Nonlinear Real Arithmetic and δ-Satisfiability

Paolo Zuliani

School of Computing Science
Newcastle University, UK

(Slides courtesy of Sicun Gao, UCSD)

Introduction

- We use hybrid systems for modelling and verifying *biological* and *cyber-physical system* models:
  - atrial fibrillation (CMSB 2014)
  - prostate cancer therapy (HSCC 2015)
  - psoriasis UVB treatment (HVC 2016)
  - artificial pancreas (this tutorial — paper coming soon)
- Hybrid systems combine continuous dynamics with discrete state changes.

Why Nonlinear Real Arithmetic and Hybrid Systems? (I)

A prostate cancer model[1]

$$\frac{dx}{dt} = \left(\frac{\alpha_x}{1+\mathrm{e}^{(k_1-z)k_2}} - \frac{\beta_x}{1+\mathrm{e}^{(z-k_3)k_4}} - m_1\left(1-\frac{z}{z_0}\right) - c_1\right)x + c_2$$

$$\frac{dy}{dt} = m_1\left(1-\frac{z}{z_0}\right)x + \left(\alpha_y\left(1-d_0\frac{z}{z_0}\right) - \beta_y\right)y$$

$$\frac{dz}{dt} = -z\gamma - c_3$$

$$v = x + y$$

- v - *prostate specific antigen* (PSA)
- x - *hormone sensitive cells* (HSCs)
- y - *castration resistant cells* (CRCs)
- z - androgen

[1] A.M. Ideta, G. Tanaka, T. Takeuchi, K. Aihara: A mathematical model of intermittent androgen suppression for prostate cancer. *Journal of Nonlinear Science*, 18(6), 593–614 (2008)

Why Nonlinear Real Arithmetic and Hybrid Systems? (I)

Intermittent androgen deprivation therapy

on-therapy

$$\frac{dx}{dt} = \left(\frac{\alpha_x}{1+e^{(k_1-z)k_2}} - \frac{\beta_x}{1+e^{(z-k_3)k_4}} - m_1\ \left(1 - \frac{z}{z_0}\right) - c_1\right)x + c_2$$

$$\frac{dy}{dt} = m_1\ \left(1 - \frac{z}{z_0}\right)x + \left(\alpha_y\left(1 - \frac{d_0 z}{z_0}\right) - \beta\right)y$$

$$\frac{dz}{dt} = -z\gamma + c_3$$

$x + y \leq r_0$

$x + y \geq r_1$

off-therapy

$$\frac{dx}{dt} = \left(\frac{\alpha_x}{1+e^{(k_1-z)k_2}} - \frac{\beta_x}{1+e^{(z-k_3)k_4}} - m_1\ \left(1 - \frac{z}{z_0}\right) - c_1\right)x + c_2$$

$$\frac{dy}{dt} = m_1\ \left(1 - \frac{z}{z_0}\right)x + \left(\alpha_y\left(1 - \frac{d_0 z}{z_0}\right) - \beta\right)y$$

$$\frac{dz}{dt} = (z_0 - z)\gamma + c_3$$

Why Nonlinear Real Arithmetic and Hybrid Systems? (II)

A model of psoriasis development and UVB treatment[2]

$$
\begin{aligned}
\frac{dSC}{dt} &= \gamma_1 \frac{\omega(1-\frac{SC+\lambda SC_d}{SC_{max}})SC}{1+(\omega-1)(\frac{TA+TA_d}{P_{ta,h}})^n} - \beta_1 In_A SC - \frac{k_{1s}\omega}{1+(\omega-1)(\frac{TA+TA_d}{P_{ta,h}})^n SC + k_1 TA} \\
\frac{dTA}{dt} &= \frac{k_{1a,s}\omega SC}{1+(\omega-1)(\frac{TA+TA_d}{P_{ta,h}})^n} + \frac{2k_{1s\omega}}{1+(\omega-1)(\frac{TA+TA_d}{P_{ta,h}})^n + \gamma_2 GA - \beta_2 In_A TA - k_{2s} TA - k_1 TA} \\
\frac{dGA}{dt} &= (k_{2a,s} + 2k_{2s}) TA - k_2 GA - k_3 GA - \beta_3 GA \\
\frac{dSC_d}{dt} &= \gamma_{1d}(1 - \frac{SC + SC_d}{SC_{max,t}} SC_d - \beta_{1d} In_A SC_d - k_{1sd} SC_d - \frac{k_p SC_d^2}{k_a^2 + SC_d^2} + k_{1d} TA_d) \\
\frac{dTA_d}{dt} &= k_{1a,sd} SC_d + 2k_{1sd} SC_d + \gamma_{2d} TA_d + k_{2d} GA_d - \beta_{2d} In_A TA_d - k_{2sd} TA_d - k_{1d} TA_d \\
\frac{dGA_d}{dt} &= (k_{2a,sd} + 2k_{2sd}) TA_d - k_{2d} GA_d - k_{3d} GA_d - \beta_{3d} GA_d
\end{aligned}
$$

