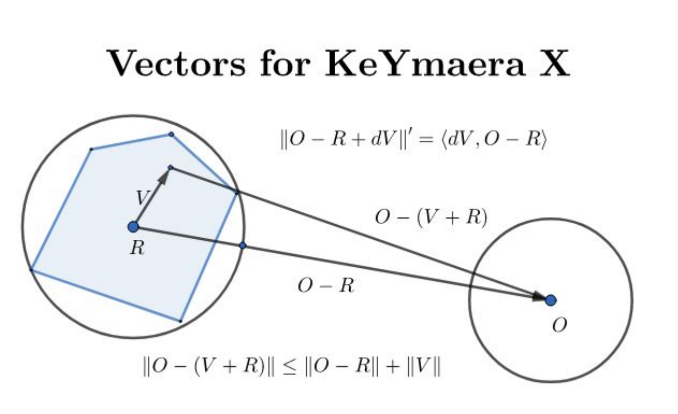
Towards Vector Reasoning in KeYmaeraX



Chris Lambert
cslamber@andrew.cmu.edu

Aditi Kabra
akabra@cs.cmu.edu

Introduction

- Hybrid systems: intersection of discrete and continuous dynamics. Model systems like self driving cars, planes.
- KeYmaeraX: hybrid system prover that uses differential dynamic logic extended with some practical constructions
- Vectors fundamental to physics thinking, would make it easier to express and reason about hybrid systems.
- We discuss implementation possibilities

- Motivation
- Syntax
- Expressivity
- New lemmas
- Implementation Possibilities
    - Syntactic sugar
    - Fixed length lists
    - Inductively-defined structures
- Conclusion

Vectors as first class types

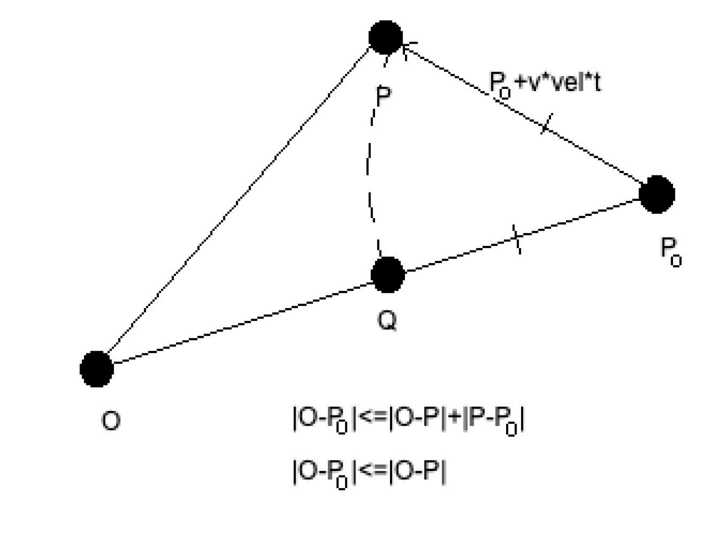
- Many models could be expressed more cleanly
- Classical physics proofs ported over more easily with vector lemmas like triangle inequality and Cauchy Schwarz, that are non-trivial from DL constructs.

Vectors can make models and proofs more intuitive

Can choose any direction

Accelerate only if can stop while traveling along a straight line to obstacle

```
StoppingDistance(vel)<|p-o| ->
[{v:=*; ?|v|=1; t:=0;
    {?StoppingDistance(vel+a*T)<|p-o|-(vel*T+(½)*a*t^2) acc:=a;
     ++ acc:=b;}
{p'=v*vel, vel'=acc, t'=1 & t<=T}
StoppingDistance(vel)<|p-o|}*]
```



Follows from 1D proof + triangle inequality

- Motivation
- Syntax
- Expressivity
- New lemmas
- Implementation Possibilities
  - Syntactic sugar
  - Fixed length lists
  - Inductively-defined structures
- Conclusion

Proposed Syntax

$$v := \{\{a_1, a_2 ... a_n\}\} \,|\, v_n + u_n \,|\, v_n - u_n \,|\, v * a \,|\, -v \,|\, v' \,|\, \mathrm{var}$$

$$a := \; a_{\mathcal{DL}_l} \,|\, v.u \,|\, \mathrm{norm}(v)$$

- Motivation
- Syntax
- Expressivity
- New lemmas
- Implementation Possibilities
    - Syntactic sugar
    - Fixed length lists
    - Inductively-defined structures
- Conclusion

