Theory of Reals

for Verification and Synthesis of

Hybrid Dynamical Systems

Ashish Tiwari
Computer Science Laboratory (CSL)
SRI International (SRI)
Menlo Park, CA 94025
Email: ashish.tiwari@sri.com

Cyber-Physical Systems

There is increasing interaction between embedded software/cyber and the physical world

- Aerospace
  - flight control: traditional to adaptive
  - unmanned vehicles
- Automobile
  - powertrain control
  - cooperative adaptive cruise control

How to design, verify, and certify such systems?

Systems Biology

The goal of Systems Biology is to study and understand biological phenomena by building and analyzing dynamic system-level models

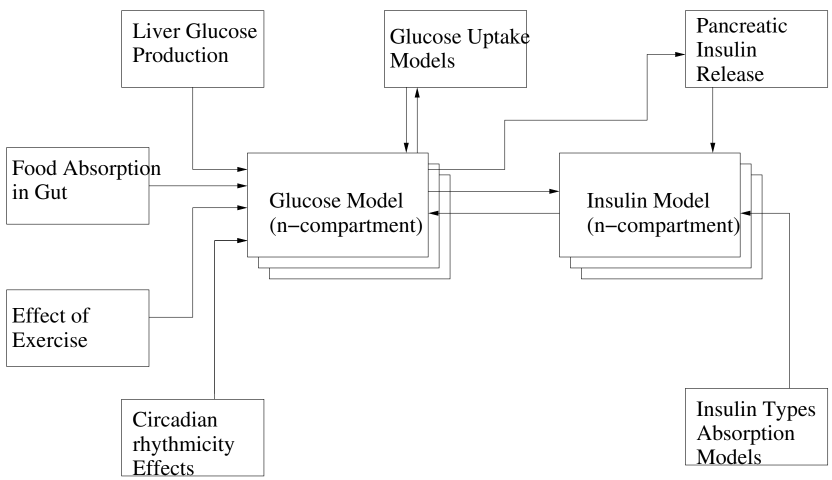
Few examples

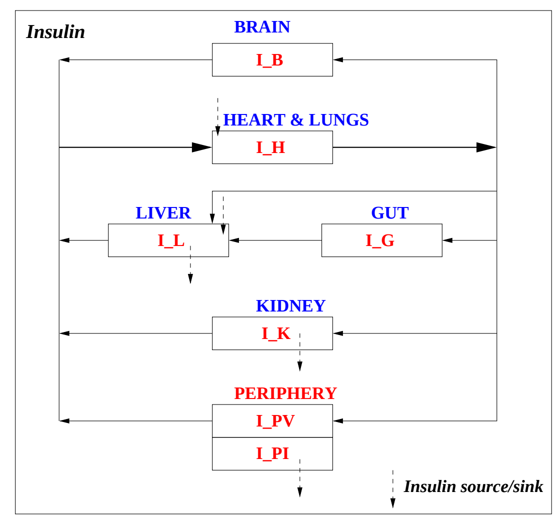
- *Aplysia*: Neural circuitry of the feeding behavior
- *B.Subtilis*: Sporulation initiation network

Symbolic Systems Biology

The goal of Symbolic Systems Biology is to study and understand biological phenomena by building and analyzing dynamic system-level models symbolically

Human Insulin-Glucose Metabolism

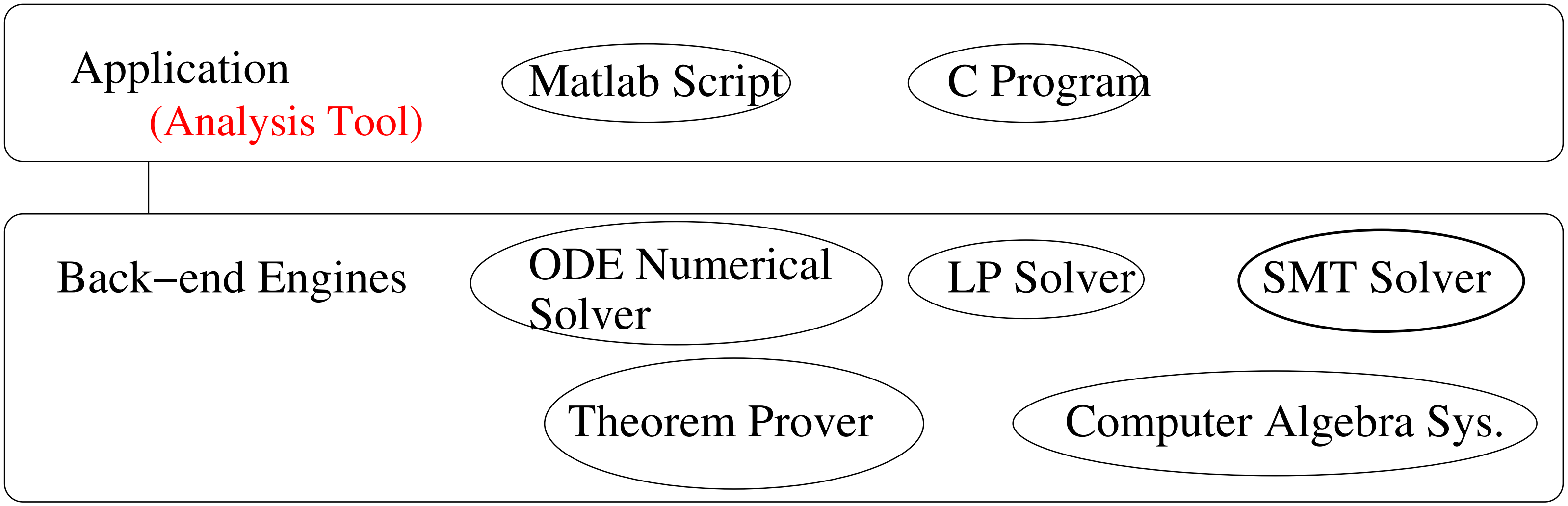




Backend Engines

We need general-purpose symbolic+numeric reasoning engines to enable analysis of these rich models

