Proof Support for Hybrid Systems Verification, IV

Hybrid Systems and other Applications

Lawrence C Paulson, Computer Laboratory, University of Cambridge
Hybrid systems

- A hybrid system has a compound state containing both discrete and continuous parts. The discrete part is a finite-state machine where each state has continuous dynamics.

- Simple examples:
  - room heater with thermostat
  - a water tank with inlet and outlet pipes with valves
  - an autopilot with multiple modes
  - any sort of physical process controlled by a computer
Room heater with thermostat

dynamics given by differential equations

target temperature: 20°C

switches at 19 and 21 to prevent oscillation
1: A collision avoidance problem

- two aircraft, $x$ and $y$, flying in two dimensions (sorry)
- first studied by Platzer (2010), using KeYmaera
- MetiTarski treatment due to Denman, using closed-form solutions of the differential equations
The system of differential equations for aircraft $x$

\[ x_1'(t) = d_1(t) \quad x_2'(t) = d_2(t) \quad d_1'(t) = -\omega d_2(t) \quad d_2'(t) = \omega d_1(t) \]
\[ x_1(0) = x_{1,0} \quad x_2(0) = x_{2,0} \quad d_1(0) = d_{1,0} \quad d_2(0) = d_{2,0} \]

$x_1$ denotes position in the **first** coordinate; $d_1$ denotes velocity.

$x_2$ denotes position in the **second** coordinate; $d_2$ denotes velocity.
... and the closed-form solution

\[
x_1(t) = x_{1,0} + \frac{d_{2,0} \cos(\omega t) + d_{1,0} \sin(\omega t) - d_{2,0}}{\omega}
\]

\[
x_2(t) = x_{2,0} - \frac{d_{1,0} \cos(\omega t) - d_{2,0} \sin(\omega t) - d_{1,0}}{\omega}
\]

Solutions of linear differential equations frequently involve simple combinations of transcendental functions.
Possible paths of the two aircraft
The desired safety property

Two aircraft following those equations...

subject to certain other parameters...

must maintain a safe distance, $p$:

$$(x_1(t) - y_1(t))^2 + (x_2(t) - y_2(t))^2 > p^2$$
The resulting MetiTarski problem

```prolog
fof(airplane_easy, conjecture, 
(! [T,X10,X20,Y10,Y20,D10,D20,E10,E20] : 
  ( 0 < T & T < 10 & X10 < -9 & X20 < -1 & Y10 > 10 & Y20 > 10 & 
    0.1 < D10 & D10 < 0.15 & 0.1 < D20 & D20 < 0.15 & 
    0.1 < E10 & E10 < 0.15 & 0.1 < E20 & E20 < 0.15 ) 
  => 
    ( (X10 - Y10 - 100*D20 - 100*E20 + (100*D20 + 100*E20)*cos(0.01*T) 
      + (100*D10 - 100*E10)*sin(0.01*T))^2 + 
      (X20 - Y20 + 100*D10 + 100*E10 + (-100*D10 - 100*E10)*cos(0.01*T) 
      + (100*D20 - 100*E20)*sin(0.01*T))^2 ) 
    > 2 )
).
include('Axioms/general.ax').
include('Axioms/sin.ax').
include('Axioms/cos.ax').
```
Remarks about this proof

• originally required 924 seconds (using Z3)
• can take as little as 30 seconds, depending on setup
• 9 variables! (the most ever in a MetiTarski proof)

William Denman attended Marktoberdorf in 2012!
2: Model checking by *qualitative abstraction*

- **Over-approximate** the (continuous) transition relation by a *discrete* state transition system

- Denman’s QUANTUM abstracter uses MetiTarski
  - to **remove** infeasible abstract states
  - to **validate** abstract transitions

- It’s based on HybridSal (Tiwari, SRI); the abstract models are checked using SAL (Symbolic Analysis Laboratory)
Model checking alongside theorem proving!

Building the model requires hundreds of MetiTarski runs.

The actual certification comes from model checking.

Failed proofs make the model more complicated, but aren’t fatal.

