15-819N/18-879L Logical Analysis of Hybrid Systems

Assignment 4 ($\sum 60$) due 03/31/11 in class

André Platzer

Carnegie Mellon University, Computer Science Department, Pittsburgh, PA

Disclaimer: No solution will be accepted that comes without an **explanation**! Exercises 3–4 are meant as alternative exercises to choose from. By solving both Exercises 3 and 4, you can get extra credit or catch up on missing points for previous exercises.

Exercise 1 Worst-Case Scenario Verification (6p)

Is there a system in which the worst-case behavior is safe, but the system itself is unsafe? By a worst-case behavior, we mean control choices that, when executed, would violate the safety property as quickly as one can imagine. If the system is safe in the particular scenario of this worst-case behavior (e.g., because the system controllers initiate appropriate countermeasures), then a common conclusion is that the system is safe. Is this conclusion a valid conclusion? Prove or disprove.

Exercise 2 Proving with Differential Dynamic Logic (16p)

KeYmaera[4] is a verification tool for hybrid systems presented in the lecture. Turn the option **Differential Saturation** in the *Hybrid Strategy* tab to **off**. Then prove the following example in KeYmaera. Send in the .key problem file and the saved proof file. Indicate which Mathematica version you have used, preferably Mathematica 8. Please explain the structure of your proof works and describe all steps in the proof that you did interactively.

1. Prove the following example in KeYmaera

```
\problem {
  \[ R h,v,t; R c,g,H,V \] (
    h=0 & v=16
  ->
   \[t:=0;
    (
      {h'=v,v'=-10,t'=1, h>=0};
      if (t>0&h=0) then
        v := -v/2; t:=0
      fi
    )*\] (0<=h&h<=13)
  )
}
```

[4] http://symbolaris.com/info/KeYmaera.html

2. For the following example, identify requirements on the free parameters that make the following property hold and prove it in KeYmaera

   ```
   \problem {
     \[ R x,v,g,r,d,H \] (
       \[(
       if (x=0) then
         v := -r*v
       fi;
       ({x'=v, v'=g+d*v^2, v<=0,x>=0}
        ++ {x'=v, v'=g-d*v^2, v>=0,x>=0})*
       )*
       \] (x<=H)
     )
   }
   ```

** Exercise 3 Hybrid Systems Modeling and Verification (38p)

TCAS (Traffic Collision and Avoidance System) is an onboard unit installed on aircraft. It is responsible for detecting upcoming possible collisions and for giving resolution advisories (RA) to prevent them. Possible RA consist of either climbing or descending actions.

1. Develop a hybrid program modeling (simplified) TCAS.
2. Specify desirable properties of the TCAS system in differential dynamic logic.
3. Prove an interesting safety property (e.g., collision freedom) of simplified TCAS in KeYmaera. Identify and explain constraints that ensure safety with respect to the property (equivalent characterizations are not required but reasonable overapproximations are perfectly fine). Send in the .key problem file and the saved proof file. Explain the proof structure and describe all steps in the proof that you did interactively.
   Hint: At all times are you allowed to simplify TCAS. But explain why your simplification is actually helpful and argue why/under what circumstances your simplifications are adequate. For instance, simplify TCAS by assuming that the continuous flows of the system are linear functions and so on. You should further work with a series of increasingly more complicated models starting from a simple one.

** Exercise 4 Hybrid Systems Modeling and Verification (38p)

Consider a robot that moves on planar ground. Suppose the floor has a (small and constant) number of dangerous areas that can be described in terms of simple shapes like rectangles, circles and/or ellipses. Devise a controller that allows the robot to move around freely according to some planning objective but always avoids the dangerous areas.

1. Consider the plan and how to achieve it as a set-value input for the controller. Develop a hybrid program modeling the robot system.
2. Specify desirable properties of the robot system in differential dynamic logic.
3. Prove an interesting safety property (including avoidance of dangerous areas) of the robot system. Identify and explain constraints that ensure safety with respect to the property (equivalent characterizations are not required but reasonable overapproximations are perfectly fine). Send in the .key problem file and the saved proof file. Explain the proof structure and describe all steps in the proof that you did interactively.

   Hint: At all times are you allowed to simplify the robot system, e.g., moving only horizontal/vertical. But explain why your simplification is actually helpful and argue why/under what circumstances your simplifications are adequate.
4. How could you incorporate the planning objectives into the system and its proofs?

** Exercise 5 Hybrid Systems Verification (__p)

EXTRA POINTS: This is an extra assignment. By solving this exercise, you can get extra credit or catch up on missing points for previous assignment.

1. For some number n of your choice, model n hybrid systems of your choice as hybrid programs that have not been modeled in KeYmaera before.
2. Specify desirable properties of the hybrid system in differential dynamic logic.
3. Prove an interesting safety property (including avoidance of dangerous areas) of the robot system. Identify and explain constraints that ensure safety with respect to the property (equivalent characterizations are not required but reasonable overapproximations are perfectly fine). Send in the .key problem file and the saved proof file. Please explain the structure of your proof works and describe all steps in the proof that you did interactively.