A popular architecture for building analysis tools



Outline

1. Part I: Why we need symbolic solvers?
2. Part II: What are SMT solvers? How to overcome complexity barriers?
3. Part III: Theory of Reals = Gröbner basis + ?

Part I:

Why we need symbolic solvers?

Safety of Cruise Control

Example. Consider a cruise control:

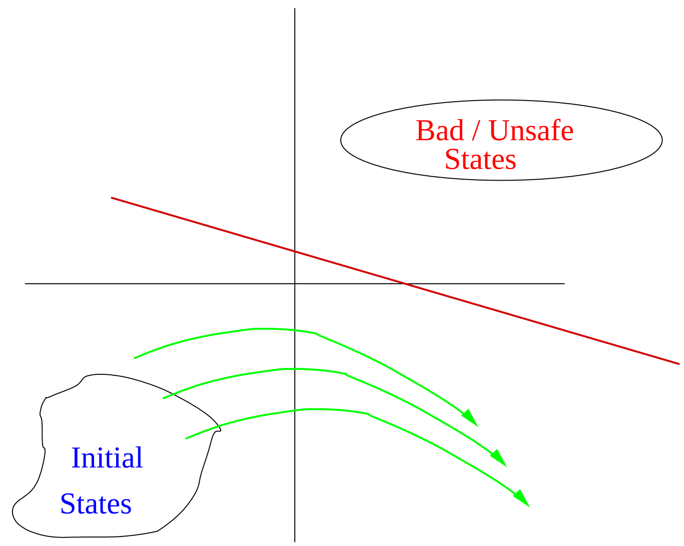
$$
\begin{aligned}
\dot{v} &= a \\
\dot{a} &= -4v + 3v_f - 3a + gap \\
g\dot{a}p &= -v + v_f
\end{aligned}
$$

where v, a is the velocity and acceleration of this car, v_f is the velocity of car in front, and gap is the distance between the two cars.

Suppose we enter the cruise control mode whenever $Init$ holds.
Prove that the cars will not crash.

Invariants / Barriers

We can prove cars will not crash if we can find an invariant set whose boundary separates unsafe states from initial states



Suppose I guess that the invariant is of the form:

$$c_1 v + c_2 v_f + c_3 a + c_4 gap \leq c_5$$

How can I find $c_1, \ldots, c_5$?

Invariants / Barriers

I need to solve:

$$
\begin{array}{rcl}
\multicolumn{3}{l}{\exists c_1, \ldots, c_5 : \forall v, v_f, a, gap :} \\
Init(v, v_f, a, gap) & \Rightarrow & c_1 v + c_2 v_f + c_3 a + c_4 gap \leq c_5 \\
\wedge & & \\
c_1 v + c_2 v_f + c_3 a + c_4 gap = c_5 & \Rightarrow & \frac{d}{dt}(c_1 v + c_2 v_f + c_3 a + c_4 gap) \leq 0 \\
\wedge & & \\
c_1 v + c_2 v_f + c_3 a + c_4 gap \leq c_5 & \Rightarrow & gap > 0
\end{array}
$$

Need backend solvers to decide satisfiability of above.

Dynamical Systems

A lot of engineering and science concerns dynamical systems

- State Space: The set of states, $\mathbf{X}$
  - Discrete: $\mathbf{X}$ is $\mathbb{N}^n$
  - Continuous: $\mathbf{X}$ is $\mathbb{R}^n$
  - Hybrid: $\mathbf{X}$ is $\mathbb{N}^{n_1} \times \mathbb{R}^{n_2}$
- Dynamics: The evolutions, $\mathbf{T} \mapsto \mathbf{X}$
  - Discrete: $\mathbf{T}$ is $\mathbb{N}$
  - Continuous: $\mathbf{T}$ is $\mathbb{R}$
  - Hybrid: $\mathbf{T}$ is $\mathbb{R} \times \mathbb{N}$

These systems can be modeled using differential equations, (Finite) state machines, or hybrid automata.

Typical Properties of Systems

What can we say (deduce, compute) about the model?

- Reachability. Is there a way to get from state $\vec{x}$ to $\vec{x'}$
- Safety. Does the system stay out of a bad region
  - Can the car ever collide with the car in front?
- Liveness. Does something good always happen
- Stability. Eventually remain in good region
- Timing Properties. Something good happens in 10 seconds

Does the model satisfy some property.

Property is described in a logic and evaluated over the semantic structure defined by the formal models.

