A CDCL-style Calculus for Solving Non-linear Constraints [1]

Franz Brauße[2] Konstantin Korovin[2] Margarita Korovina[3] Norbert Th. Müller[1]

[1] University of Trier, Germany

[2] The University of Manchester, UK

[3] IIS, Novosibirsk, Russia

MLA 2021, March 22–24

[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.

Main Problem

Checking satisfiability of (non-)linear constraints.

Linear:

$$\begin{aligned} 5x - 10y - 2z &\leq 1/2 \\ 6x - 1/3y + 2z &> 0 \\ -10x + 5y - 31z &\geq 0 \end{aligned}$$

Main Problem

Checking satisfiability of (non-)linear constraints.

Polynomial:

$$
\begin{aligned}
3x^2y - 10yzx - 2z &\leq 5 \\
-5yz^2 - 1/3y + 2z &> 1 \\
x + 5xy - 11xz &\geq 7
\end{aligned}
$$

Main Problem

Checking satisfiability of (non-)linear constraints.

Non-linear with transcendental functions

$$2\sin^2 x - 5\cos y^2 - 2z \leq 1/2 \quad \vee \quad e^{x^{-2}} + zy < y$$
$$4x - 1/3y + 2zx > 0$$
$$x^2 - y^2 - z \geq 0$$

Motivation:

- Verification: of hybrid; embedded systems; programs etc.
- Proof assistance for mathematics which rely on computations with non-linear constraints such as Hales proof of Kepler's conjecture.

In most cases the problem of solving non-linear constraints is undecidable or relates to open problems in maths.

Overview of our approach

1. separated linear form $\mathcal{L} \wedge \mathcal{N}$
2. ksmt calculus – conflict-driven calculus for solving non-linear constraints
3. local linearisations for resolving non-linear conflicts
    - approximation of non-linear problem by incremental linearisations
    - related work [A. Cimatti, A. Griggio, A. Irfan, M. Roveri, and R. Sebastiani'18;...]
4. δ-complete decision procedure for bounded instances

Overview of our approach

1. separated linear form $\mathcal{L} \wedge \mathcal{N}$
2. ksmt calculus – conflict-driven calculus for solving non-linear constraints
3. local linearisations for resolving non-linear conflicts
   - approximation of non-linear problem by incremental linearisations
   - related work [A. Cimatti, A. Griggio, A. Irfan, M. Roveri, and R. Sebastiani'18;. . .]
4. δ-complete decision procedure for bounded instances

New class $\mathcal{F}_{\mathrm{DA}}$ – functions with decidable rational approximations

- Checking "non-linear conflicts" is decidable for functions in $\mathcal{F}_{\mathrm{DA}}$
- inspired by computable analysis

Overview of our approach

1. separated linear form $\mathcal{L} \wedge \mathcal{N}$
2. ksmt calculus – conflict-driven calculus for solving non-linear constraints
3. local linearisations for resolving non-linear conflicts
   - approximation of non-linear problem by incremental linearisations
   - related work [A. Cimatti, A. Griggio, A. Irfan, M. Roveri, and R. Sebastiani'18;...]
4. δ-complete decision procedure for bounded instances

New class $\mathcal{F}_{\mathrm{DA}}$ – functions with decidable rational approximations

- Checking "non-linear conflicts" is decidable for functions in $\mathcal{F}_{\mathrm{DA}}$
- inspired by computable analysis
- $\mathcal{F}_{\mathrm{DA}}$ includes:
  - Multivariate polynomials x^2yz^5
  - Transcendental functions: $\exp, \ln, \log_b, \sin, \cos, \tan, \arctan \ldots$
  - Discontinuous functions: step-functions; piecewise linear/polynomial functions

From Logic to Arithmetic: The linear case

From Logic to Linear Arithmetic: Resolution

Motivation:
How to extend efficient SAT technology to other domains/theories?

- Black-box: CDCL(T) – separate Boolean structure and theory
- SAT-encodings: bit-vectors etc.
- White-box: extend SAT calculi to other domains

From Logic to Linear Arithmetic: Resolution

propositional	linear arithmetic
clauses	linear inequalities
$\neg x_1 \vee x_2 \vee \cdots \vee x_n$	$-5x_1 + 3x_2 + \cdots + 0.5x_n + 17 \geq 0$
clause resolution	inequality resolution
$\dfrac{\neg x \vee C \quad x \vee D}{C \vee D}$	$\dfrac{-ax + p \geq 0 \quad bx + q \geq 0}{bp + aq \geq 0}$

Combine model search and proof search

Conflict resolution – combination of model search and proof search

- Iteratively assign values (A) to variables $x_1 \mapsto 0 :: x_2 \mapsto 0.2 :: \ldots :: x_n \mapsto 5$
- If all constraints evaluate to true then – done
- Otherwise, we have a conflict
  1. resolve (R)
  2. backjump (B)
  3. refine assignment (A)

Combine model search and proof search

Conflict resolution – combination of model search and proof search

- Iteratively assign values (A) to variables $x_1 \mapsto 0 :: x_2 \mapsto 0.2 :: \ldots :: x_n \mapsto 5$
- If all constraints evaluate to true then – done
- Otherwise, we have a conflict
  1. resolve (R)
  2. backjump (B)
  3. refine assignment (A)

- Conflict Resolution [Korovin, Tsiskaridze, Voronkov, 2009]
- GDPLL [McMillan, Kuehlmann, Sagiv 2009]
- bound propagation [Korovin, Voronkov, 2011]
- MCSAT/NLSAT [Jovanović, de Moura, 2012/2013]
- CDSAT [Bonacina, Graham-Lengran, Shankar, 2017]
- ...

Example

$$
\begin{array}{rrrrrrrrrrrl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 & (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 & (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 & (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 & (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 & (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 & (6)
\end{array}
$$

variable				
bounds				
assignment				

Example

$$
\begin{array}{rcrcrcrcrcrr}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 & (1) \\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 & (2) \\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 & (3) \\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 & (4) \\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 & (5) \\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 & (6)
\end{array}
$$

variable	x_1			
bounds				
assignment				

Example

$$
\begin{array}{rrrrrrrrrrrr}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 & (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 & (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 & (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 & (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 & (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 & (6)
\end{array}
$$

variable	x_1			
bounds	$(-\infty, \infty)$			
assignment				

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)
\end{array}
$$

variable	x_1			
bounds	$(-\infty, \infty)$			
assignment	$x_1 \mapsto 0$			

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)
\end{array}
$$

variable	x_1	x_2		
bounds	$(-\infty, \infty)$			
assignment	$x_1 \mapsto 0$			

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)
\end{array}
$$

variable	x_1	x_2		
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$		
assignment	$x_1 \mapsto 0$			

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)
\end{array}
$$

variable	x_1	x_2	
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)
\end{array}
$$

variable	x_1	x_2	x_3
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad {\color{red}(5)}\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad {\color{blue}(6)}
\end{array}
$$

variable	x_1	x_2	x_3
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[{\color{red}1}; {\color{blue}5}]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)
\end{array}
$$

variable	x_1	x_2	x_3
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 5]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 4$

