KeYmaera: An In-Depth Look

Nathan Collman, Vincent Maccio, Manivanna Thevathasan

April 8, 2013

Outline

- KeYmaera Overview
- Physics Refresher
- Drag Race Example
- Race Track Example

KeYmaera Overview

- KeYmaera is based on KeY
- Hybrid Systems are systems that exhibit both continuous and discrete dynamic behaviour.
- Differential Dynamic Logic is used to specify and verify these hybrid systems by specifying and verifying correctness properties on the Hybrid Programs.
- KeYmaera is a hybrid verification tool that utilizes Differential Dynamic Logic to specify and verify hybrid systems.

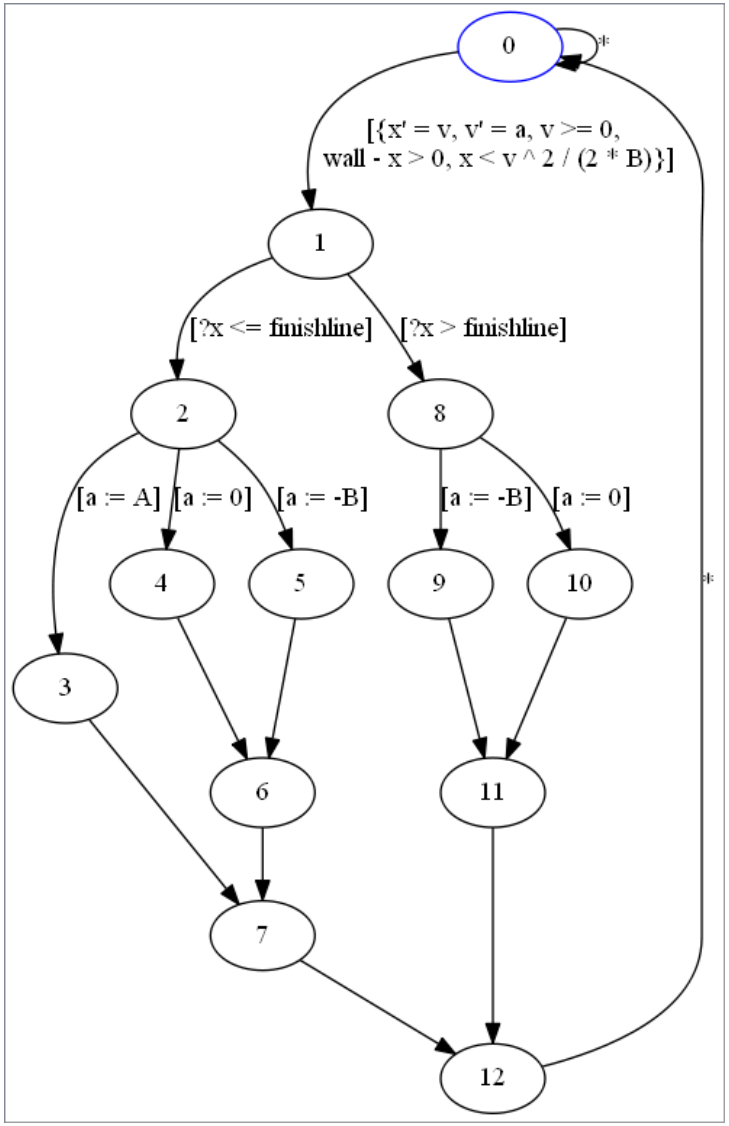
KeYmaera Overview

$$\phi \rightarrow [\alpha]\psi$$

Physics Refresher

1. $v = dx/dt$
2. $a = dv/dt$
3. $v_f^2 = v_i^2 + 2ad$
4. $v_f = v_i + at$
5. $d = v_i * t + \frac{1}{2}at^2$
6. F= ma
7. $F_c = \frac{mv^2}{r}$