Verification Problem for Dynamical Systems

- Given a dynamical system
- And a property: safety, reachability, liveness
- Show that the property is true of the model

Approaches:

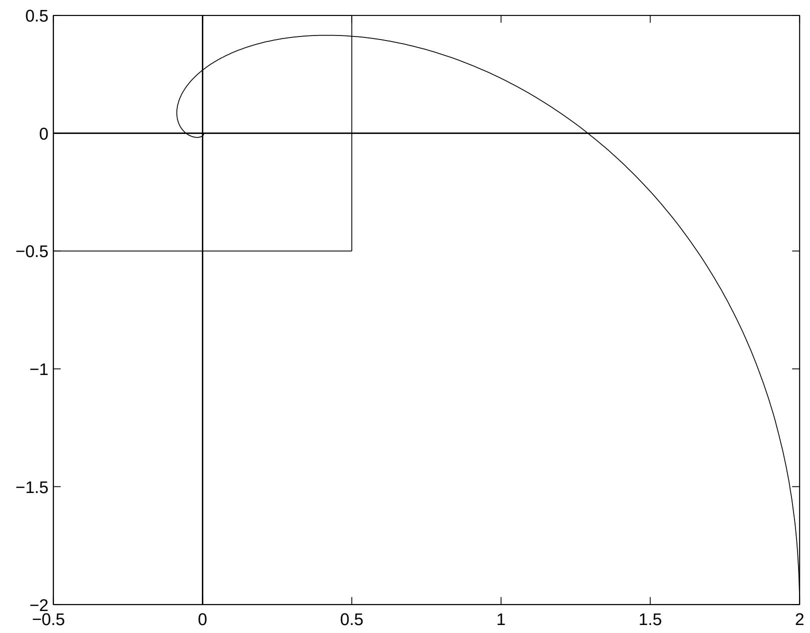
- model checking (MC), bounded MC (BMC), infinite BMC (iBMC)
- deductive verification, k-induction
- Abstract interpretation

Verification by Invariance Checking

Also called Barrier Certificates

Consider the CDS:

$$\frac{dx_1}{dt} = -x_1 - x_2$$
$$\frac{dx_2}{dt} = x_1 - x_2$$



$x_1^2 + x_2^2 \leq 0.5$ is an invariant.

Proof obligation:

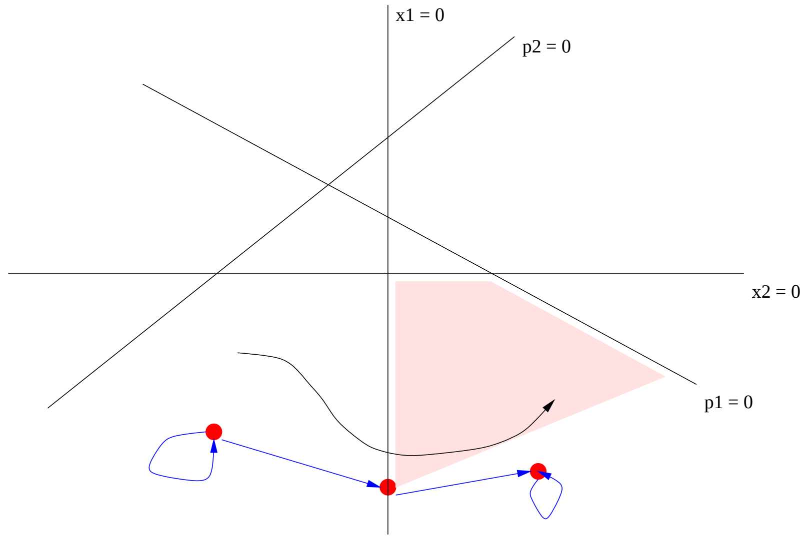
$$\forall x_1, x_2 : x_1^2 + x_2^2 = 0.5 \Rightarrow 2x_1(-x_1 - x_2) + 2x_2(x_1 - x_2) < 0$$

Verification by Abstraction

The Hybrid Abstraction Approach:

Create a finite abstraction of the continuous/hybrid system and model-check it

Consider a system with state space $\Re^2$, partitioned w.r.t signs of x_1, x_2, p_1, p_2:



$\{x_1 = 0, x_2 < 0, p_1 < 0, p2 > 0\} \stackrel{\#}{\Rightarrow} \{x_1 > 0, x_2 < 0, p_1 < 0, p2 > 0\}$ if

$$\exists x_1, x_2 : x_1 = 0 \wedge x_2 < 0 \wedge p_1 < 0 \wedge p2 > 0 \wedge \frac{dx_1}{dt} > 0$$

Verification by Invariant Generation

Consider the system:

$$\frac{dx_1}{dt} = -x_1 - x_2$$
$$\frac{dx_2}{dt} = x_1 - x_2 + x_d$$

Initially: $x_1 = 0, x_2 = 1$

Property: $|x_1| \leq 1$ always

Guess

- Template for witness $W := ax_1^2 + bx_2^2 + c$
- Template for assumption $A := |x_d| < d$

Example Continued

Verification Condition: $\exists a, b, c, d : \forall x_1, x_2, x_d :$

$$\begin{aligned} x_1 = 0 \wedge x_2 = 1 &\quad\Rightarrow\quad W \leq 0 \\ A \wedge W = 0 &\quad\Rightarrow\quad \frac{dW}{dt} < 0 \\ W \leq 0 &\quad\Rightarrow\quad |x_1| \leq 1 \end{aligned}$$

Ask contraint solver for satisfiability of above formula

Solver says: $a = 1, b = 1, c = -1, d = 1$

$$\begin{aligned} x_1 = 0 \wedge x_2 = 1 &\quad\Rightarrow\quad x_1^2 + x_2^2 - 1 \leq 0 \\ |x_d| < 1 \wedge x_1^2 + x_2^2 - 1 = 0 &\quad\Rightarrow\quad 2x_1(-x_1 - x_2) + 2x_2(x_1 - x_2 + x_d) < 0 \\ x_1^2 + x_2^2 - 1 \leq 0 &\quad\Rightarrow\quad |x_1| \leq 1 \end{aligned}$$

This proves that $|x_1| \leq 1$ always.

