Applying Exact Real Arithmetic to Solving Non-linear Constraints [1]

Franz Brauße[1] K. Korovin[2] M. Korovina[3] N. Th. Müller[1]

[1] University of Trier, Germany

[2] The University of Manchester, UK

[3] IIS, Novosibirsk, Russia

Caramba Seminar, INRIA Nancy, 2019-06-04

[1] This project has received funding from the DFG grant WERA MU 1801/5-1, the DFG/RFBR grant CAVER BE 1267/14-1 and 14-01-91334 and from the EU's Horizon 2020 programme under the Marie Skłodowska-Curie grant agreement No 731143.

Overview

1. Introduction
   - Motivation
   - Non-Linear Constraints
2. `ksmt` calculus
   - States and rules
   - Properties
3. Examples & Concrete Linearisations
4. Benchmarks
5. Conclusions

Motivation

System design ubiquitous:

- Engineering: airplanes, trains, air-traffic control, ...
- Hardware: processors, electronic circuits, ...
- Software: operating systems, security protocols, ...
- Information Management: databases, semantic web, search engines, ...

When we design a system, we would like to be sure that it will satisfy all requirements, such as reliability, safety and reachability.

Sometimes purely mathematical motivation: Formal proof of Kepler's conjecture [Hales et al., 2015].

How to employ formal methods

Problems can be stated as a first-order formula over extensions of language $\langle \mathbb{R}, +, *, 0, 1, < \rangle$.

How to employ formal methods

Problems can be stated as a first-order formula over extensions of language $\langle \mathbb{R}, +, *, 0, 1, < \rangle$.

- Real closed field axioms (+ order) provide identities on terms.

How to employ formal methods

Problems can be stated as a first-order formula over extensions of language $\langle \mathbb{R}, +, *, 0, 1, < \rangle$.

- Real closed field axioms (+ order) provide identities on terms.

Algebraic functions: Satisfiability decidable [Tarski, 1951] by symbolical methods (quantifier elimination), still (doubly) exponential complexity in dimension.

Captures just a restricted class of dynamics.

Differential equations often lead to transcendental functions like $\sin$, $\tan$, $\exp$.
$\rightsquigarrow$ Satisfiability undecidable in unrestricted cases [Richardson, 1968].

How to employ formal methods

Promising approach: combine symbolical and numerical methods to increase efficiency reducing

- numeric drawback: wrapping effect
- symbolic drawback: complexity dependency on dimension

Numerical methods to approximate transcendental reals.

- `MPFR`, `MPFI`, `arb`, `iRRAM`, ...
- Explored in some directions, e.g., `veriT+redlog+raSAT`, `cvc4`, `mathsat`, `dReal`

Our approach:
Combine reliable real computations and resolution in a CDCL-style[2] calculus called `ksmt`.

[2] Conflict-Driven Clause Learning – a powerful algorithm scheme for SAT solving

Existentially quantified formula in CNF (i.e., $\bigwedge_i \bigvee_j \ell_{ij}$) where ℓ_{ij} are predicates or negated predicates over $(\mathbb{R}, \mathcal{F}_{\mathsf{lin}} \cup \mathcal{F}_{\mathsf{nl}}, \mathcal{P})$.

- $\mathcal{F}_{\mathsf{lin}}$: constants $\in \mathbb{Q}$, addition, multiplication by constants $\in \mathbb{Q}$
- $\mathcal{F}_{\mathsf{nl}}$: non-linear functions, incl. multiplication and transcendental functions
- $\mathcal{P} = \{<, \leq, >, \geq\}$ are predicates

Example

$$\exists x, y : \Big(((\sin x)^2 + (\cos x)^2 < 1) \vee \big(\exp x < y\big) \Big) \wedge \Big(4 \cdot x > y\Big)$$

An assignment $\alpha : V \to \mathbb{Q}$ is a solution to such a CNF $\mathcal{C}$ over variables V iff

- α assigns all quantified variables
- for each clause $C \in \mathcal{C}$ there is $\ell \in C$ with evaluates to true, in symbols: $[\![\ell]\!]^\alpha = \text{true}$

Problem: finding solution to $\mathcal{C}$ or showing that none exists.

Simplify: Transformation of CNF $\mathcal{C}$ into equi-satisfiable separated linear form $\mathcal{L} \wedge \mathcal{N}$ with

- $\mathcal{L}$ contains only linear predicates $q_1 x_1 + q_2 x_2 + \cdots + q_n x_n + q_0 \diamond 0$, $(q_i \in \mathbb{Q})$, and
- $\mathcal{N}$ is of the form $x \diamond f(\boldsymbol{t})$ for $f \in \mathcal{F}_{\mathsf{nl}}$ and $\boldsymbol{t}$ is a vector of terms

is possible in polynomial time (e.g., by introducing new variables), where $\diamond \in \mathcal{P}$.

ksmt calculus

Definition: $(\alpha, \mathcal{L}, \mathcal{N})$ is a state. $(\text{nil}, \mathcal{L}, \mathcal{N})$ is the initial state for separated linear form $\mathcal{L} \wedge \mathcal{N}$. Formula $[\![\cdot]\!]^\alpha$ is partial evaluation.

Let $\Rightarrow$ be the binary relation on states, defined by the following rules.

	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	preconditions

ksmt calculus

Definition: $(\alpha, \mathcal{L}, \mathcal{N})$ is a state. $(\text{nil}, \mathcal{L}, \mathcal{N})$ is the initial state for separated linear form $\mathcal{L} \wedge \mathcal{N}$. Formula $[\![\cdot]\!]^\alpha$ is partial evaluation.

Let $\Rightarrow$ be the binary relation on states, defined by the following rules.

	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	preconditions
(A)	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	$[\![\mathcal{L} \wedge \mathcal{N}]\!]^\alpha \neq$ false, $z \in V \setminus \mathrm{dom}\,\alpha$ and $\exists q \in \mathbb{Q}$: $[\![\mathcal{L}]\!]^{\alpha::z\mapsto q} \neq$ false.

ksmt calculus

Definition: $(\alpha, \mathcal{L}, \mathcal{N})$ is a state. $(\text{nil}, \mathcal{L}, \mathcal{N})$ is the initial state for separated linear form $\mathcal{L} \wedge \mathcal{N}$. Formula $[\![\cdot]\!]^\alpha$ is partial evaluation.

Let $\Rightarrow$ be the binary relation on states, defined by the following rules.