Drag Race

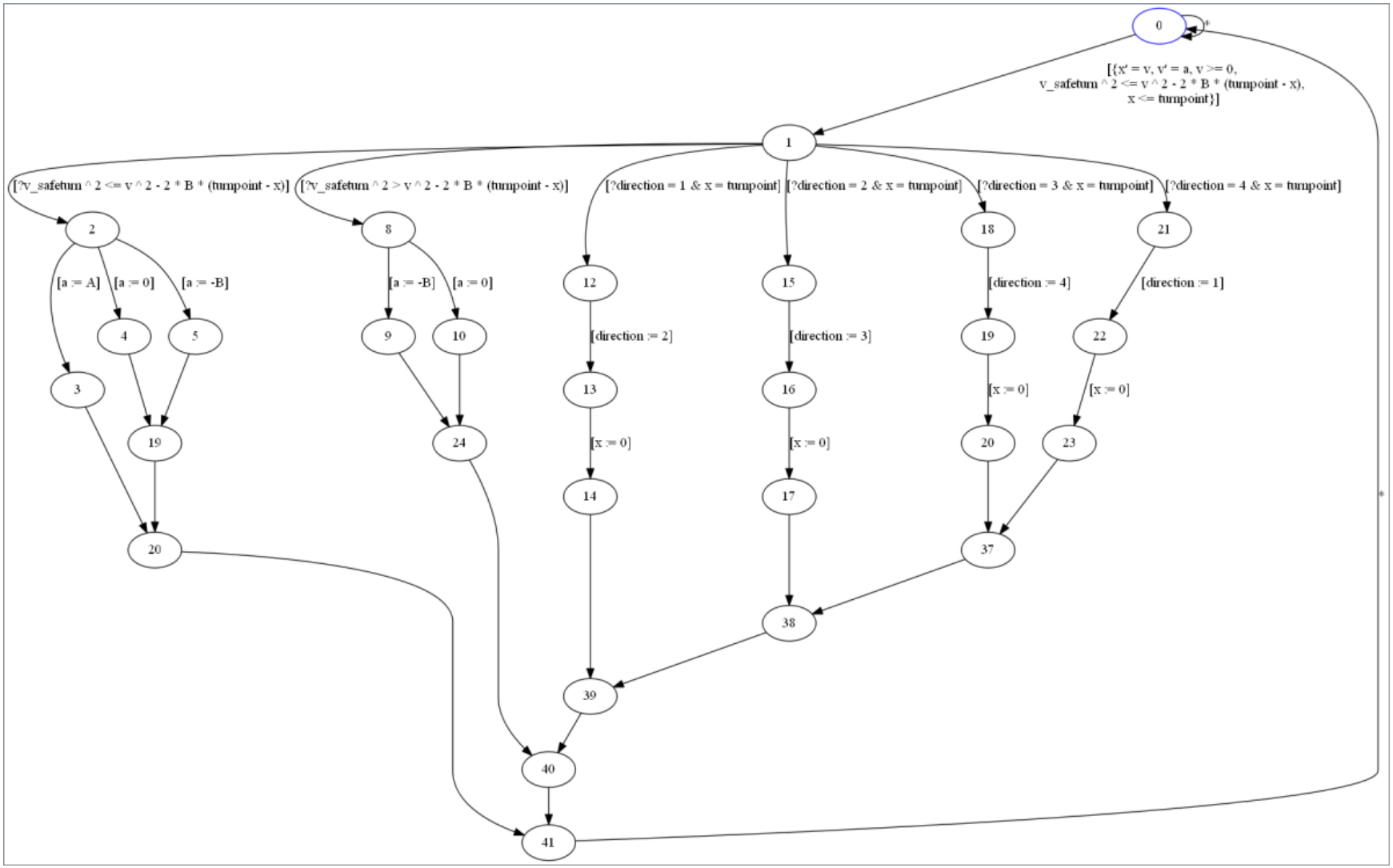

0
*
[{x' = v, v' = a, v >= 0,
wall - x > 0, x < v ^ 2 / (2 * B)}]
1
[?x <= finishline]
[?x > finishline]
2
8
[a := A]
[a := 0]
[a := -B]
[a := -B]
[a := 0]
4
5
9
10
3
6
11
7
12
*