Stability Verification

Consider the aircraft model:

$$\frac{d\vec{x}}{dt} = f(\vec{x})$$

where $\vec{x}$ is a state vector consisting of airspeed, angle of attack, pitch rate, pitch angle, . . .

Property: System is asymptotically stable

Guess template for Lyapunov function $V := \vec{x}^T A \vec{x}$

Verification Condition:

$$\exists A : \forall \vec{x} : V \geq 0 \wedge (V > 0 \Rightarrow \frac{dV}{dt} \leq 0)$$

Summary So Far

- Formulas in the theory of real-closed fields arise when verifying continuous and hybrid dynamical systems

$$\forall \text{ and } \exists\forall \text{ formulas}$$

- We need embeddable solvers that are
  - incremental and fast,
  - support rich API,
  - generate small unsatisfiable core
- We need practical methods: detect inconsistency of "easy" instances efficiently
- Ideally integrate with Satisfiability Modulo Theory (SMT) solvers

Outline

1. Part I: Why we need symbolic solvers?
2. Part II: What are SMT solvers? How to overcome complexity barriers?
3. Part III: Theory of Reals = Gröbner basis + ?

SMT Solvers

Decide satisfiability modulo theories using symbolic + algebraic techniques!

- Employ a propositional satisfiability solvers for Boolean reasoning
- Employ decision procedures for reasoning over theories
  - rational linear arithmetic: simplex
  - uninterpreted function symbols: congruence closure
  - linear arithmetic over integers
  - theory of arrays
  - theory of bitvectors
  - theory of datatypes

Example: Yices `http://yices.csl.sri.com/`

SMT Solvers: Example

Consider the following constraints:

$$
\begin{array}{rcl}
x > 3 & \vee & x < 1, \\
x < 2 & \Rightarrow & f(y) = 2, \\
x > 2 & \Rightarrow & y = x, \\
f(x) = f(y) & \Rightarrow & x = 0, \\
f(y) > 0 & \Rightarrow & x > 1
\end{array}
$$

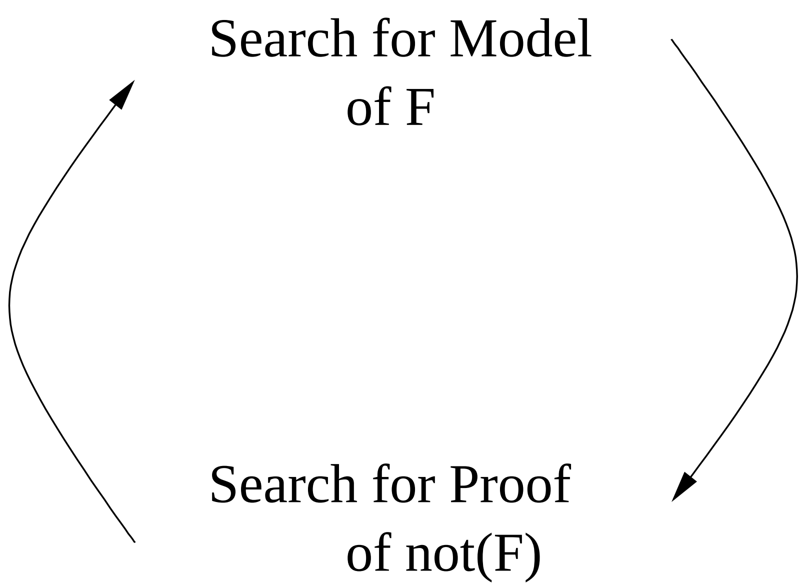
Is there a value for x, y and f such that the above constraints are satisfiable?

SMT solvers can solve such problems – with 1000s of variables and constraints

Why are SMT Solvers So Effective?

SMT is a revolution

Successful combination of model searching and proof searching



The system now learns from failures, making the search feasible

SMT has realized the dream of having embedded deduction

Nonlinear Constraint Solving

SMT solvers currently have limited support for things a computer algebra system can do

Very limited reasoning about nonlinear constraints

Nonlinear constraint solving is essential for analyzing

- complex cyber-physical systems and
- models from systems biology

SMT + CAS : Challenge is to not compromise speed and scalability of SMT solvers

Can we do it? Can we overcome the complexity barrier?

Canonical Application Area: Analysis

Model analysis is the canonical application area for symbolic engines such as SMT solvers

Most important problems in verification are undecidable

- Safety verification of infinite-state systems

and they can not be directly reduced to (decidable) SMT problems

Applications make a choice...