	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	preconditions
(A)	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	$[\![\mathcal{L} \wedge \mathcal{N}]\!]^\alpha \neq$ false, $z \in V \setminus \operatorname{dom} \alpha$ and $\exists q \in \mathbb{Q}$: $[\![\mathcal{L}]\!]^{\alpha::z \mapsto q} \neq$ false.
(R)	$(\alpha, \mathcal{L} \cup R_{\alpha,\mathcal{L},z}, \mathcal{N})$	$[\![\mathcal{L} \wedge \mathcal{N}]\!]^\alpha \neq$ false, $z \in V \setminus \operatorname{dom} \alpha$ and $\forall q \in \mathbb{Q}$: $[\![\mathcal{L}]\!]^{\alpha::z \mapsto q} =$ false.

ksmt calculus

Definition: $(\alpha, \mathcal{L}, \mathcal{N})$ is a state. $(\text{nil}, \mathcal{L}, \mathcal{N})$ is the initial state for separated linear form $\mathcal{L} \wedge \mathcal{N}$. Formula $[\![\cdot]\!]^\alpha$ is partial evaluation.

Let $\Rightarrow$ be the binary relation on states, defined by the following rules.

	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	preconditions
(A)	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	$[\![\mathcal{L} \wedge \mathcal{N}]\!]^\alpha \neq$ false, $z \in V \setminus \mathrm{dom}\, \alpha$ and $\exists q \in \mathbb{Q} : [\![\mathcal{L}]\!]^{\alpha::z\mapsto q} \neq$ false.
(R)	$(\alpha, \mathcal{L} \cup R_{\alpha,\mathcal{L},z}, \mathcal{N})$	$[\![\mathcal{L} \wedge \mathcal{N}]\!]^\alpha \neq$ false, $z \in V \setminus \mathrm{dom}\, \alpha$ and $\forall q \in \mathbb{Q} : [\![\mathcal{L}]\!]^{\alpha::z\mapsto q} =$ false.
(B)	$(\gamma, \mathcal{L}, \mathcal{N})$	$[\![\mathcal{L}]\!]^\alpha =$ false and γ is the maximal prefix of α s.t. $[\![\mathcal{L}]\!]^\gamma \neq$ false.

ksmt calculus

Definition: $(\alpha, \mathcal{L}, \mathcal{N})$ is a state. $(\text{nil}, \mathcal{L}, \mathcal{N})$ is the initial state for separated linear form $\mathcal{L} \wedge \mathcal{N}$. Formula $[\![\cdot]\!]^\alpha$ is partial evaluation.

Let $\Rightarrow$ be the binary relation on states, defined by the following rules.

	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	preconditions
(A)	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	$[\![\mathcal{L} \wedge \mathcal{N}]\!]^\alpha \neq$ false, $z \in V \setminus \operatorname{dom} \alpha$ and $\exists q \in \mathbb{Q} : [\![\mathcal{L}]\!]^{\alpha::z\mapsto q} \neq$ false.
(R)	$(\alpha, \mathcal{L} \cup R_{\alpha,\mathcal{L},z}, \mathcal{N})$	$[\![\mathcal{L} \wedge \mathcal{N}]\!]^\alpha \neq$ false, $z \in V \setminus \operatorname{dom} \alpha$ and $\forall q \in \mathbb{Q} : [\![\mathcal{L}]\!]^{\alpha::z\mapsto q} =$ false.
(B)	$(\gamma, \mathcal{L}, \mathcal{N})$	$[\![\mathcal{L}]\!]^\alpha =$ false and γ is the maximal prefix of α s.t. $[\![\mathcal{L}]\!]^\gamma \neq$ false.
(L)	$(\alpha, \mathcal{L} \cup L_{\alpha,\mathcal{N}}, \mathcal{N})$	$[\![\mathcal{L}]\!]^\alpha \neq$ false and $[\![\mathcal{N}]\!]^\alpha =$ false.

$\mathcal{N}$ remains unchanged under $\Rightarrow$ transformations.

Resolution

(R) Conflict resolution:
$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow (\alpha, \mathcal{L} \cup R_{\alpha,\mathcal{L},z}, \mathcal{N})$
if $[\![\mathcal{L} \wedge \mathcal{N}]\!]^\alpha \neq$ false,
$z \in V \setminus \operatorname{dom} \alpha$ and
$\forall q \in \mathbb{Q} : [\![\mathcal{L}]\!]^{\alpha::z \mapsto q} =$ false.

Resolution in propositional logic:

$$\frac{(A \vee \ell) \quad (B \vee \neg \ell)}{(A \vee B)}$$

Arithmetical resolution on x of clauses over linear predicates:

$$\frac{(A \vee (cx + d \leq 0)) \quad (B \vee (-c'x + d' \leq 0))}{(A \vee B \vee (c'd + cd' \leq 0))}$$

where c, c' are positive rational constants and d, d' are linear terms is sound [Korovin et al., 2009]. Similar rules exist for strict comparisons.

We denote by $R_{\alpha,\mathcal{L},z}$ a set of resolvents of clauses in $\mathcal{L}$ on variable z such that $[\![R_{\alpha,\mathcal{L},z}]\!]^\alpha =$ false.

Linearisation

(L) Linearisation:
$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow (\alpha, \mathcal{L} \cup L_{\alpha,\mathcal{N}}, \mathcal{N})$
if $[\![\mathcal{L}]\!]^\alpha \neq$ false and
$[\![\mathcal{N}]\!]^\alpha =$ false.

Definition

A linear clause L is a linearisation of non-linear predicate P at assignment α iff

- $\forall \beta : [\![P]\!]^\beta =$ true $\implies [\![L]\!]^\beta =$ true, and
- $[\![L]\!]^\alpha =$ false

$L_{\alpha,\mathcal{N}}$ contains (at least) one linearisation of a $P \in \mathcal{N}$ at α.

Some properties of the `ksmt` calculus

Lemma

Let $\mathcal{C}$ be a formula in separated linear form, let $S_i = (\alpha_i, \mathcal{L}_i, \mathcal{N})$ be states with $S_0 \Rightarrow S_1 \Rightarrow \cdots \Rightarrow S_n$ and S_0 the initial state. Then

- For any total assignment $\beta : V \to \mathbb{Q}$: $[\![\mathcal{L}_i \cap \mathcal{N}]\!]^\beta \neq \mathsf{false} \iff [\![\mathcal{L}_{i+1} \wedge \mathcal{N}]\!]^\beta \neq \mathsf{false}$.
- If no rule is applicable to S_n, then S_n is conflict-free iff $\mathcal{C}$ has a solution.

Some properties of the ksmt calculus

Lemma

Let $\mathcal{C}$ be a formula in separated linear form, let $S_i = (\alpha_i, \mathcal{L}_i, \mathcal{N})$ be states with $S_0 \Rightarrow S_1 \Rightarrow \cdots \Rightarrow S_n$ and S_0 the initial state. Then

- For any total assignment $\beta : V \to \mathbb{Q}$: $[\![\mathcal{L}_i \cap \mathcal{N}]\!]^\beta \neq$ false $\iff$ $[\![\mathcal{L}_{i+1} \wedge \mathcal{N}]\!]^\beta \neq$ false.
- If no rule is applicable to S_n, then S_n is conflict-free iff $\mathcal{C}$ has a solution.