Example

$$
\begin{array}{rcrcrcrcrcrl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 & (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 & (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 & (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 & (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 & (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 & (6)
\end{array}
$$

variable	x_1	x_2	x_3	x_4
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 5]$	
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 4$	

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1) \\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2) \\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3) \\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4) \\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5) \\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)
\end{array}
$$

variable	x_1	x_2	x_3	x_4
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 5]$	$[3; -3]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 4$	

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)
\end{array}
$$

variable	x_1	x_2	x_3	x_4
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 5]$	$[3; -3]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 4$	

Conflict: $\text{res}_{x_4}((1), (3)) \rightsquigarrow (7)$

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)\\
 & & -x_3 & - & x_2 & - & \frac{2}{3}x_1 & + & 2 & \geq & 0 \quad (7)
\end{array}
$$

variable	x_1	x_2	x_3	x_4
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 5]$	$[3; -3]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 4$	

Conflict: $\text{res}_{x_4}((1), (3)) \rightsquigarrow (7)$

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)\\
 & & -x_3 & - & x_2 & - & \frac{2}{3}x_1 & + & 2 & \geq & 0 \quad (7)
\end{array}
$$

variable	x_1	x_2	x_3
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	

Example

$$
\begin{array}{rcrcrcrcrcrl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 & (1) \\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 & (2) \\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 & (3) \\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 & (4) \\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 & (5) \\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 & (6) \\
 & & -x_3 & - & x_2 & - & \frac{2}{3}x_1 & + & 2 & \geq & 0 & (7)
\end{array}
$$

variable	x_1	x_2	x_3
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 2]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	

Example

$$
\begin{array}{rcrcrcrcrcrl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 & (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 & (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 & (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 & (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 & (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 & (6)\\
 & & -x_3 & - & x_2 & - & \frac{2}{3}x_1 & + & 2 & \geq & 0 & (7)
\end{array}
$$

variable	x_1	x_2	x_3
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 2]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 1$

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)\\
 & & -x_3 & - & x_2 & - & \frac{2}{3}x_1 & + & 2 & \geq & 0 \quad (7)
\end{array}
$$

variable	x_1	x_2	x_3	x_4
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 2]$	
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 1$	

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)\\
 & & -x_3 & - & x_2 & - & \frac{2}{3}x_1 & + & 2 & \geq & 0 \quad (7)
\end{array}
$$

variable	x_1	x_2	x_3	x_4
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 2]$	$[-3; 0]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 1$	

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)\\
 & & -x_3 & - & x_2 & - & \frac{2}{3}x_1 & + & 2 & \geq & 0 \quad (7)
\end{array}
$$

variable	x_1	x_2	x_3	x_4
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 2]$	$[-3; 0]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 1$	$x_4 \mapsto -1$

SAT

Example

$$
\begin{array}{rcrcrcrcrcl}
x_4 & - & 2x_3 & & & + & x_1 & + & 5 & \geq & 0 \quad (1)\\
x_4 & + & 2x_3 & + & x_2 & & & + & 3 & \geq & 0 \quad (2)\\
-x_4 & - & x_3 & - & 3x_2 & - & 3x_1 & + & 1 & \geq & 0 \quad (3)\\
-x_4 & + & 2x_3 & + & 2x_2 & + & x_1 & + & 6 & \geq & 0 \quad (4)\\
 & & x_3 & & & + & 3x_1 & - & 1 & \geq & 0 \quad (5)\\
 & & -x_3 & + & x_2 & - & 2x_1 & + & 5 & \geq & 0 \quad (6)\\
 & & -x_3 & - & x_2 & - & \frac{2}{3}x_1 & + & 2 & \geq & 0 \quad (7)
\end{array}
$$

variable	x_1	x_2	x_3	x_4
bounds	$(-\infty, \infty)$	$(-\infty, \infty)$	$[1; 2]$	$[-3; 0]$
assignment	$x_1 \mapsto 0$	$x_2 \mapsto 0$	$x_3 \mapsto 1$	$x_4 \mapsto -1$

SAT

CR is correct and terminating. [Korovin, Tsiskaridze, Voronkov; CP'09]

Fourier-Motzkin vs Conflict Resolution

Example:

$$
\begin{array}{rcrcrcrcrcrcl}
2x_5 & - & 3x_4 & + & x_3 & - & 3x_2 & - & 2x_1 & + & 3 & \geq & 0 \\
2x_5 & + & x_4 & - & 2x_3 & & & - & 2x_1 & + & 2 & \geq & 0 \\
-x_5 & & & & & + & 3x_2 & + & x_1 & + & 2 & \geq & 0 \\
-3x_5 & & & + & 2x_3 & & & - & 3x_1 & - & 2 & \geq & 0 \\
x_5 & - & 2x_4 & & & - & 2x_2 & + & 3x_1 & - & 2 & \geq & 0 \\
-2x_5 & + & 2x_4 & - & 3x_3 & - & x_2 & + & 2x_1 & + & 3 & > & 0 \\
3x_5 & - & 2x_4 & + & 2x_3 & + & 3x_2 & + & 2x_1 & + & 1 & > & 0 \\
x_5 & & & & & & & + & 2x_1 & + & 2 & > & 0 \\
 & & 2x_4 & - & x_3 & - & 3x_2 & - & x_1 & + & 3 & = & 0
\end{array}
$$

Fourier-Motzkin: Generates over 280 million linear inequalities.

Conflict Resolution: Generates 21 linear inequalities.

ksmt calculus – extending conflict resolution to non-linear constraints

Existentially quantified formula in CNF over predicates over $(\mathbb{R}, \mathcal{F}_{\mathsf{lin}} \cup \mathcal{F}_{\mathsf{nl}}, <, \leq, >, \geq)$.

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

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: $[\![\ell]\!]^\alpha = \mathsf{true}$

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

Overview of the ksmt approach

Main ingredients of our approach:

1. separated linear form $\mathcal{L} \wedge \mathcal{N}$
2. ksmt calculus – conflict-driven calculus for solving non-linear constraints
3. local linearisations for resolving non-linear conflicts

Separated Linear Form

Separated linear form: $\mathcal{L} \wedge \mathcal{N}$

- $\mathcal{L}$ – linear inequalities: $q_1 x_1 + q_2 x_2 + \dots + q_n x_n + q_0 \diamond 0$

$$\begin{aligned} 2x - 4y - 2u - 2 &> 0 \\ -x + 2y + 3u + 1 &> 0 \\ 4y + 2u + 1 &\geq 0 \end{aligned}$$

- $\mathcal{N}$ – non-linear units: $x \diamond f(\bar{t})$

$$\begin{aligned} y &> \sin(x^2) \\ u &\leq y^2 x \\ x &\geq \mathrm{e}^{-u} \end{aligned}$$

Separated Linear Form

Transforming non-linear constraints into separated linear form.

- Monotonic flattening: introduce fresh variables for non-linear terms.

$$5\sin(x^2) - 3xy + 2x - 13 \geq 0 \mapsto$$

Separated Linear Form

Transforming non-linear constraints into separated linear form.