- Therapy episode: 48 hours of irradiation + 8 hours of rest

[2] H. Zhang, W. Hou, L. Henrot, S. Schnebert, M. Dumas, C. Heusèle, and J. Yang. Modelling epidermis homoeostasis and psoriasis pathogenesis. *Journal of The Royal Society Interface*, 12(103), 2015.

Why Nonlinear Real Arithmetic and Hybrid Systems? (II)

A model of psoriasis development and UVB treatment[2]

$$
\begin{aligned}
\frac{dSC}{dt} &= \gamma_1 \frac{\omega(1-\frac{SC+\lambda SC_d}{SC_{max}})SC}{1+(\omega-1)(\frac{TA+TA_d}{P_{ta,h}})^n} - \boxed{\boldsymbol{\beta_1}} In_A SC - \frac{k_{1s}\omega}{1+(\omega-1)(\frac{TA+TA_d}{P_{ta,h}})^n SC + k_1 TA} \\
\frac{dTA}{dt} &= \frac{k_{1a,s}\omega SC}{1+(\omega-1)(\frac{TA+TA_d}{P_{ta,h}})^n} + \frac{2k_{1s\omega}}{1+(\omega-1)(\frac{TA+TA_d}{P_{ta,h}})^n + \gamma_2 GA - \boxed{\boldsymbol{\beta_2}} In_A TA - k_{2s} TA - k_1 TA} \\
\frac{dGA}{dt} &= (k_{2a,s} + 2k_{2s}) TA - k_2 GA - k_3 GA - \beta_3 GA \\
\frac{dSC_d}{dt} &= \gamma_{1d}(1 - \frac{SC+SC_d}{SC_{max,t}} SC_d - \beta_{1d} In_A SC_d - k_{1sd} SC_d - \frac{k_p SC_d^2}{k_a^2 + SC_d^2} + k_{1d} TA_d) \\
\frac{dTA_d}{dt} &= k_{1a,sd} SC_d + 2k_{1sd} SC_d + \gamma_{2d} TA_d + k_{2d} GA_d - \beta_{2d} In_A TA_d - k_{2sd} TA_d - k_{1d} TA_d \\
\frac{dGA_d}{dt} &= (k_{2a,sd} + 2k_{2sd}) TA_d - k_{2d} GA_d - k_{3d} GA_d - \beta_{3d} GA_d
\end{aligned}
$$

- Therapy episode: 48 hours of irradiation + 8 hours of rest
- Therapy episode = multiply β_1 and β_2 by a constant In_A

[2] H. Zhang, W. Hou, L. Henrot, S. Schnebert, M. Dumas, C. Heusèle, and J. Yang. Modelling epidermis homoeostasis and psoriasis pathogenesis. *Journal of The Royal Society Interface*, 12(103), 2015.

Bounded Reachability

- Reachability is a key property in verification, also for hybrid systems.
- Reachability is **undecidable** even for linear hybrid systems (Alur, Courcoubetis, Henzinger, Ho. 1993).
- [*Bounded* Reachability] Does the hybrid system reach a *goal* state within a finite time and number of (discrete) steps?
  - "*Can a 5-episode UVB therapy remit psoriasis for a year*?"

Bounded Reachability

- Reachability is a key property in verification, also for hybrid systems.
- Reachability is **undecidable** even for linear hybrid systems (Alur, Courcoubetis, Henzinger, Ho. 1993).
- [*Bounded* Reachability] Does the hybrid system reach a *goal* state within a finite time and number of (discrete) steps?
  - "*Can a 5-episode UVB therapy remit psoriasis for a year*?"
- Nonlinear arithmetics over the reals is **undecidable** (Tarski 1951, Richardson 1968).
- Hence, the problem needs to be simplified if we want to solve it algorithmically!