Corollary (Soundness)

If no rule can be applied to $(\alpha, \mathcal{L}, \mathcal{N})$, then $\begin{cases} \alpha \text{ is a solution to } \mathcal{C}, & \text{if } [\![\mathcal{L}]\!]^\alpha = \text{true}, \\ \mathcal{C} \text{ has no solution}, & \text{if } [\![\mathcal{L}]\!]^\alpha = \text{false}. \end{cases}$

Some properties of the ksmt calculus

Lemma

Let $\mathcal{C}$ be a formula in separated linear form, let $S_i = (\alpha_i, \mathcal{L}_i, \mathcal{N})$ be states with $S_0 \Rightarrow S_1 \Rightarrow \cdots \Rightarrow S_n$ and S_0 the initial state. Then

- For any total assignment $\beta : V \to \mathbb{Q}$: $[\![\mathcal{L}_i \cap \mathcal{N}]\!]^\beta \neq$ false $\iff$ $[\![\mathcal{L}_{i+1} \wedge \mathcal{N}]\!]^\beta \neq$ false.
- If no rule is applicable to S_n, then S_n is conflict-free iff $\mathcal{C}$ has a solution.

Corollary (Soundness)

If no rule can be applied to $(\alpha, \mathcal{L}, \mathcal{N})$, then $\begin{cases} \alpha \text{ is a solution to } \mathcal{C}, & \text{if } [\![\mathcal{L}]\!]^\alpha = \text{true}, \\ \mathcal{C} \text{ has no solution}, & \text{if } [\![\mathcal{L}]\!]^\alpha = \text{false}. \end{cases}$

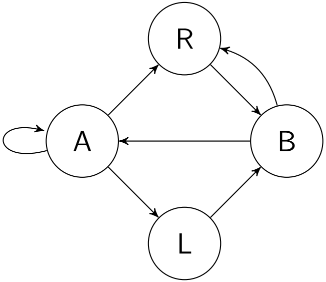
Lemma

Whenever (A) $\to$ (A) is not applicable, the search space is reduced.

Computability?

Decide which rule is applicable:

(1)	$[\![\mathcal{L}]\!]^\alpha = \mathsf{false}$	linearly inconsistent?
	? rule (B)	backjump
(2)	: $[\![\mathcal{N}]\!]^\alpha = \mathsf{false}$	non-linearly inconsistent?
	? rule (L)	linearise
(3)	: $\exists q : [\![\mathcal{L}]\!]^{\alpha::x\mapsto q} \neq \mathsf{false}$	can x be assigned?
	? rule (A)	extend assignment
	: rule (R)	resolve linear conflict

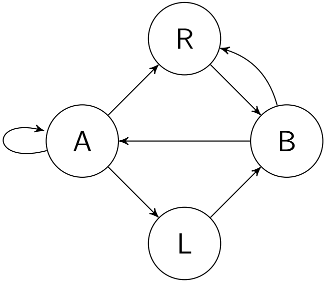


Transitions of rule applicability.

Computability?

Decide which rule is applicable:

(1)	$[\![\mathcal{L}]\!]^\alpha = \text{false}$	linearly inconsistent?
	? rule (B)	backjump
(2)	: $[\![\mathcal{N}]\!]^\alpha = \text{false}$	non-linearly inconsistent?
	? rule (L)	linearise
(3)	: $\exists q : [\![\mathcal{L}]\!]^{\alpha::x \mapsto q} \neq \text{false}$	can x be assigned?
	? rule (A)	extend assignment
	: rule (R)	resolve linear conflict



Transitions of rule applicability.

(1), (3) decidable in polynomial time.

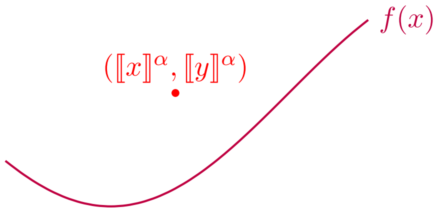
All rule applications computable in polynomial time.

(2) decidability *unknown* in general...

Linearisation required? Decide $[\![\mathcal{N}]\!]^\alpha$!

Schematic

$\underbrace{f(x) \geq y}_{P}$, $\alpha : V \to \mathbb{Q}$

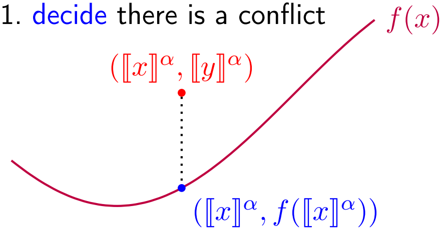


Linearisation required? Decide $[\![\mathcal{N}]\!]^\alpha$!

Schematic

$\underbrace{f(x) \geq y}_{P}$, $\alpha : V \to \mathbb{Q}$;

1. decide there is a conflict

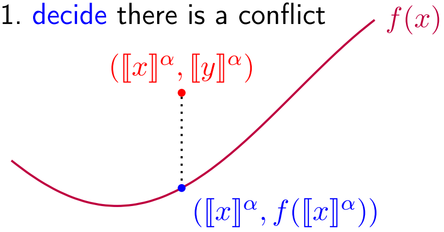


Linearisation required? Decide $[\![\mathcal{N}]\!]^\alpha$!

Schematic

$\underbrace{f(x) \geq y}_{P}$, $\alpha : V \to \mathbb{Q}$;

1. decide there is a conflict



e.g. $\exp(\exp(q)) \in \mathbb{Q}$ for $q \in \mathbb{Q}$ in general is unkown.

There is not enough theory to determine computability of predicates on general terms over transcendental functions.

- connected to open problems in transcendental number theory, e.g. Schanuel's conjecture

However...

There is a non-symbolic theory

Computable Analysis: theory of computations on continuous structures: $\mathbb{R}$, $C([0,1],\mathbb{R})$, ...
via representations [Weihrauch, 2000] of sets of continuum cardinality.

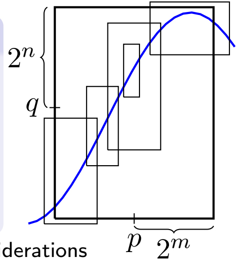
Definition (Cauchy representation of $\mathbb{R}$)

A real number $x \in \mathbb{R}$ is computable iff $\tilde{x} : \mathbb{N} \to \mathbb{Q}$ is computable with

$$\forall n : |\tilde{x}(n) - x| \leq 2^n.$$

Consequence:
If $x \neq y$ is holds, $x < y$ is decidable for computable $x, y \in \mathbb{R}$.
Premise is not decidable.

Computable Analysis (contd.)



