LFCPS Exercise Class 3
Verifying CPS with KeYmaera X

General info

Starting next week on Wednesday, November 27th, we will begin offering a consultation hour for the exercises and quizzes from 16:00 to 17:00. Attendance is not mandatory; it's simply an additional opportunity for those who cannot attend the recitations or have further questions. We will meet in my office, Room 159. If you'd like to join, please email me at jonathan.hellwig@kit.edu by Tuesday evening so I can plan accordingly.

Quiz review

Exercise 1. *Identify all axioms applicable to the top-level operator or its subformulas in the given formulas. For each case, specify the relevant axioms:*

(i) $[accel \cup brake; drive]\, x < m$

(ii) $[v := v + 1 \cup v := v - 1; a := a + 1 \cup a := a - 1][x' = v, v' = a]\, x \geq 0$

Exercise 2. *Prove or disprove soundness of the following axioms:*

(i) $([\alpha]P \leftrightarrow [\alpha]Q) \rightarrow [\alpha](P \leftrightarrow Q)$

(ii) $[\alpha](P \leftrightarrow Q) \rightarrow ([\alpha]P \leftrightarrow [\alpha]Q)$

(iii) $[\alpha]\neg P \rightarrow \neg[\alpha]P$

Loop Invariants

Exercise 3. *Prove that every loop is its own invariant.*

$$[\alpha^*]\varphi \rightarrow [\alpha^*][\alpha^*]\varphi$$

Exercise 4 (Finding loop invariants)**.** *Identify loop invariants for the following hybrid programs:*

1. $x > 2 \wedge y \geq 1 \rightarrow [(x := x + y; y := y + 2)^*]x > 1$
2. $x = -1 \rightarrow [(x := 2x + 1)^*]x \leq 0$
3. $x = -1 \rightarrow [(\{x' = 2\})^*]x \geq -5$
4. $x \geq 1 \wedge v \geq 0 \rightarrow [\{x' = v, v' = 2\}^*]x \geq 0$
5. $x = 1 \wedge u > x \rightarrow [(x := 2; \{x' = x^2 + u\})^*]x \geq 0$

Exercise 5. *Consider a hybrid program α and suppose φ is an invariant of α. Suppose β is obtained from α by replacing all sub-programs $\gamma \cup \delta$ by either γ or β. For example if α were $(y := 1 \cup y := -1); x := x - 1 \cup x := x + 1$ then β could be $x := x + 1$ or $y := 1; x := x - 1$ or $y := -1; x := x - 1$. Show that φ is an invariant of β.*

Conversely suppose φ is an invariant of all possible β obtained in the way described. Show that φ is an invariant of α.

Exercise 6 (Heron)**.** *Prove the following formulas*

1. $(z > 0 \wedge x > 0 \wedge (x - e)^2 = z) \rightarrow [(e := \frac{e^2}{2x}; x := \frac{1}{2}(x + \frac{z}{x}))^*](x - e)^2 = z$
2. $(x = 2 \wedge \varepsilon = \frac{3}{4}) \rightarrow [(\varepsilon := \varepsilon^2; x := \frac{1}{2}(x + \frac{2}{x}))^*]|x - \sqrt{2}| < \varepsilon$

Exercise 7 (A minimal model)**.** *We consider a simple flying robot modeled as a point. It can fly choose to fly forward or backward or ascend or descend. We want to make sure the robot avoids an obstacle located at a fixed point. Write a* dℒ *model, design a safe controller and prove its safety. Next week you will extend this model and prove more interesting things in KeYmaeraX.*

KeYmaeraX

Exercise 8. *Prove the following* dℒ *formulas in KeYmaeraX:*

(i) Event-triggered ping pong ball:

```
ArchiveEntry "LFCPS Ex. 8.1 An Event-Triggered Ping Pong Ball"
  Definitions
      Real g; /* gravity constant */
      Real f; /* paddle damping factor */
      Real c; /* ground damping factor */
  End.
  ProgramVariables
      Real x; /* height of ping pong ball */
      Real v; /* velocity of ping pong ball */
  End.
  Problem
      (g > 0 & f > 0 & c > 0 & 1 > c & x = 5 & v <= 0) ->
      [
          {
              {
                  {x' = v, v' = -g & x >= 0 & x <= 5}
                  ++
                  {x' = v, v' = -g & x >= 5}
              }
              ?x = 0; v := -c * v; ++ ?x != 0;
              {
                  ?(x >= 4 & x <= 5 & v >= 0);
                  v := -f * v;
                  ++
                  ?(x < 5);
              }
          }*@invariant(0 <= x & x <= 5 & (x = 5 -> v <= 0))
      ] (0 <= x & x <= 5)
  End.
End.
```

(ii) Time-triggered ping pong ball:

```
ArchiveEntry "LFCPS Ex 8.2 Time-Triggered Ping-Pong Ball"
  Definitions
      Real H; /* initial height */
      Real g; /* gravity */
      Real c; /* damping coefficient */
      Real f; /* paddle factor */
  End.
  ProgramVariables
      Real x, v; /* height, velocity */
      Real t; /* time */
  End.
  Problem
      (2*x = 2*H - v^2 & 0 <= x & x <= 5 & v <= 0) &
      (g = 1 & g > 0 & 1 = c & c >= 0 & 1 = f & f >= 0) ->
      [
          {
              {
                  ?x = 0;
                  v := -c * v;
                  ++
                  ?x != 0;
              }
              {
                  ?((x > 5 + 1/2 - v | 2 * x > 2 * 5 - v^2 & v < 1) & v >= 0);
                  v := -f * v;
                  ++
                  ? !((x > 5 + 1/2 - v | 2 * x > 2 * 5 - v^2 & v < 1) & v >= 0);
              }
              t := 0;
              {x' = v, v' = -g, t' = 1 & x >= 0 & t <= 1};
          }*@invariant(2 * x = 2 * H - v^2 & (x >= 0 & x <= 5))
      ](0 <= x & x <= 5)
  End.
End.
```