Kornilova (akornilo@andrew)

Due: **Beginning of class**, Tuesday, 3/22/16
Total Points: 60

1. **Easy as π.** In class we have started looking at some more interesting differential equations with curved motion. Use this new knowledge to create a hybrid program which has no transcendental literals or functions (example π, e, sin, cos), but at the end of execution has the exact value of π in a variable named *pi*. Does this mean that we can now use π in hybrid programs? If so, should we? Explain.

2. **Syntactic derivatives.** In lecture 11, the *syntactic derivative* of a formula is defined (Definition 12) as follows:

$$(\theta \leq \eta)' \equiv ((\theta)' \leq (\eta)') \tag{1}$$
$$(\theta < \eta)' \equiv ((\theta)' < (\eta)') \tag{2}$$
$$(\theta \neq \eta)' \equiv ((\theta)' = (\eta)') \tag{3}$$

   (a) Prove that the following slightly relaxed definition for the syntactic derivative of a strict inequality (2) would also give a sound proof rule for differential invariants

$$(\theta < \eta)' \equiv ((\theta)' \leq (\eta)')$$

   Hint: you may assume all other definitions (1)-(3) remain the same.

   (b) Suppose you remove definition (3) so that you can no longer use the differential invariant proof rule for formulas involving $\neq$. Can you derive a proof rule to prove such differential invariants regardless? If so, how? If not, why not?

3. **Valid, satisfiable, or unsatisfiable.** For each of the following, determine whether the statement is valid, satisfiable, or unsatisfiable.

   (a) $\exists x.[a := *]a = x$

   (b) $\exists x.\langle a := * \rangle a = x$

   (c) $\forall x.[a := *]a = x$

   (d) $\forall x.\langle a := * \rangle a = x$

   (e) $[a := *]a = x$

   (f) $\langle a := * \rangle a = x$

   (g) $\langle a := * \rangle (a = x \land a = y)$

   (h) $\langle a := * \rangle (a = x) \land \langle a := * \rangle (a = y)$

(i) $\langle (a := *)^* \rangle (a = x \wedge a = y)$

4. **Lab1 revisited.** In lab1, question 3, you wrote a hybrid program in which a robot accelerates along a straight line for a non-zero duration less than or equal to T, and then decelerates to stop on a charging station. We will now revisit this problem using nondeterministic assignment.

   (a) **Warm-up:** Write a hybrid program using nondeterministic assignment which assigns x to be any real number in the range $[0, A]$.

   For simplicity, you can use the following solution to Lab 1:

```
Problem.
(pos < station & vel = 0 & T > 0)
->
[
   t := 0;
   acc :=  (station-pos)/ (T*T);
   {pos' = vel, vel' = acc, t' = 1 & vel >= 0 & t <= T};
   ?(t > 0);
   acc :=  -(vel^2 / (2*(station - pos)));
   {pos' = vel, vel' = acc , t' = 1 & vel >= 0}
]
( pos<=station & (vel^2 + 2 * acc * (station - pos))=0)
End.
```

   (b) Rewrite this hybrid program using guarded nondeterministic assignment for the first choice of acceleration. Your guards should allow *all* safe choices of acceleration, thus defining a safety envelope within which all control choices are safe.

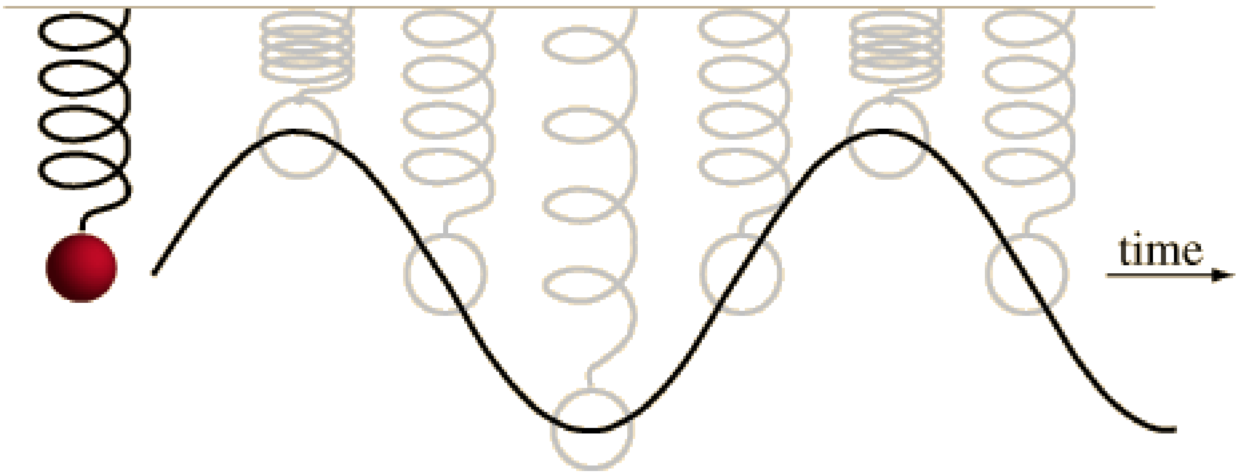
   (c) What are the pros and cons of changing the hybrid program from lab1 to use guarded nondeterministic assignment?

   (d) *Suppose* you were to prove the safety and efficiency properties from lab1 for this new hybrid program. Is this new theorem stronger than the one you proved in lab1 (in other words, would this theorem imply the original)? Or is the old theorem stronger? Or neither? Explain.

5. **Quantum's Adventures Part 2: Finding Harmony** Modeling is a crucial part of CPS design. If we write down unsuitable models we get unsuitable CPSs which will lead us to ultimiately meaningless proofs. This question will give you the opportunity to exercise and sharpen your modeling skills. Consider the following scenario:

   Quantum loves to bounce, but sometimes he gets lonely. Luckily, he is friends with a spring named Harmony that hangs all day from the ceiling. When they hang out, Quantum likes to hold on to the end of the spring and bounce. However, being a careful little ball, he is worried about running into the ceiling or the floor. Luckily hybrid programs can come to the rescue!

Let's model Quantum and Harmony's movement. This set-up can be visualized as follows:



The main force that describes the motion of a spring is called the spring force, and it is described with Hooke's Law as:

$$F = -k\Delta$$

where k is the spring force (some constant) and Δ is the initial displacement from resting position. Applying Newton's Second law of motion, we get the following differential equation for this situation:

$$\frac{d^2x}{dt^2} + \frac{k}{m}x = 0$$

(a) First, define the safety conditions that you would use: (safety first!)

(b) Next, define the ODEs of motion:

(c) Overall, this is a fairly simple model, it doesn't even have any controls! Now, come up with 2-3 ways this model can be expanded (feel free to be creative and incorporate new forces or sources for control).

(d) Now, pick one of these ideas and design a hybrid program to model it. If you can only come up with a partial model, that is okay, but explain what you tried and

the difficulties you ran into.
The goal of this question is to give you practice with open-ended models which you will need to build in the course project. So, do try to experiment!