Definition

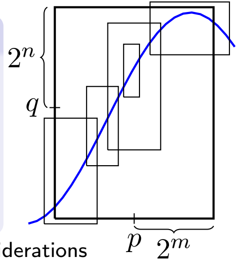
A continuous (partial) function $f : \mathbb{R} \to \mathbb{R}$ is called computable iff the set

$$\{(p, m, q, n) : f([p \pm 2^m]) \subseteq [q \pm 2^n],\ p, q \in \mathbb{Q},\ n, m \in \mathbb{Z}\}$$

is computably enumerable, where $f(I) = \{f(x) : x \in I \cap \operatorname{dom} f\}$ for $I \subseteq \mathbb{R}$.

There are friendlier, but computably equivalent, representations even allowing for complexity considerations [Ko, 1991, Weihrauch, 2000, Kawamura and Cook, 2012], e.g. [Brauße and Steinberg, 2017].

Computable Analysis (contd.)



Definition

A continuous (partial) function $f : \mathbb{R} \to \mathbb{R}$ is called computable iff the set

$$\{(p, m, q, n) : f([p \pm 2^m]) \subseteq [q \pm 2^n],\ p, q \in \mathbb{Q},\ n, m \in \mathbb{Z}\}$$

is computably enumerable, where $f(I) = \{f(x) : x \in I \cap \operatorname{dom} f\}$ for $I \subseteq \mathbb{R}$.

There are friendlier, but computably equivalent, representations even allowing for complexity considerations [Ko, 1991, Weihrauch, 2000, Kawamura and Cook, 2012], e.g. [Brauße and Steinberg, 2017].

Exact Real Arithmetic (ERA): efficient implementations of Computable Analysis, e.g. `iRRAM` [Müller, 2001] (uses a different representation).

- C++ library, LGPL-2, provides types `REAL`, `DYADIC` (based on MPFR), and a non-discrete boolean-like type to represent the result of $x < y$.
- elementary transcendental functions implemented, as well as a limit operator.

Definition

Let $\mathcal{F}_{\text{DA}}$ be the class of functions $g : \mathbb{R}^n \to \mathbb{R}$ with

- $\operatorname{dom} g \cap \mathbb{Q}^n$ decidable,
- $\operatorname{graph} g \cap \mathbb{Q}^n \times \mathbb{Q}$ decidable and
- $g|_{\mathbb{R}^n \setminus (\mathbb{Q}^n \cap g^{-1}(\mathbb{Q}))}$ computable.

Definition

Let $\mathcal{F}_{\mathsf{DA}}$ be the class of functions $g : \mathbb{R}^n \to \mathbb{R}$ with

- $\operatorname{dom} g \cap \mathbb{Q}^n$ decidable,
- $\operatorname{graph} g \cap \mathbb{Q}^n \times \mathbb{Q}$ decidable and
- $g|_{\mathbb{R}^n \setminus (\mathbb{Q}^n \cap g^{-1}(\mathbb{Q}))}$ computable.

- All multivariate polynomials
- Many elementary transcendental fn, e.g. $\exp, \ln, \log_q, \sin, \cos, \tan, \arctan$
- Many discontinuous fn, e.g. piecewise polynomials defined over a decidable set of rational intervals.

Definition

Let $\mathcal{F}_{\mathsf{DA}}$ be the class of functions $g : \mathbb{R}^n \to \mathbb{R}$ with

- $\operatorname{dom} g \cap \mathbb{Q}^n$ decidable,
- $\operatorname{graph} g \cap \mathbb{Q}^n \times \mathbb{Q}$ decidable and
- $g|_{\mathbb{R}^n \setminus (\mathbb{Q}^n \cap g^{-1}(\mathbb{Q}))}$ computable.

- All multivariate polynomials
- Many elementary transcendental fn, e.g. $\exp, \ln, \log_q, \sin, \cos, \tan, \arctan$
- Many discontinuous fn, e.g. piecewise polynomials defined over a decidable set of rational intervals.

Lemma

Let P be a predicate over reals and let α assign all variables in P to rationals. If P is linear or $P : (x \diamond f(\boldsymbol{t}))$ with $[\![f(\boldsymbol{t})]\!] \in \mathcal{F}_{\mathsf{DA}}$, then $[\![P]\!]^\alpha$ is computable.

Definition

Let $\mathcal{F}_{\mathsf{DA}}$ be the class of functions $g : \mathbb{R}^n \to \mathbb{R}$ with

- $\operatorname{dom} g \cap \mathbb{Q}^n$ decidable,
- $\operatorname{graph} g \cap \mathbb{Q}^n \times \mathbb{Q}$ decidable and
- $g|_{\mathbb{R}^n \setminus (\mathbb{Q}^n \cap g^{-1}(\mathbb{Q}))}$ computable.

- All multivariate polynomials
- Many elementary transcendental fn, e.g. $\exp, \ln, \log_q, \sin, \cos, \tan, \arctan$
- Many discontinuous fn, e.g. piecewise polynomials defined over a decidable set of rational intervals.

Lemma

Let P be a predicate over reals and let α assign all variables in P to rationals. If P is linear or $P : (x \diamond f(\boldsymbol{t}))$ with $[\![f(\boldsymbol{t})]\!] \in \mathcal{F}_{\mathsf{DA}}$, then $[\![P]\!]^\alpha$ is computable.

If all $P \in \mathcal{C}$ satisfy the condition of this Lemma, applicability of rule (L) is computable.

For the above polynomials, elementary transcendental and discontinuous functions, $[\![\mathcal{N}]\!]^\alpha$ can be checked in polynomial time if their decision procedures for domain and graph are polynomial-time.
$\leadsto$ Then also the transition relation $\Rightarrow$ is computable in polynomial time.

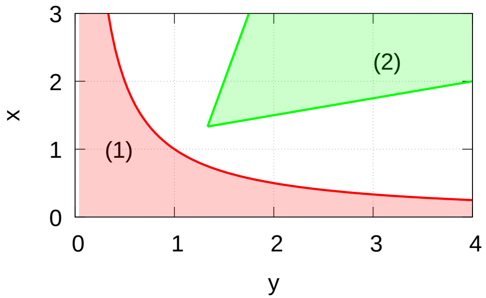
unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \leq 1/y)}_{P} \wedge (x \geq y/4 + 1) \wedge (x \leq 4 \cdot (y - 1))$$

Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \leq d), (y \leq 1/d)\}$

rule	α	note



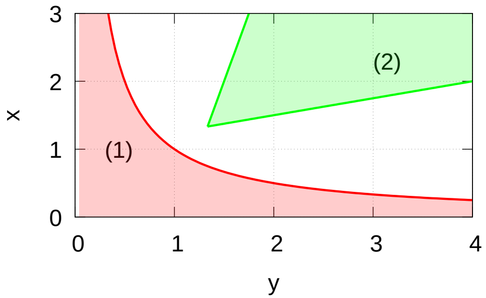
unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4 + 1) \wedge (x \le 4 \cdot (y-1))$$

Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	



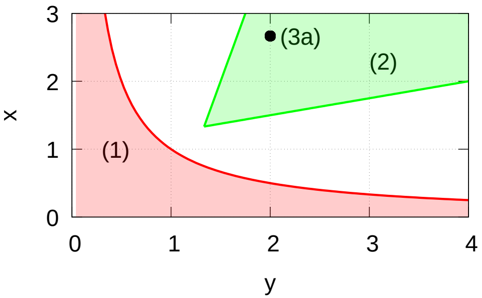
unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4 + 1) \wedge (x \le 4 \cdot (y - 1))$$

Linearisation of conflict (c_x, c_y) here:

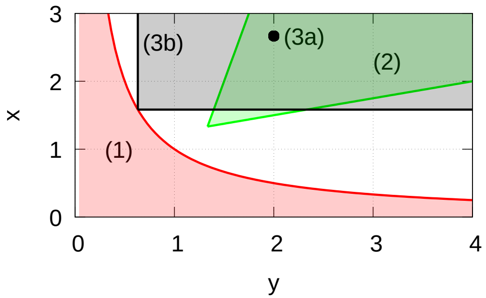
- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)



unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4+1) \wedge (x \le 4\cdot(y-1)) \wedge \left((x \le \tfrac{19}{12}) \vee (y \le \tfrac{12}{19})\right)$$



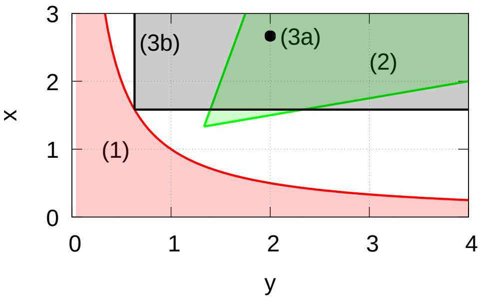
Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)
(L)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3b)

unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4 + 1) \wedge (x \le 4 \cdot (y - 1)) \wedge \left((x \le \tfrac{19}{12}) \vee (y \le \tfrac{12}{19})\right)$$



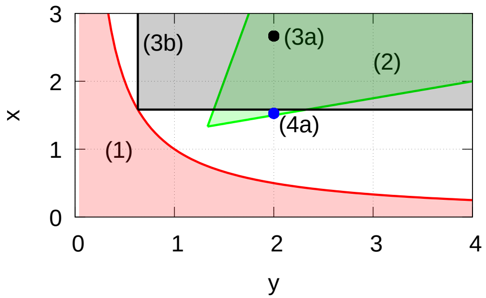
Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)
(L)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3b)
(B)	$y \mapsto 2$	

unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4 + 1) \wedge (x \le 4 \cdot (y-1)) \wedge \left((x \le \tfrac{19}{12}) \vee (y \le \tfrac{12}{19})\right)$$



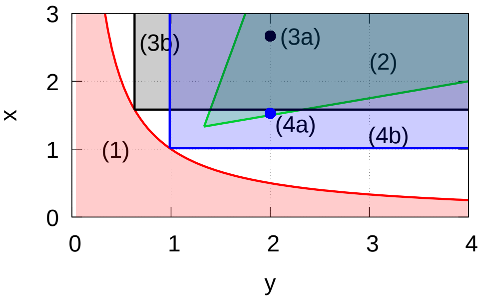
Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)
(L)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3b)
(B)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4a)

unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4 + 1) \wedge (x \le 4 \cdot (y - 1))$$
$$\wedge \left((x \le \tfrac{19}{12}) \vee (y \le \tfrac{12}{19})\right)$$
$$\wedge \left((x \le \tfrac{223}{220}) \vee (y \le \tfrac{220}{223})\right)$$



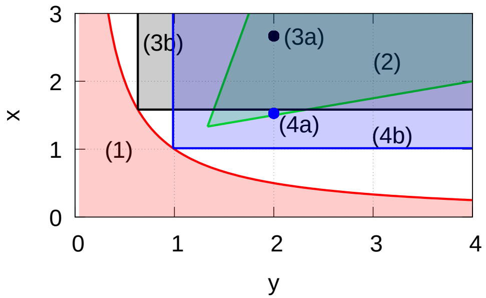
Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)
(L)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3b)
(B)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4a)
(L)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4b)

unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{\color{red}{(x \le 1/y)}}_{P} \wedge \color{green}{(x \ge y/4 + 1) \wedge (x \le 4 \cdot (y-1))}$$
$$\wedge \left((x \le \tfrac{19}{12}) \vee (y \le \tfrac{12}{19})\right)$$
$$\wedge \color{blue}{\left((x \le \tfrac{223}{220}) \vee (y \le \tfrac{220}{223})\right)}$$



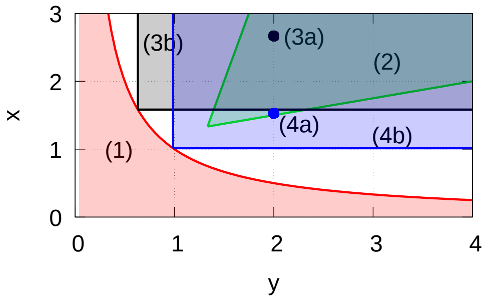
Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)
(L)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3b)
(B)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4a)
(L)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4b)
(B)	$y \mapsto 2$	

unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4 + 1) \wedge (x \le 4 \cdot (y-1))$$
$$\wedge \left((x \le \tfrac{19}{12}) \vee (y \le \tfrac{12}{19})\right)$$
$$\wedge \left((x \le \tfrac{223}{220}) \vee (y \le \tfrac{220}{223})\right)$$
$$\wedge (\tfrac{4}{3} \le y) \wedge (\tfrac{3}{55} \ge y)$$



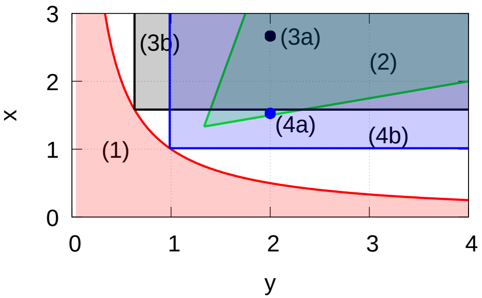
Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)
(L)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3b)
(B)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4a)
(L)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4b)
(B)	$y \mapsto 2$	
(R)	$y \mapsto 2$	on x

unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4 + 1) \wedge (x \le 4 \cdot (y-1))$$
$$\wedge \left((x \le \tfrac{19}{12}) \vee (y \le \tfrac{12}{19})\right)$$
$$\wedge \left((x \le \tfrac{223}{220}) \vee (y \le \tfrac{220}{223})\right)$$
$$\wedge (\tfrac{4}{3} \le y) \wedge (\tfrac{3}{55} \ge y)$$



