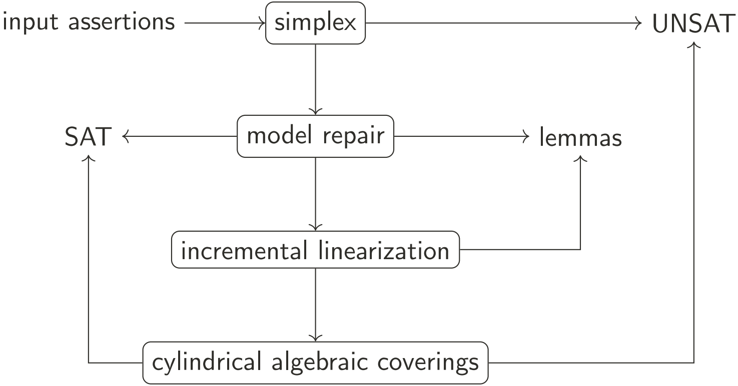
Solving nonlinear real arithmetic in cvc5

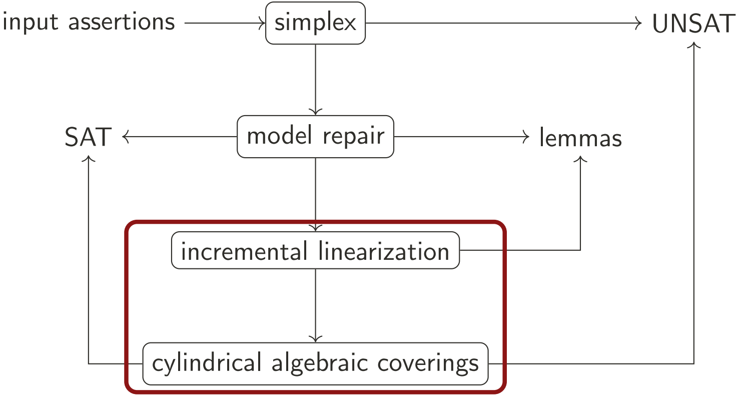
Gereon Kremer

May 9, 2022

Arithmetic solving in cvc5



Arithmetic solving in cvc5



Incremental linearization

implicitly linearize: $x \cdot y \rightsquigarrow a_{x \cdot y}$

$$x > 2 \land y > -1 \land x \cdot y < 2$$

[Cimatti et al. 2018]

Incremental linearization

implicitly linearize: $x \cdot y \rightsquigarrow a_{x \cdot y}$

$$\begin{aligned} & x > 2 \wedge y > -1 \wedge x \cdot y < 2 \\ \text{Model:}\ & x \mapsto 3, y \mapsto 0, x \cdot y \mapsto 1 \\ \text{Lemma:}\ & y = 0 \Rightarrow x \cdot y = 0 \end{aligned}$$

[Cimatti et al. 2018]

Incremental linearization

implicitly linearize: $x \cdot y \rightsquigarrow a_{x \cdot y}$

$$
\begin{aligned}
& x > 2 \land y > -1 \land x \cdot y < 2 \\
\text{Model: } & x \mapsto 3, y \mapsto 0, x \cdot y \mapsto 1 \\
\text{Lemma: } & y = 0 \Rightarrow x \cdot y = 0 \\
\text{Model: } & x \mapsto 3, y \mapsto 1, x \cdot y \mapsto 0 \\
\text{Lemma: } & (x = 3 \land y = 1) \Rightarrow x \cdot y = 3
\end{aligned}
$$

[Cimatti et al. 2018]

Incremental linearization

implicitly linearize: $x \cdot y \rightsquigarrow a_{x \cdot y}$

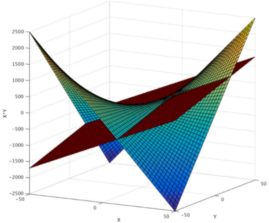
$$
\begin{aligned}
& x > 2 \land y > -1 \land x \cdot y < 2 \\
\text{Model: } & x \mapsto 3, y \mapsto 0, x \cdot y \mapsto 1 \\
\text{Lemma: } & y = 0 \Rightarrow x \cdot y = 0 \\
\text{Model: } & x \mapsto 3, y \mapsto 1, x \cdot y \mapsto 0 \\
\text{Lemma: } & \sout{(x = 3 \land y = 1) \Rightarrow x \cdot y = 3}
\end{aligned}
$$