View from the Application Layer

Any application that solves an undecidable problem L, when given an instance ϕ, focuses on either

- showing $\phi \in L$, or
- proving $\phi \notin L$

but not both

A verification tool will target either

- exhibiting an error or
- proving correctness

but not both

View from the Application Layer

Depending on what the application targets, the needs are different

Verification Approach	Commitment	Useful definitive answer
Abstraction	Proving correctness	Proof of not(F)
Invariant Checking	Proving correctness	Proof of not(F)
Bounded Model-Checking	Showing a bug	Model for F

Both SAT and UNSAT answers are useful

But only ONE answer needs to be definitive for soundness claims

Skewing the Symmetry

There is a market for asymmetric tools

```
Tool+(φ):
  Input:  φ
  Output:
     DEFINITELY SAT or
     MAYBE UNSAT
```

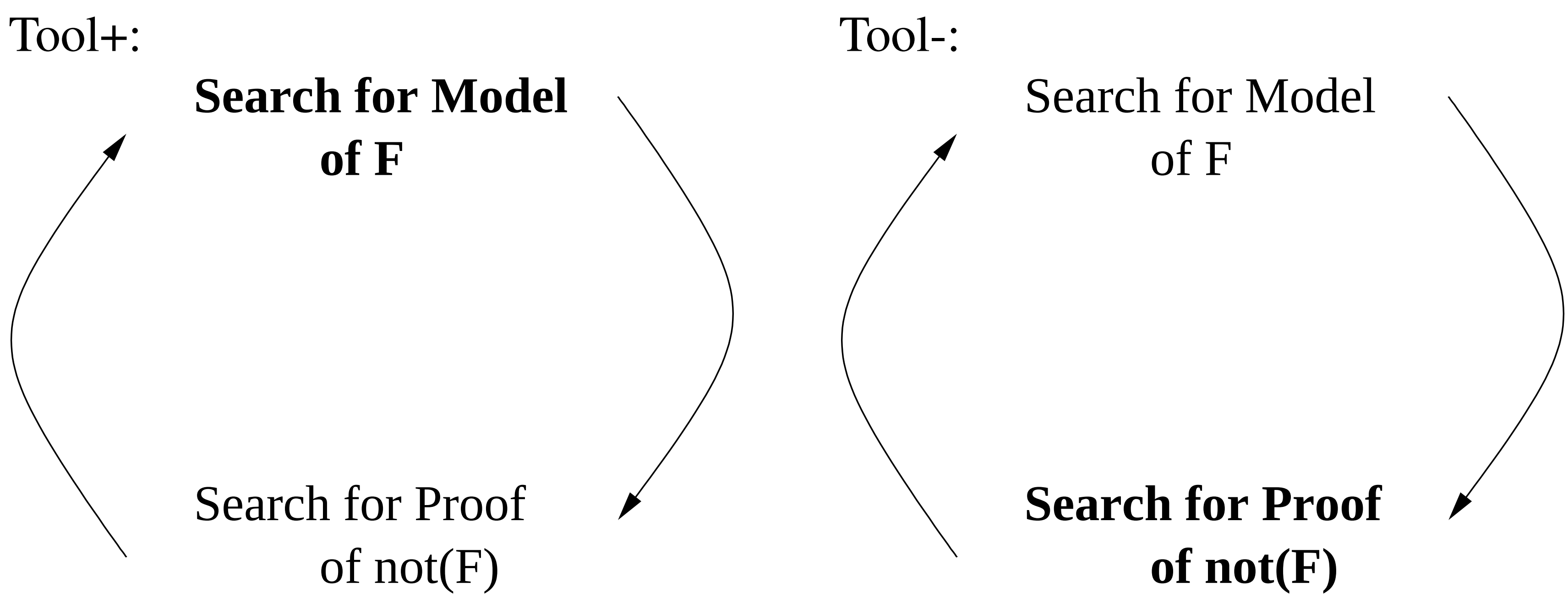
```
Tool-(φ):
  Input:  φ
  Output:
     DEFINITELY UNSAT or
     MAYBE SAT
```

If output = DEFINITELY SAT, then ϕ should indeed be satisfiable

If output = DEFINITELY UNSAT, then ϕ should indeed be unsatisfiable

If output = MAYBE SAT/UNSAT, then nothing can be inferred about ϕ.

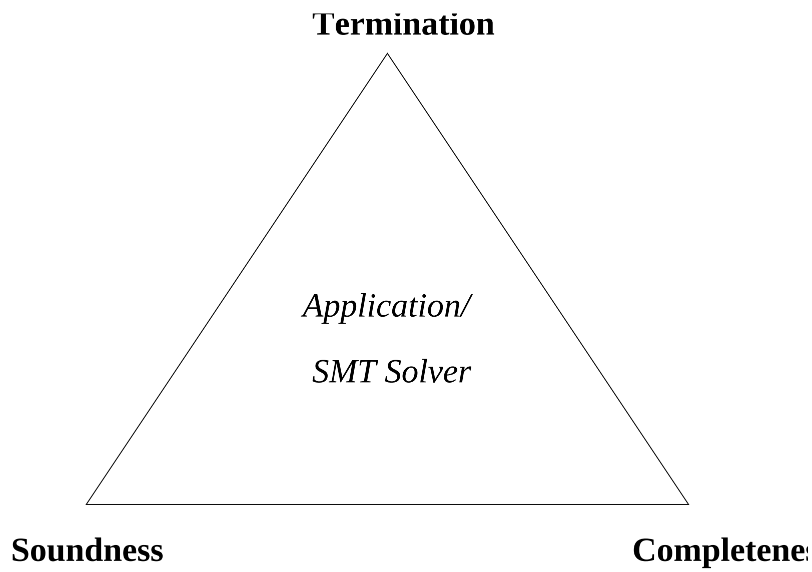
Skewing the Symmetry



Can still build sound tools

That continue to be incomplete

Landscape



If a certain problem is undecidable, then we cannot have a sound, complete and terminating technique.