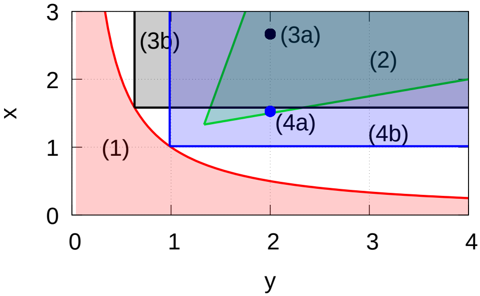
Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)
(L)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3b)
(B)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4a)
(L)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4b)
(B)	$y \mapsto 2$	
(R)	$y \mapsto 2$	on x
(B)		

unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4+1) \wedge (x \le 4\cdot(y-1))$$
$$\wedge \left((x \le \tfrac{19}{12}) \vee (y \le \tfrac{12}{19})\right)$$
$$\wedge \left((x \le \tfrac{223}{220}) \vee (y \le \tfrac{220}{223})\right)$$
$$\wedge (\tfrac{4}{3} \le y) \wedge (\tfrac{3}{55} \ge y)$$
$$\wedge (\tfrac{4}{3} \le \tfrac{3}{55})$$



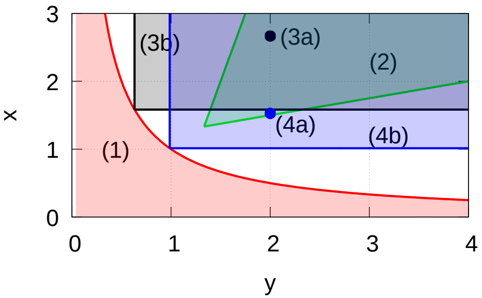
Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)
(L)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3b)
(B)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4a)
(L)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4b)
(B)	$y \mapsto 2$	
(R)	$y \mapsto 2$	on x
(B)		
(R)		on y

unsat example run using Interval linearisation

$$\mathcal{C} = \underbrace{(x \le 1/y)}_{P} \wedge (x \ge y/4 + 1) \wedge (x \le 4 \cdot (y-1))$$
$$\wedge \left((x \le \tfrac{19}{12}) \vee (y \le \tfrac{12}{19})\right)$$
$$\wedge \left((x \le \tfrac{223}{220}) \vee (y \le \tfrac{220}{223})\right)$$
$$\wedge (\tfrac{4}{3} \le y) \wedge (\tfrac{3}{55} \ge y)$$
$$\wedge (\tfrac{4}{3} \le \tfrac{3}{55})$$



Linearisation of conflict (c_x, c_y) here:

- choose $d := (c_x + 1/c_y)/2$,
- $L_{\alpha,P} = \{(x \le d), (y \le 1/d)\}$

rule	α	note
(A)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3a)
(L)	$y \mapsto 2,\ x \mapsto \frac{8}{3}$	(3b)
(B)	$y \mapsto 2$	
(A)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4a)
(L)	$y \mapsto 2,\ x \mapsto \frac{84}{55}$	(4b)
(B)	$y \mapsto 2$	
(R)	$y \mapsto 2$	on x
(B)		
(R)		on y
n/a		unsat

Types of Linearisations

Let $P : (x \diamond f(\boldsymbol{t})) \in \mathcal{N}$, $g = [\![f(\boldsymbol{t})]\!] : \mathbb{R}^n \to \mathbb{R}$ with $g \in \mathcal{F}_{\mathsf{DA}}$ and where $\boldsymbol{y}$ is the vector of free variables in $\boldsymbol{t}$. Let α be an assignment total for P and $(c_x, \boldsymbol{c_y}) = ([\![x]\!]^\alpha, [\![\boldsymbol{y}]\!]^\alpha)$ be a conflict, i.e., $(c_x \diamond g(\boldsymbol{c_y}))$ is false.

Point linearisation: $(\boldsymbol{y} = \boldsymbol{c_y} \implies x \neq c_x)$

Half-Line linearisation: $(\boldsymbol{y} = \boldsymbol{c_y} \implies \neg(x \diamond c_x))$

Interval linearisation: $(\boldsymbol{y} \in R \implies x \diamond d)$

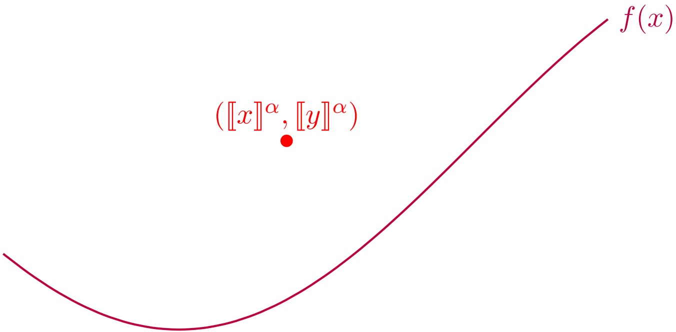
where R is a polytope and d between c_x and $g(\boldsymbol{c_y})$ found by interval arithmetic (due to computability of g).

Tangent space linearisation: $(\boldsymbol{y} \in R \implies x \diamond d + \boldsymbol{c}_\partial \cdot (\boldsymbol{y} - \boldsymbol{c_y}))$

where R is a polytope, d between c_x and $g(\boldsymbol{c_y})$, and c_∂ is an approximation to $\nabla g(\boldsymbol{c_y})$ found by interval arithmetic (due to computability of g).

Tangent space linearisation (schematic)

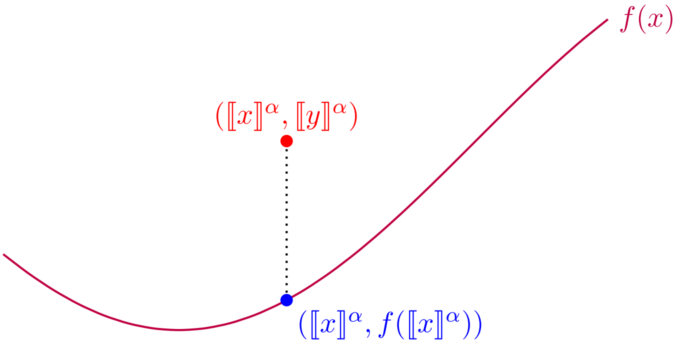
$\underbrace{f(x) \geq y}_{P}, \ \alpha : V \to \mathbb{Q}$



Tangent space linearisation (schematic)

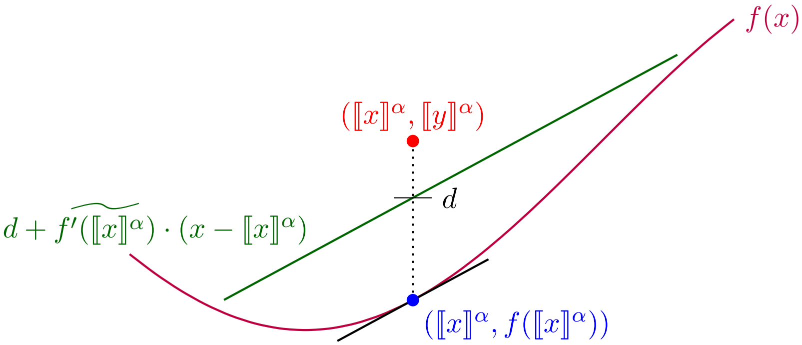
$\underbrace{f(x) \geq y}_{P}$, $\alpha : V \to \mathbb{Q}$;