[Cimatti et al. 2018]

Incremental linearization

implicitly linearize: $x \cdot y \rightsquigarrow a_{x \cdot y}$

$$
\begin{aligned}
& x > 2 \wedge y > -1 \wedge x \cdot y < 2 \\
\text{Model: } & x \mapsto 3, y \mapsto 0, x \cdot y \mapsto 1 \\
\text{Lemma: } & y = 0 \Rightarrow x \cdot y = 0 \\
\text{Model: } & x \mapsto 3, y \mapsto 1, x \cdot y \mapsto 0 \\
\text{Lemma: } & \cancel{(x = 3 \wedge y = 1) \Rightarrow x \cdot y = 3}
\end{aligned}
$$



[Cimatti et al. 2018]

[Cimatti et al. 2018]

Incremental linearization

implicitly linearize: $x \cdot y \rightsquigarrow a_{x \cdot y}$

$$x > 2 \land y > -1 \land x \cdot y < 2$$

Model: $x \mapsto 3, y \mapsto 0, x \cdot y \mapsto 1$

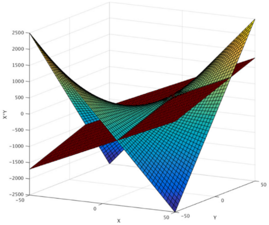
Lemma: $y = 0 \Rightarrow x \cdot y = 0$

Model: $x \mapsto 3, y \mapsto 1, x \cdot y \mapsto 0$

Lemma: ~~$(x = 3 \land y = 1) \Rightarrow x \cdot y = 3$~~

$$(x \leq 3 \land y \leq 1) \lor (x \geq 3 \land y \geq 1)$$

$$\Leftrightarrow (x \cdot y \geq 1 \cdot x + 3 \cdot y - 3 \cdot 1)$$



[Cimatti et al. 2018]

[Cimatti et al. 2018]

Incremental linearization – schemas

split zero	$\top \Rightarrow (t = 0 \lor t \neq 0)$
sign	$x > 0 \land y > 0 \Rightarrow xy > 0$
	$x = 0 \Rightarrow xyz = 0$
magnitude	$\|x\| > \|y\| \Rightarrow \|xz\| > \|yz\|$
	$\|z\| > \|y\| \land \|u\| > \|w\| \land \|x\| \geq 1 \Rightarrow \|zuxx\| > \|yw\|$
bounds	$x > 0 \land y > z + w \Rightarrow xy > x(z + w)$
resolution bounds	$y \geq 0 \land s \leq xz \land xy \leq t \Rightarrow ys \leq zt$
tangent plane	$(x \leq 3 \land y \leq 1) \lor (x \geq 3 \land y \geq 1) \Rightarrow xy \geq x + 3y - 3$

Cylindrical Algebraic Coverings

- Guess partial assignment

$$s_1 \times \cdots \times s_k \times s_{k+1}$$

[Ábrahám et al. 2021] [Kremer et al. 2021]

Cylindrical Algebraic Coverings

- Guess partial assignment

$$s_1 \times \cdots \times s_k \times s_{k+1}$$

- Refute partial assignment using intervals

$$s \notin s_1 \times \cdots \times s_k \times (a, b)$$

[Ábrahám et al. 2021] [Kremer et al. 2021]

Cylindrical Algebraic Coverings

- Guess partial assignment

$$s_1 \times \cdots \times s_k \times s_{k+1}$$

- Refute partial assignment using intervals

$$s \notin s_1 \times \cdots \times s_k \times (a, b)$$

- Project covering to lower dimension

$$s_1 \times \cdots \times s_k \times \{(-\infty, a), [a, b], \ldots (z, \infty)\} \rightarrow s_1 \times \cdots \times s_{k-1} \times (\alpha, \beta)$$