- Monotonic flattening: introduce fresh variables for non-linear terms.

$$5\sin(x^2) - 3xy + 2x - 13 \geq 0 \mapsto$$

$$5x_1 - 3x_2 + 2x - 13 \geq 0$$

$$x_1 \leq \sin(x^2)$$
$$x_2 \geq xy$$

Separated Linear Form

Transforming non-linear constraints into separated linear form.

- Monotonic flattening: introduce fresh variables for non-linear terms.

$$5\sin(x^2) - 3xy + 2x - 13 \geq 0 \mapsto$$

$$5x_1 - 3x_2 + 2x - 13 \geq 0$$
$$x_1 \leq \sin(x^2)$$
$$x_2 \geq xy$$

Lemma. Any CNF over non-linear constraints can be efficiently transformed into separated linear form using monotonic flattening.

ksmt calculus

- Transition rules define relation $\Rightarrow$ on states $(\alpha, \mathcal{L}, \mathcal{N})$.
  - (partial) assignment α
  - linear inequalities $\mathcal{L}$
  - non-linear units $\mathcal{N}$
- initial state is $(\text{nil}, \mathcal{L}_0, \mathcal{N}_0)$ for formula in separated linear form $\mathcal{L}_0 \wedge \mathcal{N}_0$.
- sat and $unsat$ are final states.

$\mathcal{N}_0$ remains unchanged under $\Rightarrow$ transformations.

ksmt calculus

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

ksmt calculus

Rules	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	
(A) assign	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	z unassigned, $q \in \mathbb{Q}$ and no linear conflict

ksmt calculus

Rules	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	
(A) assign	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	z unassigned, $q \in \mathbb{Q}$ and no linear conflict
(R) resolve	$(\alpha, \mathcal{L} \cup R, \mathcal{N})$	R resolvent excluding the linear conflict

ksmt calculus

Rules	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	
(A) assign	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	z unassigned, $q \in \mathbb{Q}$ and no linear conflict
(R) resolve	$(\alpha, \mathcal{L} \cup R, \mathcal{N})$	R resolvent excluding the linear conflict
(B) backjump	$(\gamma, \mathcal{L}, \mathcal{N})$	to γ maximal conflict-free prefix of α

ksmt calculus

Rules	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	
(A) assign	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	z unassigned, $q \in \mathbb{Q}$ and no linear conflict
(R) resolve	$(\alpha, \mathcal{L} \cup R, \mathcal{N})$	R resolvent excluding the linear conflict
(B) backjump	$(\gamma, \mathcal{L}, \mathcal{N})$	to γ maximal conflict-free prefix of α
(L) linearise	$(\alpha, \mathcal{L} \cup L, \mathcal{N})$	L linearisation, excluding the non-linear conflict
(F^{sat})	sat	all variables are assigned, no linear conflict and (A), (R), (B), (L) are not applicable
(F^{unsat})	$unsat$	$[\![\mathcal{L}]\!]^{nil} =$ false

$[\![\cdot]\!]^{\alpha}$ is the partial evaluation under α.

Local linearisations

Definition

Assignment α with non-linear conflict $[\![P]\!]^\alpha = \text{false}$ for a non-linear unit P. A linear clause L is a linearisation if

- for any assignment β, $[\![P]\!]^\beta = \text{true}$ implies $[\![L]\!]^\beta = \text{true}$, and
- $[\![L]\!]^\alpha = \text{false}$.

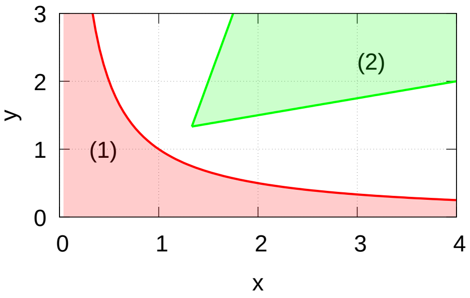
unsat example run using Interval linearisation

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

Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = \left(x \le 1/d \ \vee \ y \le d\right)$

rule	α	note



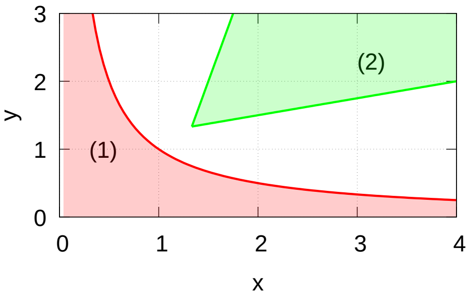
unsat example run using Interval linearisation

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

Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = \left(x \leq 1/d \;\vee\; y \leq d\right)$

rule	α	note
(A)	$x \mapsto 2$	



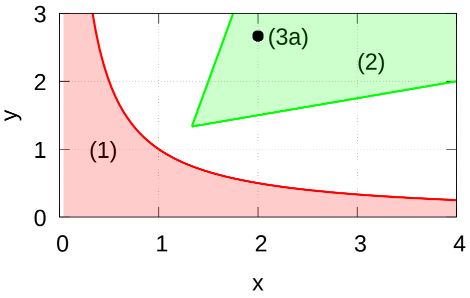
unsat example run using Interval linearisation

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

Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = \big(x \leq 1/d \ \vee \ y \leq d\big)$

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



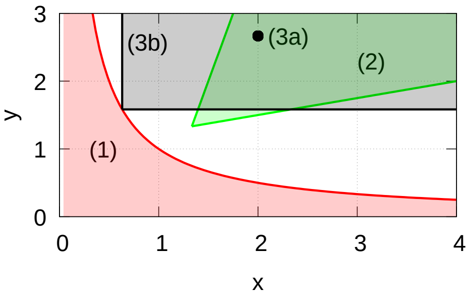
unsat example run using Interval linearisation

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

Linearisation of conflicts (x, y) at α here:

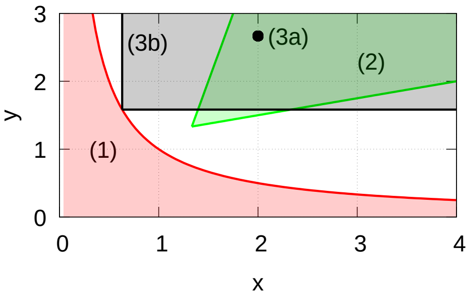
- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = (x \le 1/d \ \lor \ y \le d)$

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



unsat example run using Interval linearisation

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



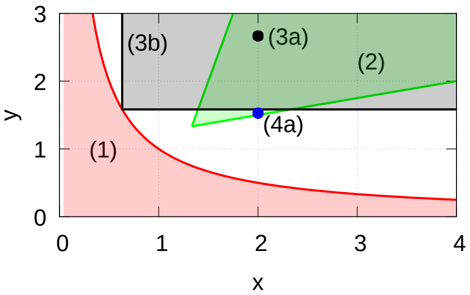
Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = \left(x \le 1/d \;\vee\; y \le d\right)$