Drag Race

```
\programVariables {
R x, v, a, A, B, finishline, wall;
}

\problem {
x = 0 & v = 0 & a = 0 & A > 0 & B > 0 & finishline > x
& wall > finishline
& A * finishline < B * (wall - finishline)
-> \[
(
   {x' = v, v' = a, v >= 0, (wall - x) > 0, x < v^2/(2 * B)};
   (?x <= finishline; a:= A ++ a:= 0 ++ a:= -B) ++
   (?x > finishline; a:= -B ++ a:= 0)
)*
\] (v >= 0 & x < wall)
}
```

Race Track


0
[{x' = v, v' = a, v >= 0,
v_safetum ^ 2 <= v ^ 2 - 2 * B * (turnpoint - x),
x <= turnpoint}]
1
[?v_safetum ^ 2 <= v ^ 2 - 2 * B * (turnpoint - x)]
[?v_safetum ^ 2 > v ^ 2 - 2 * B * (turnpoint - x)]
[?direction = 1 & x = turnpoint]
[?direction = 2 & x = turnpoint]
[?direction = 3 & x = turnpoint]
[?direction = 4 & x = turnpoint]
2
8
12
15
18
21
[a := A]
[a := 0]
[a := -B]
[a := -B]
[a := 0]
[direction := 4]
[direction := 1]
4
5
9
10
[direction := 2]
[direction := 3]
19
22
3
13
16
[x := 0]
[x := 0]
19
24
[x := 0]
[x := 0]
20
23
14
17
20
37
38
39
40
41

Race Track

```
/*Direction: 1 = North, 2 = East, 3 = South, 4 = West*/
\programVariables {
     R x, v, a, A, B, finishline, wall, direction, turnpoint, v_safeturn;
}

\problem {
x = 0 & v = 0 & a = 0 & A > 0 & B > 0 & direction = 1 & v_safeturn >= 0
& turnpoint > 0
-> \[
(
     {x' = v, v' = a, v >= 0, ((v_safeturn^2) <= v^2 - 2*B*(turnpoint - x)),
     x <= turnpoint};

     (?(v_safeturn^2) > v^2 - 2*B*(turnpoint - x); a:= A ++ a:= 0 ++ a:= -B) ++
     (?(v_safeturn^2) <= v^2 - 2*B*(turnpoint - x); a:= -B) ++
     (?direction = 1 & x = turnpoint; direction:= 2; x:= 0) ++
     (?direction = 2 & x = turnpoint; direction:= 3; x:= 0) ++
     (?direction = 3 & x = turnpoint; direction:= 4; x:= 0) ++
     (?direction = 4 & x = turnpoint; direction:= 1; x:= 0)
)*
\] (x = turnpoint -> v <= v_safeturn)
}
```

KeYmaera Conclusion: What we learned

- KeYmaera can adapt equations used for continuous dynamics based on the given constraints.
- KeYmaera cannot prove properties about systems that have a definite beginning or definite end.
- When specifying systems, we have to ensure that it makes sense for all possible runs.
- KeYmaera has a different notion of the diamond modality.

Bibliography


Jan-David Quesel, Sarah Loos, Nikos Aréchiga, André Platzer. How to Prove Complex Properties of Hybrid Systems with KeYmaera: A Tutorial


André Platzer and Jan-David Quesel. KeYmaera: A hybrid theorem prover for hybrid systems.


André Platzer. http://symbolaris.com/info/KeYmaera.html


André Platzer. http://symbolaris.com/logic/dL.html


André Platzer http://symbolaris.com/info/KeYmaera-guide.html