1. decide there is a conflict



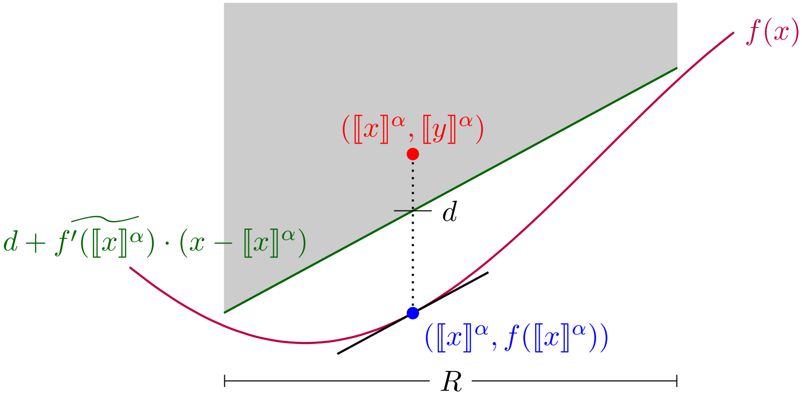
Tangent space linearisation (schematic)

$\underbrace{f(x) \geq y}_{P}$, $\alpha : V \to \mathbb{Q}$;   1. decide there is a conflict;   2. compute linearization



Tangent space linearisation (schematic)

$\underbrace{f(x) \geq y}_{P}$, $\alpha : V \to \mathbb{Q}$; 1. decide there is a conflict; 2. compute linearization

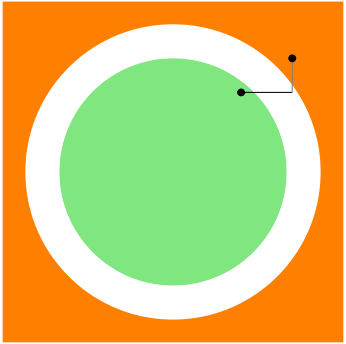


Specialised linearisation algorithms for specific combinations of subclasses of functions $g \in \mathcal{F}_{\mathsf{DA}}$ and $\boldsymbol{c_y}$:

- **Differentiable g:** Use Tangent Space Linearisation.
- **Convex/Concave g:** Derive polytope R from computability of unique intersections between g and the linear bound on $\boldsymbol{y}$.
- **Piecewise g:** Meta-class: $\operatorname{dom} g$ partitioned into $(P_i)_i$ where P_i are linear or non-linear predicates in $\boldsymbol{y}$ and for each i there is a linearisation algorithm; decision on which to use is based on membership of exact rational $\boldsymbol{c_y}$ in one of P_i.
- **Rational $c_g = g(\boldsymbol{c_y})$:** Evaluate c_g exactly in order to determine which linearisation to use.
- **Irrational c_g:** Bound $|c_x - c_g|$ by a rational from below via approximating c_g by the ERA implementation `iRRAM` in order to compute d.

BB: $\exists \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 \leq r \wedge \|\boldsymbol{y}\|_2 \geq \sqrt{64} \wedge \|\boldsymbol{x} - \boldsymbol{y}\|_\infty \leq \frac{1}{100}$.

BB: $\exists \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 \leq r \wedge \|\boldsymbol{y}\|_2 \geq \sqrt{64} \wedge \|\boldsymbol{x} - \boldsymbol{y}\|_\infty \leq \frac{1}{100}$.



BB: $\exists \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 \le r \wedge \|\boldsymbol{y}\|_2 \ge \sqrt{64} \wedge \|\boldsymbol{x} - \boldsymbol{y}\|_\infty \le \frac{1}{100}$.

s : 'sat'
δ : 'δ-sat', $\delta = 10^{-3}$
u : 'unsat'
? : 'unknown'
$>$: timeout

r	ksmt		cvc4		z3		mathsat		yices		dReal		raSAT	
$\sqrt{37}$	u	0.07s	u	0.76s	u	510.67s	u	40.55s	u	0.07s	u	0.01	$>$	8h
$\sqrt{49}$	u	0.40s	u	2.46s	u	23211.20s	u	6307.18s	u	0.11s	u	0.03	$>$	8h
$\sqrt{62}$	u	11.61s	u	5.07s	u	210.16s	$>$	14.5h	u	76.82s	u	2.00	$>$	8h
$\sqrt{63}$	u	55.84s	?	0.48s	u	3925.65s	$>$	14.5h	u	0.10s	u	12.38	$>$	8h
$\sqrt{64}$	s	0.01s	?	0.01s	s	0.00s	$>$	21.6h	s	0.00s	δ	0.01	$>$	8h

Timings on Linux, Core-i7 3.6GHz, (all except `mathsat`: g++-7.3.0), `ksmt` submitted to FroCoS'19 [Brauße et al., 2019], `http://informatik.uni-trier.de/~brausse/ksmt/`

BB: $\exists \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 \leq r \wedge \|\boldsymbol{y}\|_2 \geq \sqrt{64} \wedge \|\boldsymbol{x} - \boldsymbol{y}\|_\infty \leq \frac{1}{100}$.

s : 'sat'
δ : 'δ-sat', $\delta = 10^{-3}$
u : 'unsat'
? : 'unknown'
>: timeout

r	ksmt		cvc4		z3		mathsat		yices		dReal		raSAT	
$\sqrt{37}$	u	0.07s	u	0.76s	u	510.67s	u	40.55s	u	0.07s	u	0.01	>	8h
$\sqrt{49}$	u	0.40s	u	2.46s	u	23211.20s	u	6307.18s	u	0.11s	u	0.03	>	8h
$\sqrt{62}$	u	11.61s	u	5.07s	u	210.16s	>	14.5h	u	76.82s	u	2.00	>	8h
$\sqrt{63}$	u	55.84s	?	0.48s	u	3925.65s	>	14.5h	u	0.10s	u	12.38	>	8h
$\sqrt{64}$	s	0.01s	?	0.01s	s	0.00s	>	21.6h	s	0.00s	δ	0.01	>	8h

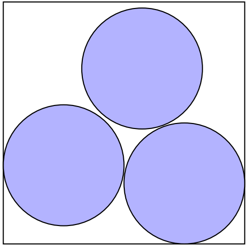
KK: $\exists \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d : \bigwedge_{1 \leq i \leq n} \|\boldsymbol{x}_i\|_\infty \leq 1 \wedge \bigwedge_{1 \leq i < j \leq n} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 > 2$

BB: $\exists \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 \leq r \wedge \|\boldsymbol{y}\|_2 \geq \sqrt{64} \wedge \|\boldsymbol{x} - \boldsymbol{y}\|_\infty \leq \frac{1}{100}$.