Application will compromise completeness, so backend solver can compromise completeness too!

Applications overcome undecidability, backend solvers overcome inefficiency/undecidability

Outline

1. Part I: Why we need symbolic solvers?
2. Part II: What are SMT solvers? How to overcome complexity barriers?
3. Part III: Theory of Reals = Gröbner basis + ?

Nonlinear Real Arithmetic: Problem

Focus on $\forall$ formulas first

Given a set of nonlinear equations and inequalities:

$$\begin{aligned} p = 0, &\qquad p \in P \\ q > 0, &\qquad q \in Q \\ r \geq 0, &\qquad r \in R \end{aligned}$$

where $P, Q, R \subset \mathbb{Q}[\vec{x}]$ are sets of polynomials over $\vec{x}$

Is the above set unsatisfiable over the reals?

Examples

Examples of satisfiable constraints:

$$\{x^2 = 2\}$$
$$\{x^2 = 2, \;\; x < 0, y \geq x\}$$

Examples of unsatisfiable constraints:

$$\{x^2 = -2, \; y \geq x\}$$
$$\{x^2 = 2, \;\; 2x > 3\}$$

Applications in: control, robotics, solving games, static analysis, hybrid systems, . . .

Known Results

- The full FO theory of reals is decidable [Tarski48]
  Nonelementary decision procedure, impractical
- Double-exponential time decision procedure [Collins74, MonkSolovay74]
- Exponential space lower bound
- Collin's algorithm based on "cylindrical algebraic decomposition" has been improved over the years and implemented in QEPCAD.
  In practice, could fail on $p > 0 \wedge p < 0$.

Obtaining efficient, sound and complete method unlikely

SMT+/SMT-: Can we obtain efficiency by relaxing completeness?

SMT- Procedure for NRA

The approach is reminiscent of Simplex

- Introduce slack variables s.t. all inequality constraints are of the form $v > 0$, or $w \geq 0$

$$\begin{array}{llll} P = 0, & Q > 0, & R \geq 0 & \mapsto \\ \underline{P = 0}, & \underline{Q - \vec{v} = 0}, & \underline{R - \vec{w} = 0}, & \vec{v} > 0,\ \vec{w} \geq 0 \end{array}$$

- Search for a polynomial p s.t.

$$\begin{array}{rcl} P = 0 \wedge Q = \vec{v} \wedge R = \vec{w} & \Rightarrow & p = 0 \\ \vec{v} > 0,\ \vec{w} \geq 0 & \Rightarrow & p > 0 \end{array}$$

- If we find such a p, return "unsatisfiable" else return "maybe satisfiable"

How to search for p?

Witness for unsatisfiability p satisfies:

$$P = 0 \wedge Q = \vec{v} \wedge R = \vec{w} \quad \Rightarrow \quad p = 0 \tag{1}$$

$$\vec{v} > 0,\ \vec{w} \geq 0 \quad \Rightarrow \quad p > 0 \tag{2}$$

We need efficient sufficient checks

Sufficient check for Condition 1: $p \in Ideal(P, Q - \vec{v}, R - \vec{w})$

Sufficient check for Condition 2: p is a positive polynomial over $\vec{v}, \vec{w}$

To search for p, compute the Gröbner basis for P making $\vec{v}, \vec{w}$ smaller in the ordering

Example: Easy Instance

Consider $E = \{x^3 = x,\ x > 2\}$.

$$\frac{\dfrac{\dfrac{x^3 - x = 0, \qquad x - v - 2 = 0}{(v+2)^3 - (v+2) = 0, \quad x - v - 2 = 0}}{v^3 + 6v^2 + 11v + 6 = 0, \quad x - v - 2 = 0}}{\bot}$$

Computing GB and projecting it onto the slack variables discovers the witness p for unsatisfiability

May not work always ...

Example: Harder Instance

Let $I = \{v_1 > 0, v_2 > 0, v_3 > 0\}$.

$$v_1 + v_2 - 1 = 0, \qquad v_1 v_3 + v_2 - v_3 - 2 = 0$$

$$v_1 + v_2 - 1 = 0, \qquad (1 - v_2)v_3 + v_2 - v_3 - 2 = 0$$

$$v_1 + v_2 - 1 = 0, \qquad v_2 v_3 - v_2 + 2 = 0$$

This is a Gröbner basis.

There is an unsatisfiability witness p for this example, but we failed to find it.