Formal Reasoning for Nonlinear Arithmetics

- Novak and Woźniakowski (J of Complexity, 1992) studied the *relaxed* verification problem:
  - verify that a candidate is close to a problem solution
  - introduce a parametric "safety zone" for which either answer is deemed correct
  - focus on computational complexity

Formal Reasoning for Nonlinear Arithmetics

- Novak and Woźniakowski (J of Complexity, 1992) studied the *relaxed* verification problem:
  - verify that a candidate is close to a problem solution
  - introduce a parametric "safety zone" for which either answer is deemed correct
  - focus on computational complexity
- Fränzle's work on hybrid automata (since 1999).

Formal Reasoning for Nonlinear Arithmetics

- Novak and Woźniakowski (J of Complexity, 1992) studied the *relaxed* verification problem:
  - verify that a candidate is close to a problem solution
  - introduce a parametric "safety zone" for which either answer is deemed correct
  - focus on computational complexity
- Fränzle's work on hybrid automata (since 1999).
- Ratschan's work on constraint solving (since 2001).

Formal Reasoning for Nonlinear Arithmetics

- Novak and Woźniakowski (J of Complexity, 1992) studied the *relaxed* verification problem:
  - verify that a candidate is close to a problem solution
  - introduce a parametric "safety zone" for which either answer is deemed correct
  - focus on computational complexity
- Fränzle's work on hybrid automata (since 1999).
- Ratschan's work on constraint solving (since 2001).
- Gao, Avigad, Clarke (LICS 2012): bounded δ-satisfiability over the reals is **decidable**:
  - δ-complete decision procedure.

Background: Type 2 Computability

Turning machines operate on finite strings, *i.e.*, integers, which cannot capture real-valued functions.

- Real numbers can be encoded on *infinite* tapes.
  - Real numbers are functions over integers.
- Real functions can be computed by machines that take infinite tapes as inputs, and output infinite tapes encoding the values.

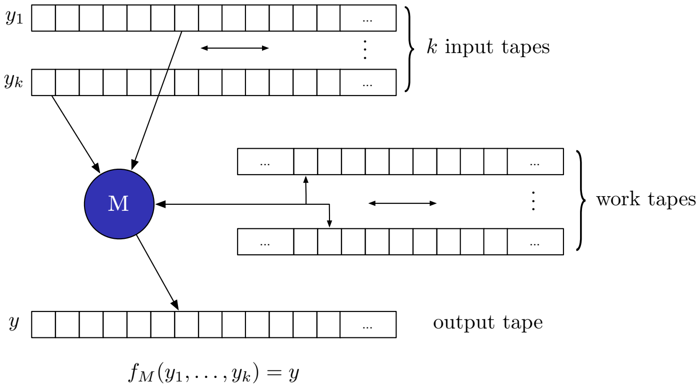
Definition (Name of a real number)

A real number a can be encoded by an **infinite sequence** of rationals $\gamma_a : \mathbb{N} \to \mathbb{Q}$ such that

$$\forall i \in \mathbb{N}\ |a - \gamma_a(i)| < 2^{-i}.$$

Background: Type 2 Computability

A function $f(x) = y$ is computable if any name of x can be algorithmically mapped to a name of y



Writing on any finite segment of the output tape takes finite time.

Background: Type 2 Computability

- Type 2 computability implies continuity.
- "Numerically computable" roughly means Type 2 computable.
- Approximation up to arbitrary numerical precisions.

Ker-I Ko. *Complexity Theory of Real Functions*. 1991.

Background: Type 2 Computability

Type 2 Computable:

- polynomials, sin, exp, ...
- numerically feasible ODEs, PDEs, ...

Type 2 Complexity:

- sin, exp, etc. are in $\mathrm{P}_{[0,1]}$
- Lipschitz-continuous ODEs are in $\mathrm{PSPACE}_{[0,1]}$; in fact, can be $\mathrm{PSPACE}_{[0,1]}$-complete (Kawamura, CCC 2009).

See Ko's book for many more results ...

$\mathcal{L}_{\mathbb{R}_{\mathcal{F}}}$-Formulas (Gao, Avigad, and Clarke. LICS 2012)

Let $\mathcal{F}$ be the class of all Type 2 computable real functions.