Expressivity

- Can define translation to existing KeYmaeraX language
- Exactly as expressive

- Motivation
- Syntax
- Expressivity
- New lemmas
- Implementation Possibilities
  - Syntactic sugar
  - Fixed length lists
  - Inductively-defined structures
- Conclusion

Target Vector Lemmas

- Cauchy-Schwarz Inequality
  - Surprisingly powerful in proving things about how points move ‘along’ lines or away from other points in ODEs
- Triangle Inequality
- Inner product properties (linear, commutes, etc)
- Derivatives of norm / dot product

- Motivation
- Syntax
- Expressivity
- New lemmas
- Implementation Possibilities
  - Syntactic sugar
  - Fixed length lists
  - Inductively-defined structures
- Conclusion

Syntactic Sugar:
Cheap, but ineffective

```
(X*X)*(Y*Y) >= (X*Y)^2

Vs

(x1*x1+x2*x2+x3*x3)*(y1*y1+y2*y2+y3*y3) >=
(x1*y1+x2*y2+x3*y3)^2
```

- Motivation
- Syntax
- Expressivity
- New lemmas
- Implementation Possibilities
  - Syntactic sugar
  - Fixed length lists
  - Inductively-defined structures
- Conclusion

Fixed-length lists:
Nice, but missing something

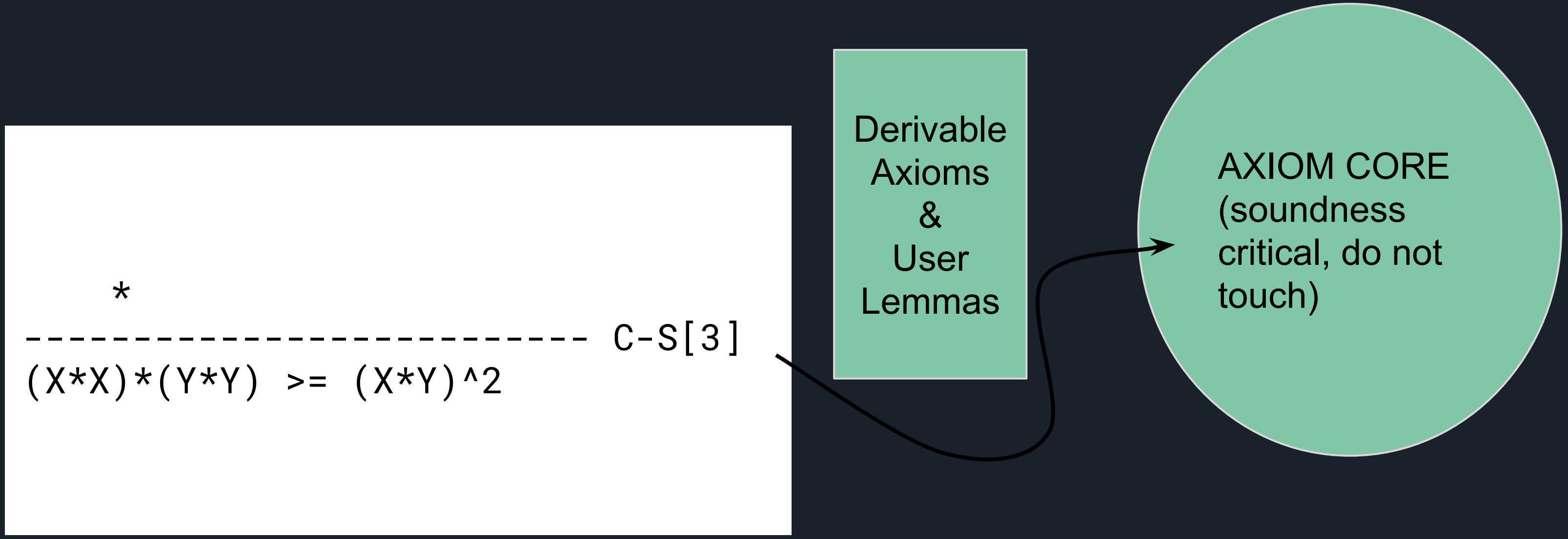
```
New Axioms!

{x,y}*a     = {x*a,y*a}
{x,y}*{a,b} =  x*a+y*b
{x,y}+{a,b} = {x+a,y+b}
...
{x,y,z}*a = {x*a,y*a,z*a}
...
{x,y,z,t,o,O,m,a,n,y} = ...
...
```

```
New Unifier Problems

Is
{x+54,y-12,z*92+x}*{a,b,c} =
similar to the form
{x,y,z}*{a,b,c} = x*a + b*y + c*z
?
```