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

unsat example run using Interval linearisation

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



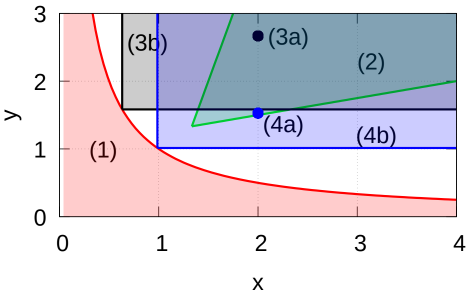
Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = (x \le 1/d \ \vee\ y \le d)$

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

unsat example run using Interval linearisation

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



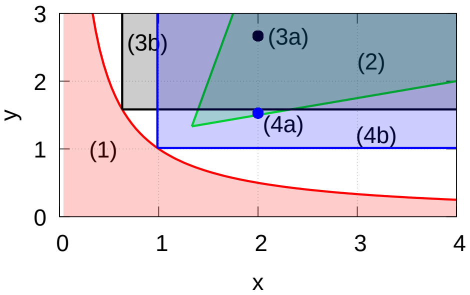
Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = (x \leq 1/d \ \vee \ y \leq d)$

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

unsat example run using Interval linearisation

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



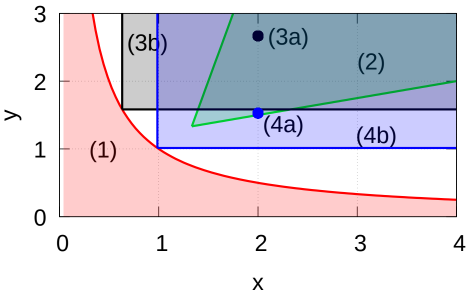
Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = (x \le 1/d \ \vee \ y \le d)$

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

unsat example run using Interval linearisation

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



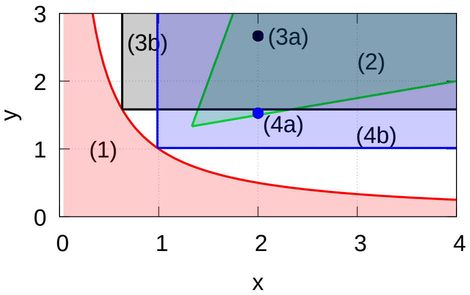
Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = (x \le 1/d \ \vee \ y \le d)$

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

unsat example run using Interval linearisation

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



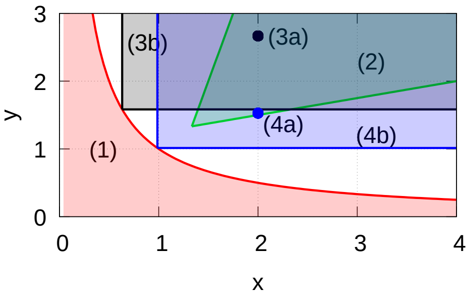
Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = (x \le 1/d \ \vee \ y \le d)$

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

unsat example run using Interval linearisation

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



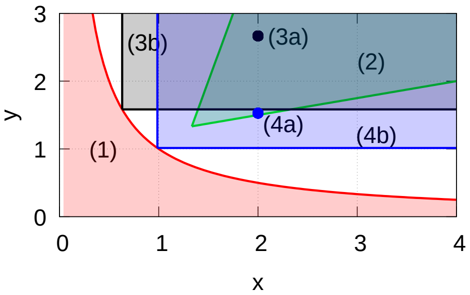
Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = (x \leq 1/d \ \vee \ y \leq d)$

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

unsat example run using Interval linearisation

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

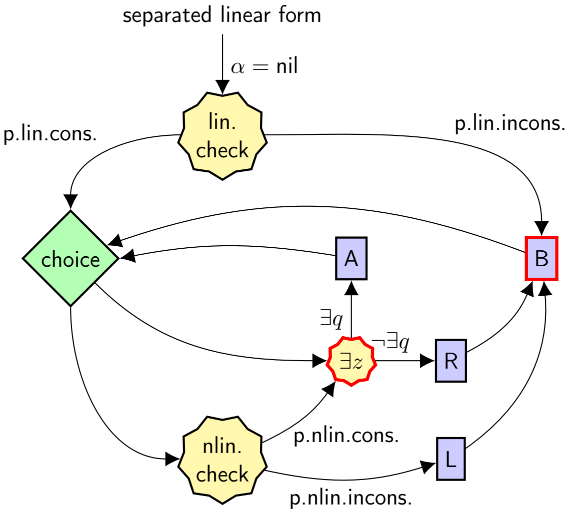


Linearisation of conflicts (x, y) at α here:

- choose $d := (1/[\![x]\!]^\alpha + [\![y]\!]^\alpha)/2$,
- $L = (x \le 1/d \ \vee \ y \le d)$

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

Core of ksmt calculus



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

Corollary (Soundness)

Let $S = (\alpha, \mathcal{L}, \mathcal{N})$ be derivable from $(\text{nil}, \mathcal{L}_0, \mathcal{N})$ in ksmt.

- If (F^{sat}) is applicable to S, then α is a solution to $\mathcal{L}_0 \wedge \mathcal{N}$.
- If (F^{unsat}) is applicable to S, then $\mathcal{L}_0 \wedge \mathcal{N}$ is unsatisfiable.

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

Corollary (Soundness)

Let $S = (\alpha, \mathcal{L}, \mathcal{N})$ be derivable from $(\text{nil}, \mathcal{L}_0, \mathcal{N})$ in ksmt.

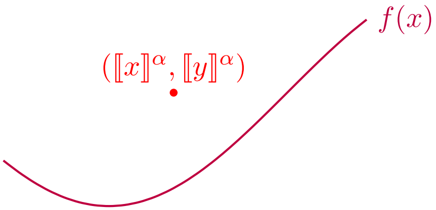
- If (F^{sat}) is applicable to S, then α is a solution to $\mathcal{L}_0 \wedge \mathcal{N}$.
- If (F^{unsat}) is applicable to S, then $\mathcal{L}_0 \wedge \mathcal{N}$ is unsatisfiable.

Lemma (Progress)

After at most $(\#\text{variables} + 2)$ steps the search space is reduced.

Deciding non-linear conflicts

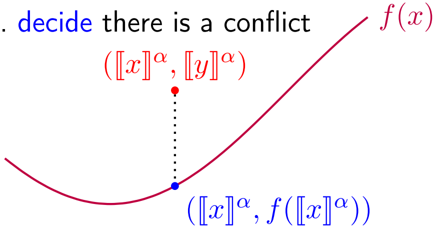
$f(x) \geq y$, $\alpha : V \to \mathbb{Q}$



Deciding non-linear conflicts

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

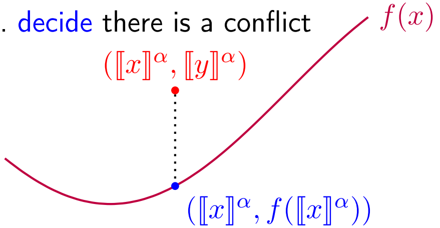
1. decide there is a conflict



Deciding non-linear conflicts

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

1. decide there is a conflict



Computable Analysis: theory of computations on continuous structures: $\mathbb{R}$, $C([0,1],\mathbb{R})$, ...