[Ábrahám et al. 2021] [Kremer et al. 2021]

Cylindrical Algebraic Coverings

- Guess partial assignment

$$s_1 \times \cdots \times s_k \times s_{k+1}$$

- Refute partial assignment using intervals

$$s \notin s_1 \times \cdots \times s_k \times (a, b)$$

- Project covering to lower dimension

$$s_1 \times \cdots \times s_k \times \{(-\infty, a), [a, b], \ldots (z, \infty)\} \to s_1 \times \cdots \times s_{k-1} \times (\alpha, \beta)$$

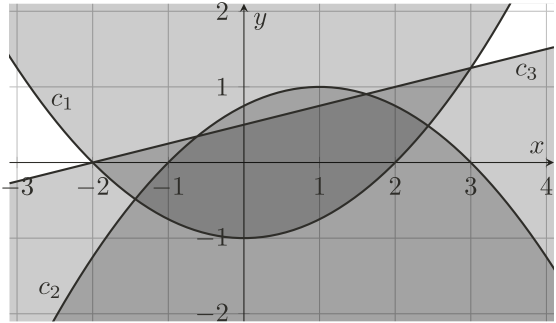
- Eventually get satisfying assignment or a covering in first dimension

$$s = s_1 \times \cdots \times s_n \qquad \text{or} \qquad s_1 \notin \{(-\infty, a), [a, b], \ldots (z, \infty)\}$$

[Ábrahám et al. 2021] [Kremer et al. 2021]

An example

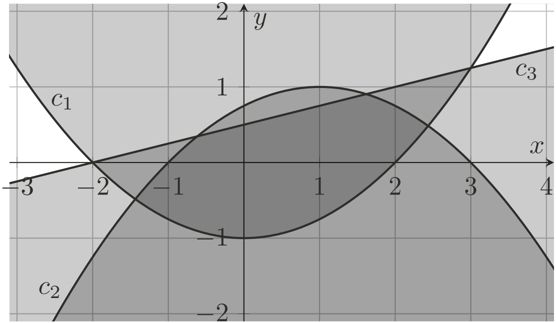
$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$



An example

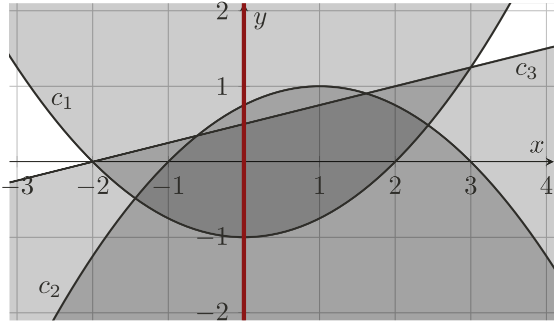
$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$

No constraint for x



An example

$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$

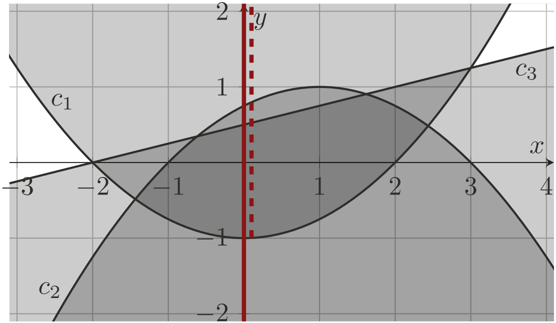


No constraint for x

Guess $x \mapsto 0$

An example

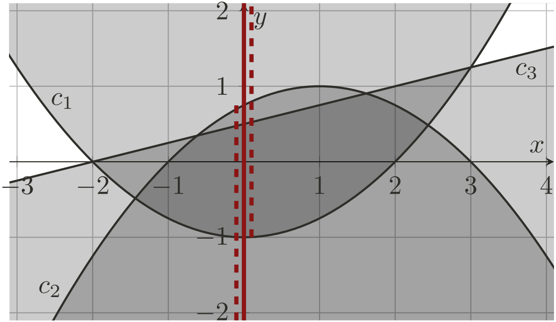
$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$