- s : ‘sat’
- δ : ’δ-sat’, $\delta = 10^{-3}$
- u : ‘unsat’
- ? : ‘unknown’
- $>$: timeout

r	ksmt	cvc4	z3	mathsat	yices	dReal	raSAT
$\sqrt{37}$	u 0.07s	u 0.76s	u 510.67s	u 40.55s	u 0.07s	u 0.01	$>$ 8h
$\sqrt{49}$	u 0.40s	u 2.46s	u 23211.20s	u 6307.18s	u 0.11s	u 0.03	$>$ 8h
$\sqrt{62}$	u 11.61s	u 5.07s	u 210.16s	$>$ 14.5h	u 76.82s	u 2.00	$>$ 8h
$\sqrt{63}$	u 55.84s	? 0.48s	u 3925.65s	$>$ 14.5h	u 0.10s	u 12.38	$>$ 8h
$\sqrt{64}$	s 0.01s	? 0.01s	s 0.00s	$>$ 21.6h	s 0.00s	δ 0.01	$>$ 8h

KK: $\exists \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d : \bigwedge_{1 \leq i \leq n} \|\boldsymbol{x}_i\|_\infty \leq 1 \wedge \bigwedge_{1 \leq i < j \leq n} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 > 2$



BB: $\exists \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 \leq r \wedge \|\boldsymbol{y}\|_2 \geq \sqrt{64} \wedge \|\boldsymbol{x} - \boldsymbol{y}\|_\infty \leq \frac{1}{100}$.

s: 'sat'
δ: 'δ-sat', $\delta = 10^{-3}$
u: 'unsat'
?: 'unknown'
>: timeout

r	ksmt		cvc4		z3		mathsat		yices		dReal		raSAT	
$\sqrt{37}$	u	0.07s	u	0.76s	u	510.67s	u	40.55s	u	0.07s	u	0.01	>	8h
$\sqrt{49}$	u	0.40s	u	2.46s	u	23211.20s	u	6307.18s	u	0.11s	u	0.03	>	8h
$\sqrt{62}$	u	11.61s	u	5.07s	u	210.16s	>	14.5h	u	76.82s	u	2.00	>	8h
$\sqrt{63}$	u	55.84s	?	0.48s	u	3925.65s	>	14.5h	u	0.10s	u	12.38	>	8h
$\sqrt{64}$	s	0.01s	?	0.01s	s	0.00s	>	21.6h	s	0.00s	δ	0.01	>	8h

KK: $\exists \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^d : \bigwedge_{1 \leq i \leq n} \|\boldsymbol{x}_i\|_\infty \leq 1 \wedge \bigwedge_{1 \leq i < j \leq n} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 > 2$

d	n	ksmt		cvc4		z3		mathsat		yices		dreal		rasat	
2	2	s	0.01s	?	0.03s	s	0.01s	s	0.02s	s	0.01s	δ	0.01	s	0.02
	3	s	0.03s	?	0.08s	>	60m	s	0.24s	s	0.03s	δ	0.02	>	8h
	4	>	8h	u	1474.16s	>	60m	u	8.11s	>	17h	δ	0.05	>	8h
	5	u	1.43s	u	0.45s	>	8h	u	0.28s	>	8h	u	3581.96	>	8h
	6	u	5.00s	u	0.75s	>	8h	u	0.40s	>	166m	>	8h	>	8h
3	5	s	0.93s	?	465.45s	>	8h	s	0.12s	s	0.06s	>	8h	>	8h
	6	s	6.02s	>	143m	>	7h	>	8h	>	6h	>	8h	>	8h
4	5	s	0.38s	?	1544.87s	s	2165.78s	s	0.10s	s	7.34s	>	8h	>	8h
	6	s	0.57s	>	91m	>	8h	s	0.23s	s	0.38s	>	8h	>	8h
	7	s	14.27s	>	160m	>	8h	s	0.18s	>	8h	>	8h	>	8h

Timings on Linux, Core-i7 3.6GHz, (all except `mathsat`: `g++-7.3.0`), `ksmt` submitted to FroCoS'19 [Brauße et al., 2019], `http://informatik.uni-trier.de/~brausse/ksmt/`

Conclusions and future work

Sound `ksmt` calculus:

- model-guided search & resolution of non-linear conflicts via local linearisation
- prototypical implementation with promising results
- identified broad class of functions supported by this approach

Future:

- more precise linearisations for specific functions
- analyze complexity of deciding conflicts
- more extensive evaluation
- theoretical properties of calculus:
  - completeness in restricted settings
  - δ-completeness

Brauße, F., Korovin, K., Korovina, M. V., and Müller, N. T. (2019).
A CDCL-style calculus for solving non-linear constraints.
CoRR, abs/1905.09227.
submitted to FroCoS 2019.

Brauße, F. and Steinberg, F. (2017).
A minimal representation for continuous functions.
CoRR, abs/1703.10044.

Hales, T. C., Adams, M., Bauer, G., Dang, D. T., Harrison, J., Hoang, T. L., Kaliszyk, C., Magron, V., McLaughlin, S., Nguyen, T. T., Nguyen, T. Q., Nipkow, T., Obua, S., Pleso, J., Rute, J. M., Solovyev, A., Ta, A. H. T., Tran, T. N., Trieu, D. T., Urban, J., Vu, K. K., and Zumkeller, R. (2015).
A formal proof of the kepler conjecture.
CoRR, abs/1501.02155.

Kawamura, A. and Cook, S. (2012).
Complexity theory for operators in analysis.
ACM Trans. Comput. Theory, 4(2):5:1–5:24.

Ko, K.-I. (1991).
Complexity theory of real functions.
Birkhauser Boston Inc., Cambridge, MA, USA.

Korovin, K., Tsiskaridze, N., and Voronkov, A. (2009).
Conflict resolution.
In *CP'09, LNCS v.5732*, pages 509–523.

Müller, N. T. (2001).
The iRRAM: Exact arithmetic in C++.
In *Computability and Complexity in Analysis, 4th International Workshop, CCA 2000, Swansea, UK, September 17-19, 2000, Selected Papers*, volume 2064 of *LNCS*, pages 222–252. Springer.

Richardson, D. (1968).
Some undecidable problems involving elementary functions of a real variable.
J. Symb. Log., 33(4):514–520.

Tarski, A. (1951).
A decision method for elementary algebra and geometry.
In *2nd ed. Univ. Cal.*

Weihrauch, K. (2000).
Computable analysis: an introduction.
Springer-Verlag New York, Inc., Secaucus, NJ, USA.