Definition ($\mathcal{L}_{\mathbb{R}_{\mathcal{F}}}$-Formulas)

First-order language over $\langle >, \mathcal{F} \rangle$:

$$t := x \mid f(t(\vec{x}))$$
$$\varphi := t(\vec{x}) > 0 \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x_i \varphi \mid \forall x_i \varphi$$

Example

Let $dx/dt = f(x)$ be an n-dimensional dynamical system.
Lyapunov stability is expressed as:

$$\forall\varepsilon\exists\delta\forall t\forall x_0\forall x_t.\ \left(||x_0|| < \delta \wedge x_t = x_0 + \int_0^t f(s)ds\right) \to ||x_t|| < \varepsilon$$

Hybrid Automata

A hybrid automaton is a tuple

$$H = \langle X, Q, \{\mathsf{flow}_q(\vec{x}, \vec{y}, t) : q \in Q\}, \{\mathsf{jump}_{q \to q'}(\vec{x}, \vec{y}) : q, q' \in Q\}, \\ \{\mathsf{inv}_q(\vec{x}) : q \in Q\}, \{\mathsf{init}_q(\vec{x}) : q \in Q\}\rangle$$

- $X \subseteq \mathbb{R}^n$ for some $n \in \mathbb{N}$
- $Q = \{q_1, ..., q_m\}$ is a finite set of modes
- Other components are finite sets of quantifier-free $\mathcal{L}_{\mathbb{R}_{\mathcal{F}}}$-formulas.

Example: Nonlinear Bouncing Ball

- $X = \mathbb{R}^2$ and $Q = \{q_u, q_d\}$.
- $\text{flow}_{q_d}(x_0, v_0, x_t, v_t, t)$, dynamics in the falling phase:

$$(x_t = x_0 + \int_0^t v(s)ds) \wedge (v_t = v_0 + \int_0^t g(1 + \beta v(s)^2)ds)$$

- $\text{jump}_{q_u \to q_d}(x, v, x', v')$:

$$(v = 0 \wedge x' = x \wedge v' = v)$$

- inv_{q_d}: $(x >= 0 \wedge v >= 0)$.
- init_{q_d}: $(x = 10 \wedge v = 0)$.

Encode Reachability

Continuous case:

$$\mathrm{init}(\vec{x}_0) \wedge \mathrm{flow}(\vec{x}_0, t, \vec{x}_t) \wedge \mathrm{goal}(\vec{x}_t)$$

Make one jump:

$$\mathrm{init}(\vec{x}_0) \wedge \mathrm{flow}(\vec{x}_0, t, \vec{x}_t) \wedge \mathrm{jump}(\vec{x}_t, \vec{x}'_t) \wedge \mathrm{goal}(\vec{x}'_t)$$

Encode Reachability: invariant-free case

$$\exists^X \vec{x}_0 \exists^X \vec{x}_0^t \dots \exists^X \vec{x}_k \exists^X \vec{x}_k^t \exists^{[0,M]} t_0 \dots \exists^{[0,M]} t_k$$

$$\bigvee_{q \in Q} \Big(\mathrm{init}_q(\vec{x}_0) \wedge \mathrm{flow}_q(\vec{x}_0, \vec{x}_0^t, t_0) \Big)$$

$$\wedge \quad \bigwedge_{i=0}^{k-1} \Bigg(\bigvee_{q,q' \in Q} \Big(\mathrm{jump}_{q \to q'}(\vec{x}_i^t, \vec{x}_{i+1}) \wedge \mathrm{flow}_{q'}(\vec{x}_{i+1}, \vec{x}_{i+1}^t, t_{i+1}) \Big) \Bigg)$$

$$\wedge \quad \bigvee_{q \in Q} (\mathrm{goal}_q(\vec{x}_k^t))$$

(There's some simplification here.)

Difficulty

Suppose $\mathcal{F}$ is $\{+, \times\}$.

$$\mathbb{R} \models \exists a \forall b \exists c \ (ax^2 + bx + c > 0)?$$

- Decidable [Tarski 1948] but double-exponential lower-bound.

Suppose $\mathcal{F}$ further contains **sine**.

$$\mathbb{R} \models \exists x, y, z \ (\sin^2(\pi x) + \sin^2(\pi y) + \sin^2(\pi z) = 0 \bigwedge x^3 + y^3 = z^3)?$$

- **Undecidable**.

Towards δ-Decisions

Defining δ-decision problems of $\mathcal{L}_{\mathbb{R}_{\mathcal{F}}}$-formulas leads to a totally different outlook.