Recall that in the linear case, Simplex performs pivoting

What is the nonlinear analogue of pivoting

First, let us revisit GB computation

Gröbner Basis

Algorithm for computing Gröbner basis is a completion algorithm

Idea behind completion:

- Starting with a set of facts
- Add new facts (saturation)
  - that do not have a smaller proof using existing facts
- Delete any fact (simplification)
  - that do have a smaller proof using other facts

Gröbner Basis: Example

View as completion enables optimizations

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{xy^2 - x = 0,\ x^2y - y^2 = 0}{xy^2 \to x,\ x^2y \to y^2}}{xy^2 \to x,\ x^2y \to y^2[y],\ x^2 = y^3}}{xy^2 \to x,\ x^2y \to y^2[y],\ y^3 \to x^2}}{xy^2 \to x[y],\ x^2y \to y^2[y],\ y^3 \to x^2,\ xy = x^3}}{xy^2 \to x[y],\ x^2y \to y^2[y],\ y^3 \to x^2,\ x^3 \to xy}}{xy^2 \to x[y, x^2],\ x^2y \to y^2[y, x],\ y^3 \to x^2,\ x^3 \to xy}$$

Property of Gröbner Basis

If

$$p' \in Ideal(P)$$

$$G \; : \; \text{Gröbner basis for } \; P$$

Then

$$p' \leftrightarrow^*_P 0 \qquad \text{definition of ideal}$$

$$p' \rightarrow^*_G 0 \qquad \text{definition of GB}$$

Claim. If there is no $p'' \prec p'$ s.t. $p'' \in Ideal(P)$, then $p' \in G$.
Proof. If $p' \rightarrow_G p'' \rightarrow^*_G 0$, then $p' \succ p''$ and both $p', p'' \in Ideal(P)$.

Example: Easy Instance

Recall: We prove unsatisfiability of $P = 0 \wedge Q > 0 \wedge R \geq 0$ by searching for a polynomial p s.t.

$$\begin{aligned} P = 0 \wedge Q = \vec{v} \wedge R = \vec{w} \quad &\Rightarrow \quad p = 0 \\ \vec{v} > 0,\ \vec{w} \geq 0 \quad &\Rightarrow \quad p > 0 \end{aligned}$$

Consider $E = \{x^3 = x,\ x > 2\}$.

$$\frac{\dfrac{\dfrac{x^3 - x = 0, \qquad x - v - 2 = 0}{(v+2)^3 - (v+2) = 0, \quad x - v - 2 = 0}}{v^3 + 6v^2 + 11v + 6 = 0, \quad x - v - 2 = 0}}{\perp}$$

Finding p

We know $p \in Ideal(P)$.

If p is "small-enough" in the ordering $\succ$, then p will appear explicitly in the Gröbner basis for P constructed using $\succ$.

Example: $P = \{w_1 - 2w_3 + 2,\ w_2 + 2w_3 - 1\}$ and $I = \{w_1 \geq 0, w_2 \geq 0\}$.

If $w_1 \succ w_2 \succ w_3$, then $GB_{\succ}(P) = P$.

If we make $w_3 \succ w_1$ and $w_3 \succ w_2$ in the ordering, then

$$GB_{\succ}(P) = \{2w_3 - w_1 - 2,\ \underline{w_2 + w_1 + 1}\}.$$

For linear polynomials, this is pivoting, but what is its analogue for nonlinear systems ?

Finding p: Nonlinear Issues

It is not always possible to change $\succ$ to get witness $p \in GB_{\succ}(P)$.

- Problem 1:

$$P_1 = \{v + w_1 - 1,\ w_1 w_2 - w_1 + 1\}$$

Need $w_1 \succ w_1 w_2$ to "get" $v + w_1 w_2$ in $GB(P_1)$.

Solution: Introduce new definitions and get flexibility in choosing $\succ$
Add $w_1 w_2 - w_3$ to P_1 and have $w_1 \succ w_3$.

Problem 1: Example

$$\frac{\frac{\frac{\frac{\frac{v + w_1 - 1 = 0,\ w_1 w_2 - w_1 + 1 = 0}{v \to -w_1 + 1,\ w_1 w_2 \to w_1 - 1}}{v \to -w_1 + 1,\ w_1 w_2 \to w_1 - 1,\ w_1 w_2 \to w_3}}{v \to -w_1 + 1,\ w_1 \to w_3 + 1,\ w_1 w_2 \to w_3}}{v \to -w_3,\ w_1 \to w_3 + 1,\ w_1 w_2 \to w_3}}{\bot}$$

Finding p: Nonlinear Issues

It is not always possible to change $\succ$ to get witness $p \in GB_{\succ}(P)$.

- Problem 2:

$$P_2 = \{w_1^2 - 2w_1w_2 + w_2^2 + 1\}$$

Need $w_1, w_2 \succ (w_1 - w_2)^2$ to "get" the witness $(w_1 - w_2)^2 + 1$ in $GB(P_2)$.

Solution: Introduce new definitions and get flexibility in choosing $\succ$

Add $(w_1 - w_2)^2 - w_3$ to P_2 and have $w_1, w_2 \succ w_3$.

Problem 2: Example

$$\frac{\frac{\frac{\frac{\frac{w_1^2 - 2w_1w_2 + w_2^2 + 1 = 0}{w_1^2 \to 2w_1w_2 - w_2^2 - 1}}{w_1^2 \to 2w_1w_2 - w_2^2 - 1,\ (w_1 - w_2)^2 = w_3}}{w_1^2 \to 2w_1w_2 - w_2^2 - 1,\ w_1^2 \to 2w_1w_2 - w_2^2 + w_3}}{w_3 \to -1,\ w_1^2 \to 2w_1w_2 - w_2^2 + w_3}}{\bot}$$

Positivstellensatz

What guarantees the existence of such a witness?