- efficient implementation: iRRAM [Müller '00]

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

$x \in \mathbb{R}$ is computable iff $\tilde{x} : \mathbb{N} \to \mathbb{Q}$ is computable with $\forall n : |\tilde{x}(n) - x| \leq 2^{-n}$.

In general, $f(\llbracket x \rrbracket^\alpha) \geq \llbracket y \rrbracket^\alpha$ is not decidable, so we need more information about f.

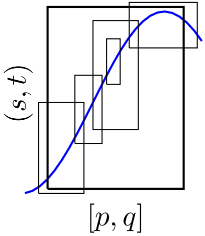
Approximability

Definition

A partial function $f : \mathbb{R} \to \mathbb{R}$ is called approximable iff

$$\{(p, q, s, t) : f([p, q]) \subset (s, t)\} \subset \mathbb{Q}^4$$

is computably enumerable.



Lemma

For total continuous real functions, approximability coincides with the notion of computability known from Computable Analysis.

The class $\mathcal{F}_{\mathrm{DA}}$

Definition

$\mathcal{F}_{\mathrm{DA}}$ – functions with decidable rational approximations; $g \in \mathcal{F}_{\mathrm{DA}}$ if

- $\operatorname{dom} g \cap \mathbb{Q}^n$ decidable,
- $\operatorname{graph} g \cap \mathbb{Q}^n \times \mathbb{Q}$ decidable and
- g approximable.

The class $\mathcal{F}_{\mathrm{DA}}$

Definition

$\mathcal{F}_{\mathrm{DA}}$ – functions with decidable rational approximations; $g \in \mathcal{F}_{\mathrm{DA}}$ if

- $\operatorname{dom} g \cap \mathbb{Q}^n$ decidable,
- $\operatorname{graph} g \cap \mathbb{Q}^n \times \mathbb{Q}$ decidable and
- g approximable.

- 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.

The class $\mathcal{F}_{\mathrm{DA}}$

Definition

$\mathcal{F}_{\mathrm{DA}}$ – functions with decidable rational approximations; $g \in \mathcal{F}_{\mathrm{DA}}$ if

- $\operatorname{dom} g \cap \mathbb{Q}^n$ decidable,
- $\operatorname{graph} g \cap \mathbb{Q}^n \times \mathbb{Q}$ decidable and
- g approximable.

- 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.

Theorem

For functions in $\mathcal{F}_{\mathrm{DA}}$, checking non-linear conflicts is decidable and linearisations are computable.

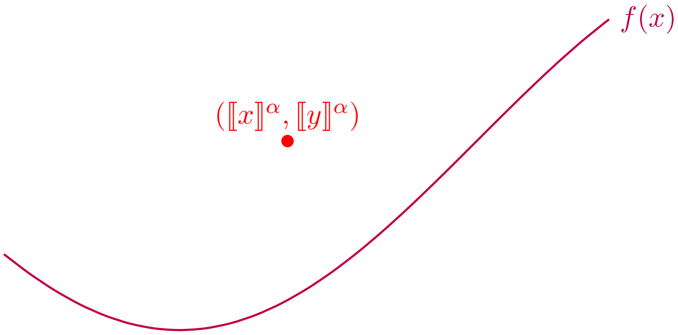
Using functions' known properties

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

- Differentiable g: Use Tangent Space Linearisation.
- Convex/Concave g: Derive polytope R from computability of unique intersections
- Piecewise g: Meta-class: $\operatorname{dom} g$ partitioned by linear or non-linear predicates, each with a linearisation algorithm attached.
- Rational $g(\boldsymbol{x})$: Evaluate exactly in order to determine which linearisation to use.
- Irrational $g(\boldsymbol{x})$: Bound difference from below by a rational via successive approximations by the Computable Analysis implementation `iRRAM`.

Tangent space linearisation (schematic)

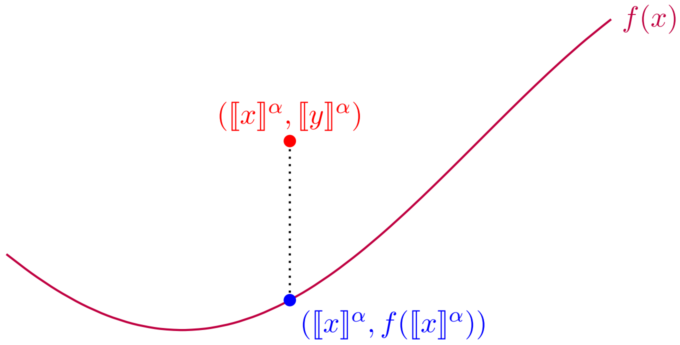
$\underbrace{f(x) \geq y}_{P}$, $\alpha : V \to \mathbb{Q}$



Tangent space linearisation (schematic)

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

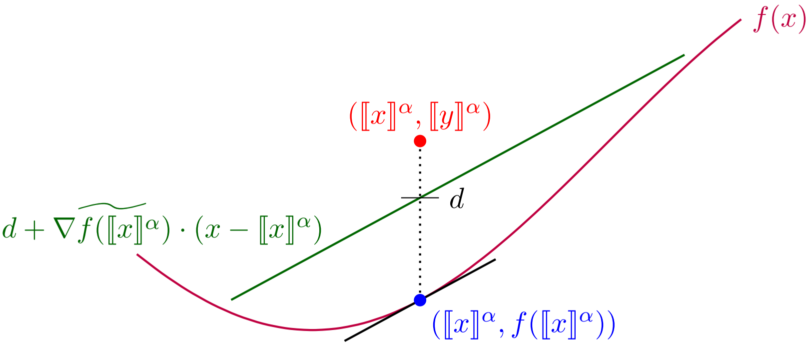
1. decide there is a conflict



Tangent space linearisation (schematic)

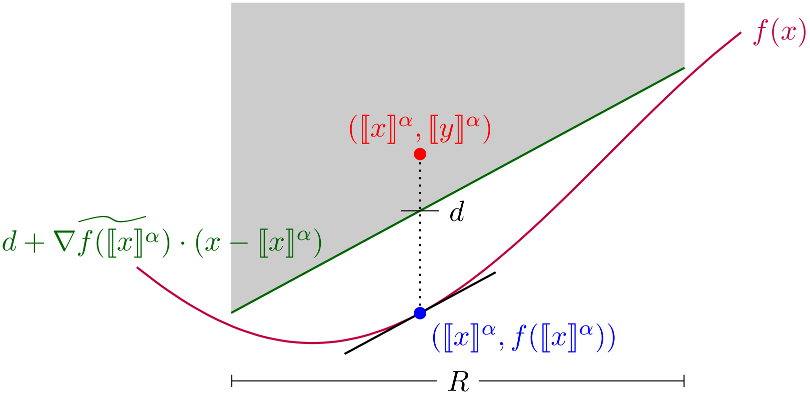
$\underbrace{f(x) \geq y}_{P}$, $\alpha : V \to \mathbb{Q}$

