Assignment 1: Introduction to Hybrid Programs
15-424/15-624/15-824 Foundations of Cyber-Physical Systems
TAs: Nathan Fulton (nathanfu@cs), Anastassia Kornilova (akornilo@andrew)

1. **Terms, formulas, hybrid programs, oh my!**
   For each of the following, determine if the expression is a dℒ term, a well-formed dℒ formula, a well-formed hybrid program, or none of the above (ie. it is not well-formed). In the case that the expression is none of the above, give a short explanation.

   (a) $z := x^5$

   (b) $?(x > \frac{3}{4})$

   (c) x

   (d) $z := \pi$

   (e) 42

   (f) $[g := 42]$

2. **Evolve Nondeterministically!**
   This question will test your understanding of non-deterministic evolution.

$$\beta \equiv \quad x := x_0;\ v := v_0;\ t := 0;\ (x' = v, v' = a, t' = 1 \ \&\ v \geq 0); ?(v = 0)$$

   (a) Assume that $a < 0 \wedge v_0 \geq 0$. At the end of a run of hybrid program β, what is the value of t as a function of x_0, v_0, and a?

   Let's modify our program a little by removing the test:

$$\beta' \equiv \quad x := x_0;\ v := v_0;\ t := 0;\ (x' = v, v' = a, t' = 1 \ \&\ v \geq 0))$$

   (b) Again assuming that $a < 0 \wedge v_0 \geq 0$, what are the possible values of v at the end of any run of β'?
   What about the possible values of t?

   (c) Suppose we assume instead that $a < 0 \wedge v_0 \leq 0$ (v_0 is **less than** or equal to zero). What are the possible values of v and t at the end of any run of β?

   (d) Let's consider some dℒ formulas that use our program. For each of the following, state whether the formula is valid and give a brief explanation of why: (The antecedent simply ensures that the assumption from the previous parts are true)

      i. $a < 0 \wedge v_0 \geq 0 \rightarrow [\beta]v = 0$

      ii. $a < 0 \wedge v_0 < 0 \rightarrow [\beta]v = 0$

iii. $a < 0 \land v_0 < 0 \rightarrow \langle\beta\rangle v = 0$

iv. $a < 0 \land v_0 \geq 0 \rightarrow [\beta']v = 0$

v. $a < 0 \land v_0 \geq 0 \rightarrow \langle\beta'\rangle v = 0$

3. **Search for the truth!**
   Determine whether each formula is valid/satisfiable/unsatisfiable. If the formula is satisfiable, state for which value it is so. If it is unsatisfiable, briefly explain why.

   (a) $\exists y\ xy = 1$

   (b) $\forall x \exists y\ xy = 1$

   (c) $\forall x \langle x' = c\rangle x > 0$

   (d) $[?x \geq 0; x := -x]x < 0$

   (e) $\langle\{z' = -c\ \&\ z > 0\}; \{z' = c\ \&\ z < 0\}\rangle z = z_0$

4. **Find a Program!**

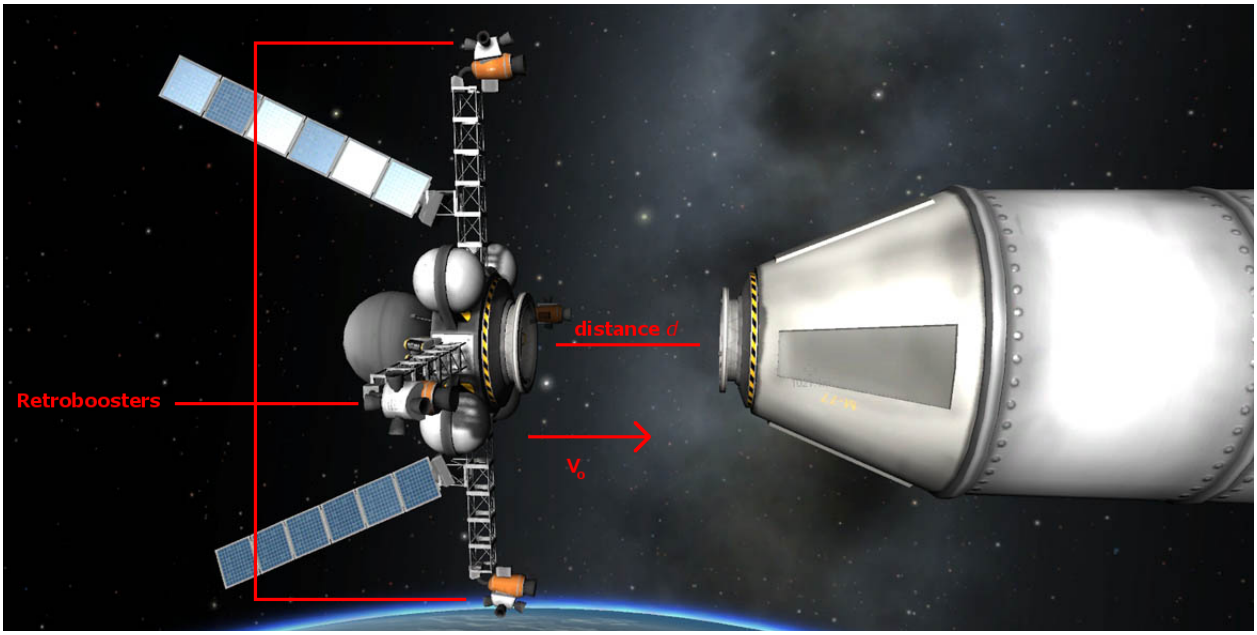
   (a) Write down a program α that makes the formula $\forall z(x = z \rightarrow [\alpha]x > z)$ valid.

   (b) Write down a program α that makes the following formula satisfiable, but not valid: $[\alpha]z > 5$

   (c) Write down a program α that makes the formula $\forall x(\forall n(\langle\alpha\rangle x = n))$ valid. The program may mention x but not n.

5. **Build a Model in SPACE!**

   In theory 0 we considered the problem of docking a lander with a stationary mothership.



Recall that the lander and mothership are perfectly aligned and the lander is already moving toward the mothership with non-zero velocity v_0. We already computed the *acceleration* a_0 that the retro-boosters on the lander should fire with so that the lander's velocity reaches 0 precisely when it reaches the mothership.

(a) Write a hybrid program to model the situation, assuming you (the pilot) knows a_0. Be sure to model all the relevant physical properties. (Hint: Be sure to model the movement of the lander and stopping once it reaches the ship)

(b) A *safety* property is something that a cyber-physical system should *always* maintain. Write down a dℒ formula that expresses the safety property that the lander does not collide with the mothership.

We also want to make sure that our cyber-physical system is efficient. In this case, it means that the lander should not stop before it reaches the mothership (we don't want to be stuck in outer space :/). Write down a dℒ formula that expresses this property.

(c) Let's put together the program and the properties into a system we can later verify in KeYmaera. Don't worry if you don't know how to verify this yet, we will be going over that in Lab 1. For now, fill in the following template:

```
(  )     -> /* Requires: set the known variables here */
[{ ________ }]       /* Continuous dynamics (from part a) */
( ________ )      /* Safety and efficiency conditions (from part b)*/
```