Fixed-length lists:
Nice, but missing something



- Motivation
- Syntax
- Expressivity
- New lemmas
- Implementation Possibilities
  - Syntactic sugar
  - Fixed length lists
  - Inductively-defined structures
- Conclusion

Inductively-defined semantics: Promising, finite, extensible

```
Vector = [] | <Real> :: <Vector>

(x::y)+(a::b) = (x+y)::(a+b)
(x::y)*(a::b) = (x*y)+ (a*b)
[] + [] = []
[] * [] = []
...

2 axioms / operation = 8 core
```

```
  *
-----------------------------
a+d=x1 & b+e=x2 & c+f = x3
----------------------------- {=}*3
{a+d,b+e,c+f}   = X
---------------------- {+}*3
{a,b,c}+{d,e,f} = X

Easy to chase
```

Inductively-defined semantics: Promising, finite, extensible

User lemmas without modifying the core:

- Prove, potentially using induction
  - Powerful enough to prove triangle inequality and Cauchy-Schwarz
  - Simpler proofs same way
  - Captures important meaning
- Save lemma as tactic
- Use as normal
- Prove to core via repeated application
  - Complex matching axioms only required in tactics
  - Core only needs a *slightly* stronger uniform substitution

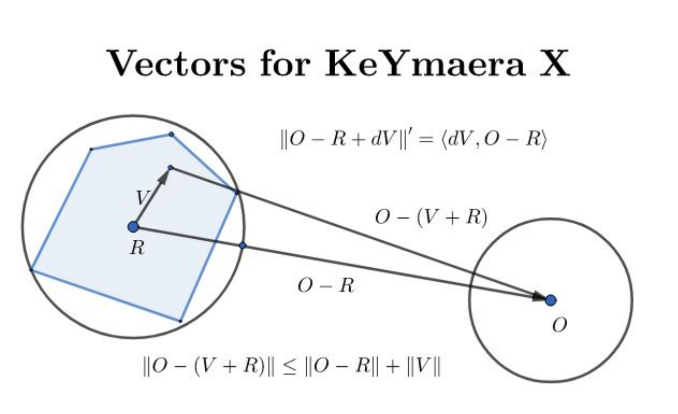
$$
\begin{aligned}
&((a :: x) \cdot (a :: x))((b :: y) \cdot (b :: y)) - ((a :: x) \cdot (b :: y))^2 \\
&= (a^2 + x \cdot x)(b^2 + y \cdot y) - (ab - x \cdot y)^2 \\
&= a^2b^2 + a^2(x \cdot x) + b^2(y \cdot y) + (x \cdot x)(y \cdot y) - a^2b^2 - 2ab(x \cdot y) - (x \cdot y)^2 \\
&\geq a^2b^2 - a^2b^2 + (x \cdot y)^2 - (x \cdot y)^2 + a^2(y \cdot y) + b^2(x \cdot x) - 2ab(x \cdot y) \\
&= (a * y - b * x) \cdot (a * y - b * x) \geq 0
\end{aligned}
$$

- Motivation
- Syntax
- Expressivity
- New lemmas
- Implementation Possibilities
  - Syntactic sugar
  - Fixed length lists
  - Inductively-defined structures
- Conclusion

Conclusion

- Proposed vector types, syntax
- Identified some useful vector lemmas, case studies
- Explored implementation choices with partial subset implementation
- Future work: implement, identify more lemmas and case studies

Thank you



Aditi Kabra
akabra@cs.cmu.edu

Chris Lambert
cslamber@andrew.cmu.edu