1. decide there is a conflict
2. compute linearization



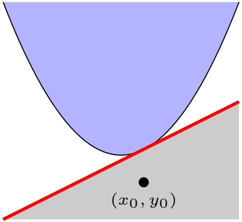
Tangent space linearisation (schematic)

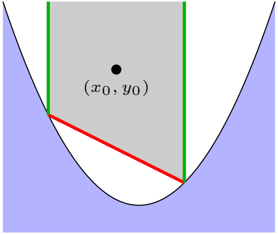
$\underbrace{f(x) \geq y}_{P}$, $\alpha : V \to \mathbb{Q}$ 1. decide there is a conflict 2. compute linearization



Special classes: convex/concave

- f convex:

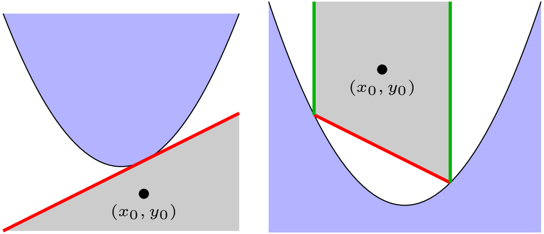




abs or $x \mapsto x^{2n}$ for $n \in \mathbb{N}$

Special classes: convex/concave

- f convex:

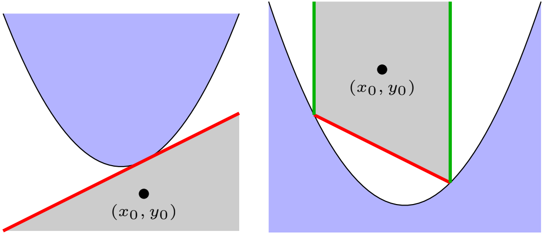


abs or $x \mapsto x^{2n}$ for $n \in \mathbb{N}$

- f concave $\iff$ $-f$ convex

Special classes: convex/concave

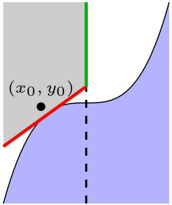
- f convex:



abs or $x \mapsto x^{2n}$ for $n \in \mathbb{N}$

- f concave $\iff$ $-f$ convex

- f piecewise convex/-cave:



e.g. $x \mapsto x^{2n+1}$ for $n \in \mathbb{N}$

δ–`ksmt` termination

Sufficient termination conditions

Let $\epsilon > 0$.

- A linearisation C at an assignment α is ϵ-full if it excludes all assignments in an open ϵ-ball around α.

Sufficient termination conditions

Let $\epsilon > 0$.

- A linearisation C at an assignment α is ϵ-full if it excludes all assignments in an open ϵ-ball around α.
- A `ksmt` run is ϵ-full, if all but finitely many linearisations in this run are ϵ-full.

Sufficient termination conditions

Let $\epsilon > 0$.

- A linearisation C at an assignment α is ϵ-full if it excludes all assignments in an open ϵ-ball around α.
- A ksmt run is ϵ-full, if all but finitely many linearisations in this run are ϵ-full.

A formula F is a bounded instance if

- $F = \mathcal{L}_0 \wedge \mathcal{N}$ is in separated linear form, and
- ranges of all variables are bounded by linear predicates from $\mathcal{L}_0$, and
- closure of ranges is $\subseteq \operatorname{dom} f$ for all f in $\mathcal{N}$.

Sufficient termination conditions

Let $\epsilon > 0$.

- A linearisation C at an assignment α is ϵ-full if it excludes all assignments in an open ϵ-ball around α.
- A ksmt run is ϵ-full, if all but finitely many linearisations in this run are ϵ-full.

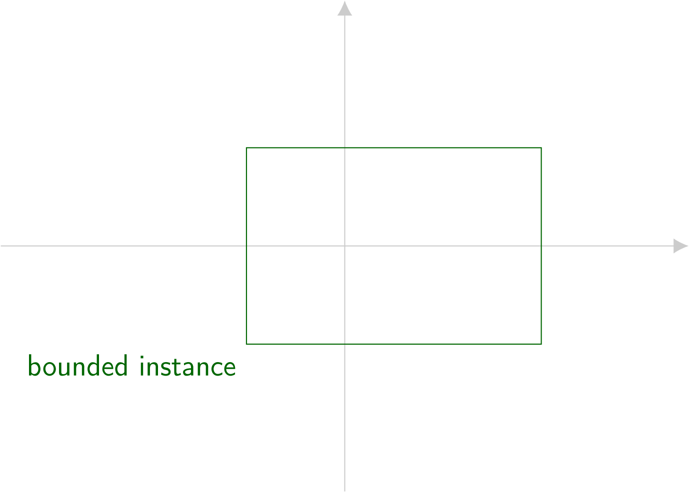
A formula F is a bounded instance if

- $F = \mathcal{L}_0 \wedge \mathcal{N}$ is in separated linear form, and
- ranges of all variables are bounded by linear predicates from $\mathcal{L}_0$, and
- closure of ranges is $\subseteq \operatorname{dom} f$ for all f in $\mathcal{N}$.

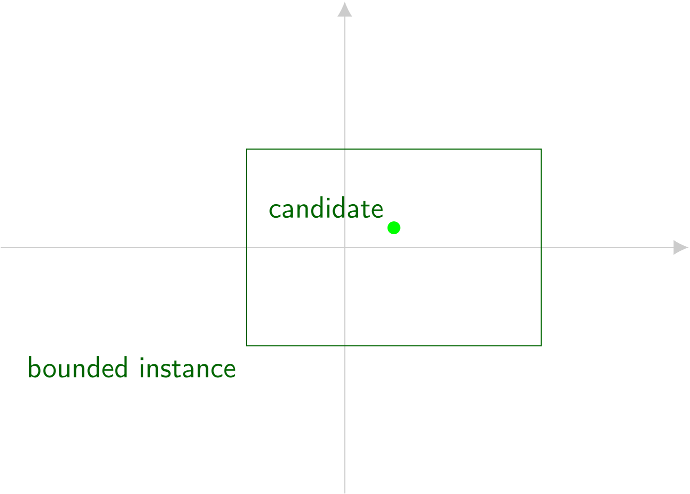
Theorem

On bounded instances, ϵ-full ksmt runs are terminating.