Bounded $\mathcal{L}_{\mathcal{F}}$-Sentences

Definition (Normal Form)

Any bounded $\mathcal{L}_{\mathcal{F}}$-sentence φ can be written in the form

$$Q_1^{[u_1,v_1]}x_n \cdots Q_n^{[u_n,v_n]}x_n \ \bigwedge\Big(\bigvee t(\vec{x}) > 0 \vee \bigvee t(\vec{x}) \geq 0\Big)$$

- Negations are pushed into atoms.
- Bounded quantifiers: the bounds can use any terms that contain previously-quantified variables.

δ-Variants

Definition (Numerical Perturbation)

Let $\delta \in \mathbb{Q}^+ \cup \{0\}$. The **$\delta$-weakening** $\varphi^{-\delta}$ of φ is

$$Q_1^{[u_1,v_1]}x_1 \cdots Q_n^{[u_n,v_n]}x_n \bigwedge(\bigvee t(\vec{x}) > -\delta \vee \bigvee t(\vec{x}) \geq -\delta)$$

- Obviously, $\varphi \to \varphi^{-\delta}$ (but not the other way round!)
- **δ-strengthening** $\varphi^{+\delta}$ is defined by replacing $-\delta$ by δ.

δ-Decisions

Let $\delta \in \mathbb{Q}^+$ be arbitrary.

Definition (δ-Decisions)

Decide, for any given bounded φ and $\delta \in \mathbb{Q}^+$, whether

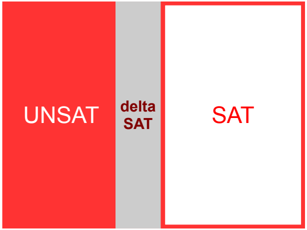
- φ is false, or
- $\varphi^{-\delta}$ is true.

When the two cases overlap, either answer can be returned.

The dual can be defined on δ-strengthening.

δ-Decisions

There is a grey area that a δ-complete algorithm can be wrong about.



Corollary

In undecidable theories, it is undecidable whether a formula falls into this grey area.

δ-Decidability

Let $\mathcal{F}$ be an arbitrary collection of Type 2 computable functions.

Theorem

The δ-decision problem over $\mathbb{R}_{\mathcal{F}}$ is decidable.

See [Gao *et al*. LICS 2012].

It stands in sharp contrast to the high undecidability of simple formulas containing sine.

δ-Robustness

- A bounded $\mathcal{L}_{\mathbb{R}}$-sentence ϕ is δ-**robust** iff $\phi^\delta \to \phi$.
  - ϕ is **robust** if it is δ-robust for some $\delta > 0$.

- Suppose ϕ is robust
  - if ϕ is true, then $\forall \delta > 0 : \phi^\delta \to \phi$,
  - if ϕ is false, then $\exists \delta > 0 : \neg\phi \to \neg\phi^\delta$.

Theorem

Given a **robust** *bounded $\mathcal{L}_{\mathbb{R}}$-sentence ϕ, there exists $\delta > 0$ for which a δ-complete decision procedure* **correctly** *decides whether ϕ is true or false.*

- Thus, robustness $\Rightarrow$ decidability.
  - However, decidability $\not\Rightarrow$ robustness.

Complexity

Let S be some class of $\mathcal{L}_{\mathcal{F}}$-sentences such that all the terms appearing in S are in Type 2 complexity class C. Then for any $\delta \in \mathbb{Q}^+$:

Theorem

The δ-decision problem for a Σ_k-**sentence** *from S is in* $(\Sigma_k^{\mathsf{P}})^{\mathsf{C}}$.

Corollary

- $\mathcal{F} = \{+, \times, \exp, \sin, ...\}$*:* Σ_k^{P}-**complete**.
- $\mathcal{F} = \{$*ODEs with* P *right-hand sides*$\}$*:* PSPACE-**complete**.

These are very reasonable!

Exactness

The definition of δ-decisions is exact in the following sense.

Theorem

If $\mathcal{F}$ is allowed to be arbitrary, then φ is decidable **iff** *we consider bounded δ-decisions.*

Theorem

Bounded sentences are δ-decidable **iff** *$\mathcal{F}$ is computable.*

Conclusions

The notion of δ-complete decision procedures allows formal analysis and use of numerical algorithms in decision procedures.

- Standard completeness is impossible.
- δ-completeness: strong enough and achievable.
    - Correctness guarantees on both sides