Contrast with other approaches where a conjunction of properties MUST be proved.
Self-driving car performing a curved turn

one of the 1190 “feasibility” problems

```prolog
fof(stdin, conjecture,  
    (![X,TT,G] : (~(G<1.5 & G>-1.5 & TT>=0 & TT<6 & X + 2>0 & X - 2>0 & X - 0.2*cos(TT - 1.5) + 0.25>0 & X - 0.2*sin(TT) - 0.5>0 & G=0 & G - 0.5>0)))).
```

one of the 2123 “transitions” problems

```prolog
fof(checkTransition, conjecture,  
    (![X,TT,G] : (G<1.5 & G>-1.5 & TT>=0 & TT<6 & X + 2>0 & X - 2>0 & X - 0.2*cos(TT - 1.5) + 0.25>0 & X - 0.2*sin(TT) - 0.5=0 & G>0 & G - 0.5=0 => (-2*sin(G) + 0.2*sin(TT - 1.5) > 0 |  
                 -2*sin(G) + 0.2*sin(TT - 1.5) = 0)))).
```

QUANTUM verifies *unbounded* safety properties, unlike many other hybrid system verifiers.

Other examples include a ball *bouncing on a sine curve* and a system with two water tanks.
3: Error analysis for numerical computation

```prolog
fof(atan_error_analysis_5,conjecture, ! [X] : ((abs(X) <= 1/30) => 
  abs (arctan(X) - 
    (X - 11184811/33554432 * X^3 - 13421773/67108864 * X^5)) 
  <= (1/2)^8 ) ).
```

Bespoke applications could need any precision

Could have 100s of cases, verifying *table entries*

Typically a narrow range is checked

Such problems are trivial for MetiTarski!
But can MetiTarski be trusted? Some issues

- *Arithmetic simplification:* reducing polynomials to canonical form, using ad hoc code

- *Specialised axioms* giving upper or lower bounds of special functions

- Reliance on an external RCF decision procedure

but we do get machine-readable proofs! Could they be validated independently?


Algebraic Simplification

Translation to canonical form

Obvious cancellation laws

\[
\left( \frac{x}{y} \right) \frac{1}{(x + \frac{1}{x})} = \frac{x^2}{y(x^2 + 1)}
\]

Transformation of quotients!

Independent reconstruction in an should be straightforward...
Verifying the axioms

- The bounds for $\sqrt{}$, $\sin$ and $\cos$ already verified in Isabelle (infinite families of bounds).

- No general Isabelle theory of continued fractions yet, but the bounds used in MetiTarski have been verified:
  - For $\exp$ and $\ln$, all including the most complicated
  - For inverse tangent, verified all but two bounds
The weakest link: the decision procedure

- The best-known procedure (*cylindrical algebraic composition*) is complicated and requires efficient computer algebra tools.

- Real quantifier elimination is *doubly exponential* in the number of variables (Davenport and Heintz, 1988)

- Few implementations of any sort exist; fewer justify their answers with any sort of *evidence*. (Mainly in HOL-Light.)

  - *Hörmander’s procedure*: far too slow

  - *Sum-of-squares methods*
Must we trust the decision procedure?

- During search, MetiTarski may call the decision procedure hundreds of times, also to discard redundant clauses.
- We only need to trust calls appearing in the proof, but there could still be dozens!
- These are specific polynomial inequalities, which could be checked by other means (not necessarily deductive).
- There are multiple, independent implementations of CAD.
Summary

Hybrid systems can be verified on the basis of closed-form solutions of their dynamics.

The dynamic behaviour can be specified using transcendental functions.

MetiTarski can also be used to support model checking by qualitative abstraction.

Simple numerical algorithms can also be verified!

The trust issues are significant but tractable.
The Cambridge team

James Bridge

Grant Passmore

Behzad Akbarpour

William Denman

Zongyan Huang
acknowledgements

• Help from C. W. Brown, A. Cuyt, I. Grant, J. Harrison, J. Hurd, D. Lester, C. Muñoz, U. Waldmann, etc.

• EPSRC grant EP/C013409/1, Beyond Linear Arithmetic

• EPSRC grant EP/I011005/1, Automatic Proof Procedures for Polynomials and Special Functions

• And many thanks to the Marktoberdorf administration!