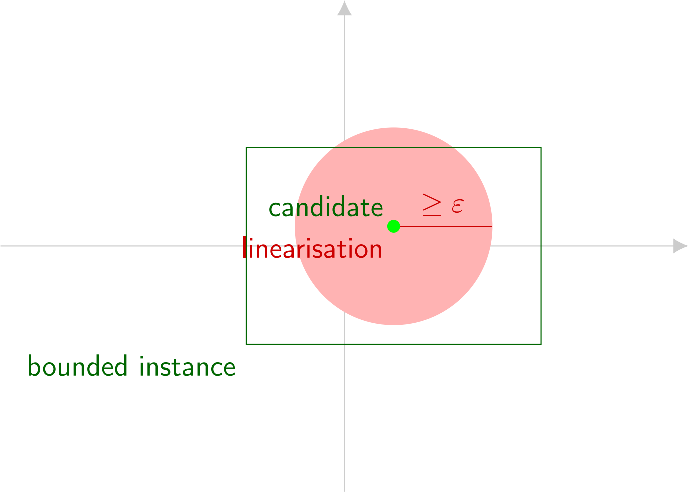
Terminating runs by Example



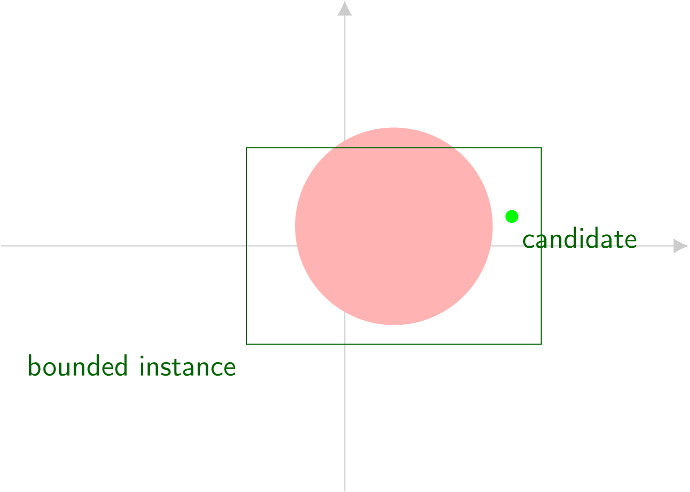
Terminating runs by Example



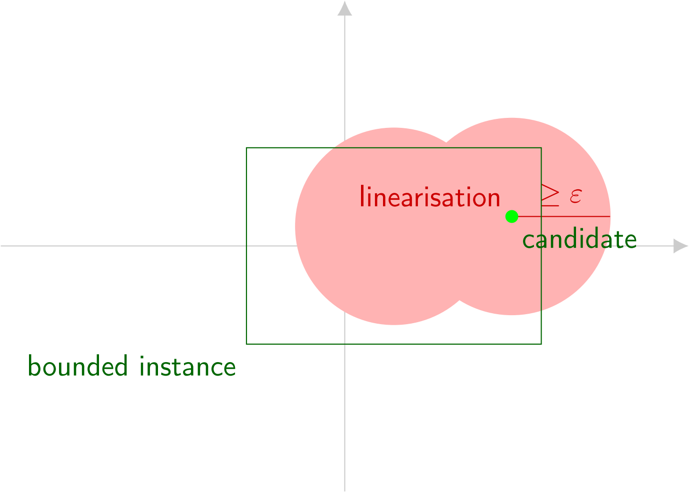
Terminating runs by Example



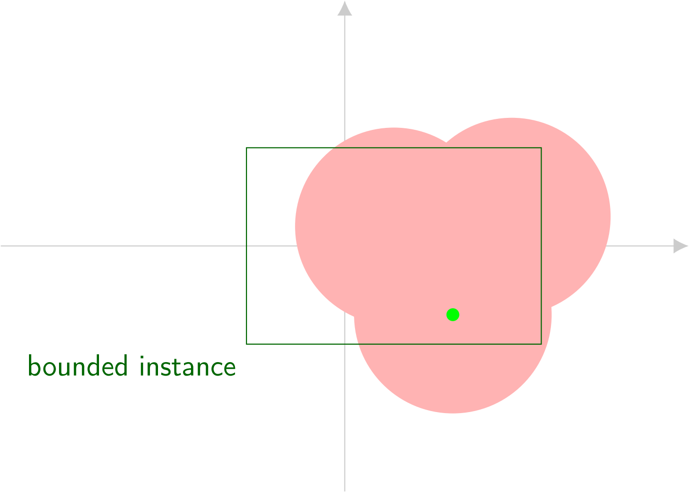
Terminating runs by Example



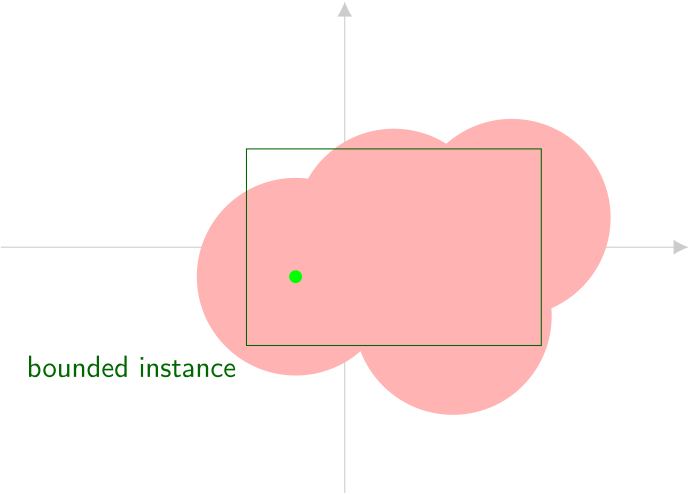
Terminating runs by Example



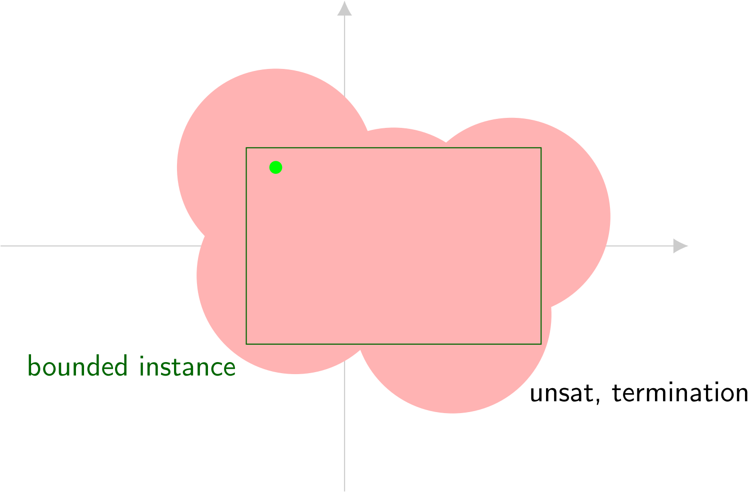
Terminating runs by Example



Terminating runs by Example



Terminating runs by Example

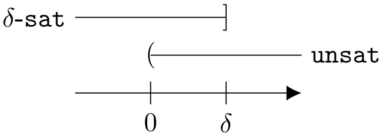


`ksmt` is a δ-complete decision procedure for non-linear constraints

δ-decidability

Let $\delta > 0$ be rational. The δ-relaxation P_δ of a constraint $P : f(\boldsymbol{x}) \diamond 0$ is

- $P_\delta : |f(\boldsymbol{x})| \leq \delta$ when $\diamond \in \{=\}$,
- $P_\delta : f(\boldsymbol{x}) \diamond \delta$ when $\diamond \in \{<, \leq\}$, and
- $P_\delta : f(\boldsymbol{x}) \diamond -\delta$ when $\diamond \in \{>, \geq\}$.