The constraint

$$\{p = 0 : p \in P\} \cup \{q \geq 0 : q \in Q\} \cup \{r \neq 0 : r \in R\}$$

is unsatisfiable (over the reals) iff
there exist polynomials p, q, and r such that

$$\begin{array}{ll} p \in Ideal(P) & \{\Sigma_i p_i q_i : p_i \in P\} \\ q \in Cone[Q] & \{\Sigma_i s_i^2 q_1 q_2 \ldots q_k : q_j \in Q\} \\ r \in [R] & \{r_1 r_2 \ldots r_k : r_i \in R\} \\ p + q + r^2 \equiv 0 & \end{array}$$

Positivstellensatz Corollary

The constraint

$$\{p = 0 : p \in P\} \cup \{v > 0 : v \in \vec{v}\} \cup \{w \geq 0 : w \in \vec{w}\}$$

is unsatisfiable iff
$\exists p'$ such that

$$p' \in Ideal(P) \cap (Cone[\vec{v}, \vec{w}] + [\vec{v}])$$

Hence, the method is "refutationally complete"

Example: Harder Instance

Let $I = \{v_1 > 0, v_2 > 0, v_3 > 0\}$.

$$v_1 + v_2 - 1 = 0, \quad v_1 v_3 + v_2 - v_3 - 2 = 0$$

$$v_1 + v_2 - 1 = 0, \quad (1 - v_2)v_3 + v_2 - v_3 - 2 = 0$$

$$v_1 + v_2 - 1 = 0, \quad v_2 v_3 - v_2 + 2 = 0$$

$$v_1 + v_2 - 1 = 0, \quad v_2 v_3 - v_2 + 2 = 0, \quad v_2 v_3 - v_4 = 0$$

$$v_1 + v_2 - 1 = 0, \quad -v_2 + v_4 + 2 = 0, \quad v_2 v_3 - v_4 = 0$$

$$v_1 + v_4 + 1 = 0, \quad -v_2 + v_4 + 2 = 0, \quad v_2 v_3 - v_4 = 0$$

$$\perp$$

The polynomial $v_1 + v_4 + 1$ is the required witness to the unsatisfiability of the constraints.

Summary of the Procedure

- Turn all inequalities into equations by introducing slack variables
- Compute Gröbner basis of the equations
- If a positive polynomial is ever generated, return unsatisfiable
- If not, introduce new definitions to try different orderings and repeat

Solving $\exists\forall$ Formulas

Farkas' Lemma converts $\forall$ to $\exists$ in linear arithmetic

Its generalization can be used for nonlinear arithmetic

$\forall \vec{x} : p_1 \geq 0 \wedge p_2 \geq 0 \Rightarrow p_3 \geq 0$, if

$$\exists s_1, s_2, s_3 : s_3 p_3 = s_1 p_1 + s_2 p_2 \ \wedge \ s_1 \geq 0 \ \wedge \ s_2 \geq 0 \ \wedge \ s_3 \geq 0$$

A sufficient condition for guaranteeing $s_1, s_2 \geq 0$ is that they are sums of squares

Once $\forall$ is eliminated, we can use the procedure for $\exists$

Solving $\exists\forall$ Formulas

Another approach we are pursuing is based on
Combining symbolic and numeric techniques

Suppose we wish to solve $\exists x_1, x_2 : \forall y : p(x_1, x_2, y) \geq 0 \wedge q(x_1, x_2, y) \geq 0$

- Use QEPCAD to eliminate $\forall$ from $\forall y : p(x_1, x_2, y) \geq 0$
- Use numerical techniques to get a value for x_1
- Use QEPCAD to eliminate $\forall$ from $\forall y : q(x_1, x_2, y) \geq 0$ with x_1 instantiated

Sum-of-Squares Programming

The need for nonlinear reasoning and optimization has been recognized by several communities

This has lead to the formulation of SOS programming

$$min_{\vec{u}\in\mathbb{R}^n} c_1u_1 + \cdots + c_nu_n$$

subject to

$p_{i1}u_1 + \cdots + p_{in}u_n$ is a SOS, $i = 1, 2, \ldots, k$

SOS programs can be converted into semidefinite programs using the observation that
p is SOS iff $p = z^TQz$ for some symmetric positive-semidefinite matrix Q
(z is a vector of all monomials of degree $deg(p)/2$)

Semidefinite Programming

Semidefinite Programming:

$$min_{\vec{u} \in \mathbb{R}^n} c_1 u_1 + \cdots + c_n u_n$$

subject to

$F_0 + u_1 F_1 + \cdots + u_n F_n$ is positive semidefinite

where c_i's are given constants and F_i's are given symmetric matrices.

SDPs can be solved using numerical convex optimization toolboxes

Is there a good way to combine SOS techniques with symbolic techniques?

Conclusion

Symbolic and algebraic techniques will play increasingly important role as we design, build and understand complex systems

We need fast and scalable tools that can be embedded in applications: SMT+CAS?

There is a market for incomplete but fast tools

Reasoning about nonlinear constraints is presently a critical bottleneck

We will need to augment sound symbolic techniques with fast numerical approaches