$P : f(\boldsymbol{x}) \leq 0.$

Definition [S. Gao, J. Avigad, E. Clarke, '12]

δ-deciding a formula F denotes computing

- δ-`sat`, if there is α s.t. $[\![F_\delta]\!]^\alpha =$ true.
- `unsat`, if F is unsatisfiable.

In case both answers are valid, either output is acceptable.

For `ksmt`, just relaxing the non-linear part for δ-`sat` suffices: $\mathcal{L}_0 \wedge \mathcal{N}_\delta$.

δ-ksmt calculus

- Transition rules define relation $\Rightarrow$ on states $(\alpha, \mathcal{L}, \mathcal{N})$.
  - (partial) assignment α
  - linear inequalities $\mathcal{L}$
  - non-linear units $\mathcal{N}$
- initial state is $(\text{nil}, \mathcal{L}_0, \mathcal{N}_0)$ for formula in separated linear form $\mathcal{L}_0 \wedge \mathcal{N}_0$.
- sat, $unsat$ and δ-sat are final states.

δ-ksmt calculus

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

δ-ksmt calculus

Rules	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	
(A) assign	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	z unassigned, $q \in \mathbb{Q}$ and no linear conflict

δ-ksmt calculus

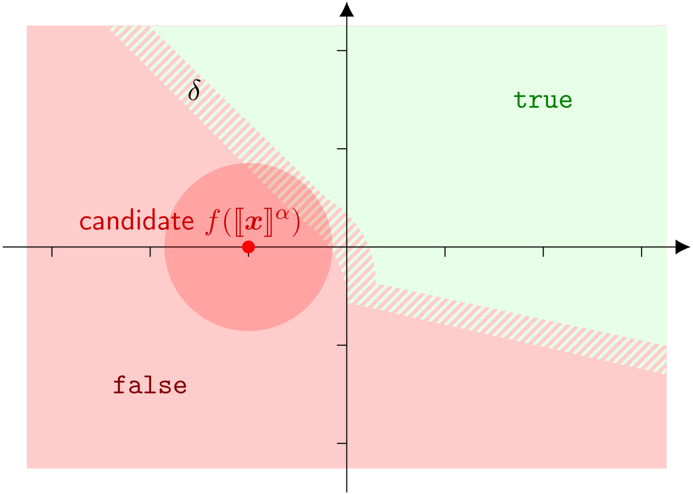
Rules	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	
(A) assign	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	z unassigned, $q \in \mathbb{Q}$ and no linear conflict
(R) resolve	$(\alpha, \mathcal{L} \cup R, \mathcal{N})$	R resolvent excluding the linear conflict

δ-ksmt calculus

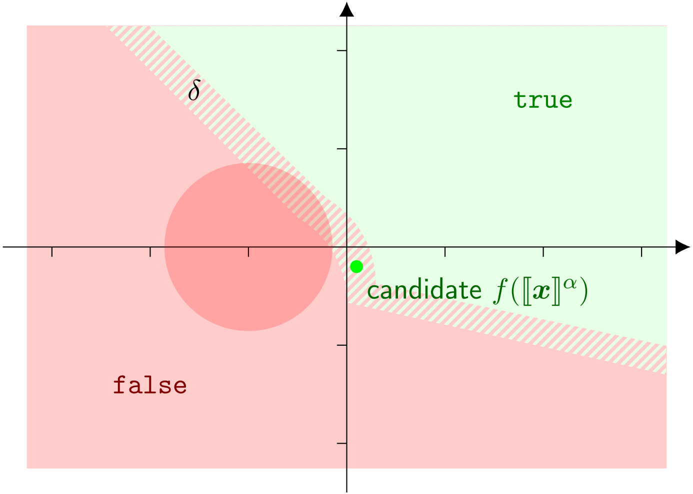
Rules	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	
(A) assign	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	z unassigned, $q \in \mathbb{Q}$ and no linear conflict
(R) resolve	$(\alpha, \mathcal{L} \cup R, \mathcal{N})$	R resolvent excluding the linear conflict
(B) backjump	$(\gamma, \mathcal{L}, \mathcal{N})$	to γ maximal conflict-free prefix of α

δ-ksmt calculus

Rules	$(\alpha, \mathcal{L}, \mathcal{N}) \Rightarrow \circ$	
(A) assign	$(\alpha :: z \mapsto q, \mathcal{L}, \mathcal{N})$	z unassigned, $q \in \mathbb{Q}$ and no linear conflict
(R) resolve	$(\alpha, \mathcal{L} \cup R, \mathcal{N})$	R resolvent excluding the linear conflict
(B) backjump	$(\gamma, \mathcal{L}, \mathcal{N})$	to γ maximal conflict-free prefix of α
(L) linearise	$(\alpha, \mathcal{L} \cup L, \mathcal{N})$	L linearisation, excluding the non-linear conflict
(F^{sat})	sat	all variables are assigned, no linear conflict and (A), (R), (B), (L) are not applicable
(F^{unsat})	$unsat$	$[\![\mathcal{L}]\!]^{nil} =$ false
(F^{sat}_{δ})	δ-sat	all variables are assigned and $[\![\mathcal{L} \wedge \mathcal{N}_{\delta}]\!]^{\alpha} =$ true



δ
true
candidate $f([\![x]\!]^\alpha)$
false



δ
true
candidate $f(\llbracket \boldsymbol{x} \rrbracket^{\alpha})$
false

Theorem

Soundness of `ksmt` carries over to δ-`ksmt`.

Computable functions on $\mathbb{R}$ instead of $\mathcal{F}_{\mathsf{DA}}$ provide a computable modulus of continuity.

We provide algorithms computing ϵ-full linearisations via:

- Linearise_δ (uniform) modulus of continuity, and
- $\text{LineariseLocal}_\delta$ local modulus of continuity extracted from computability of f.

Theorem

On bounded instances, there is $\epsilon > 0$ such that δ-`ksmt` runs with linearisations computed by Linearise_δ and $\text{LineariseLocal}_\delta$ are ϵ-full.

Theorem

δ-`ksmt` is a δ-complete decision procedure.

ksmt implementation/evaluation

ksmt implementation

ksmt system:

- SMT solver for non-linear arithmetic
- Model guided architecture in the spirit of conflict resolution/MCSAT
- Including SAT/linear/non-linear in one incremental framework
- Integrates iRRAM – system for exact real arithmetic based on computable analysis developed by Norbert Th. Müller and colleagues.
- Open source: `http://informatik.uni-trier.de/~brausse/ksmt`

Conclusions and future work

`ksmt` calculus:

- model-guided search & resolution of non-linear conflicts via local linearisation
- prototypical implementation with promising results
- identified broad class of functions $\mathcal{F}_{\mathrm{DA}}$ for which conflicts are decidable
- δ-complete decision procedure for bounded instances

Future:

- more precise linearisations for specific functions
- analyze complexity of deciding conflicts