No constraint for x
Guess $x \mapsto 0$
$c_1 \to y \notin (-1, \infty)$

An example

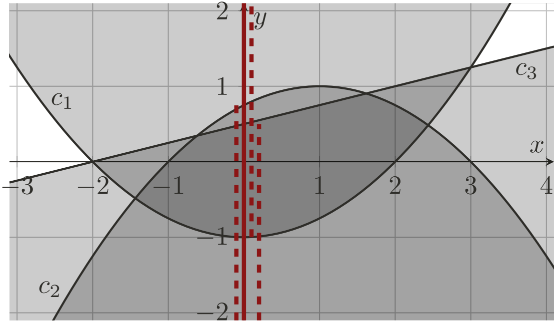
$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$



No constraint for x
Guess $x \mapsto 0$
$c_1 \to y \notin (-1, \infty)$
$c_2 \to y \notin (-\infty, 0.75)$

An example

$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$



No constraint for x

Guess $x \mapsto 0$

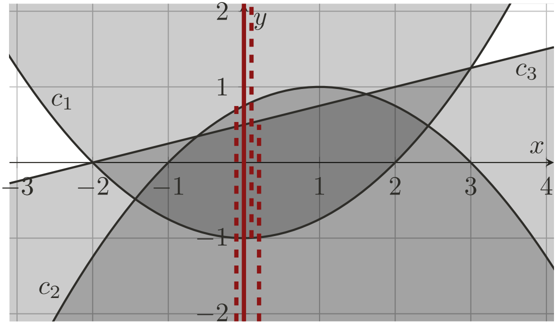
$c_1 \to y \notin (-1, \infty)$

$c_2 \to y \notin (-\infty, 0.75)$

$c_3 \to y \notin (-\infty, 0.5)$

An example

$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$



No constraint for x

Guess $x \mapsto 0$

$c_1 \to y \notin (-1, \infty)$

$c_2 \to y \notin (-\infty, 0.75)$

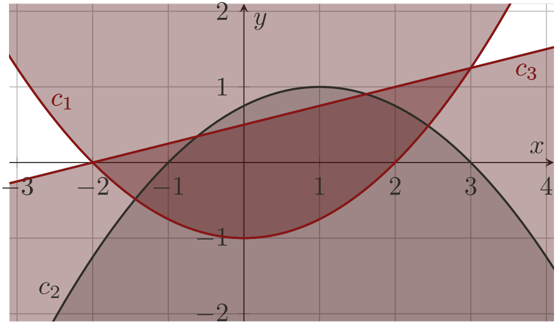
$c_3 \to y \notin (-\infty, 0.5)$

Construct covering

$(-\infty, 0.5), (-1, \infty)$

An example

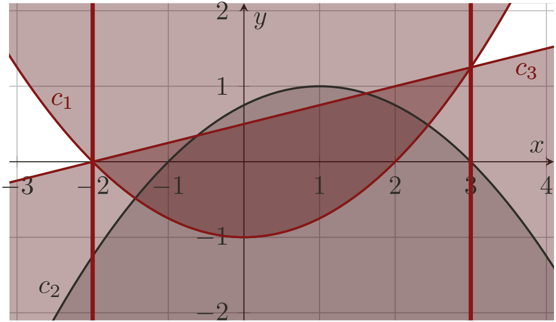
$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$



No constraint for x
Guess $x \mapsto 0$
$c_1 \to y \notin (-1, \infty)$
$c_2 \to y \notin (-\infty, 0.75)$
$c_3 \to y \notin (-\infty, 0.5)$
Construct covering
$(-\infty, 0.5), (-1, \infty)$

An example

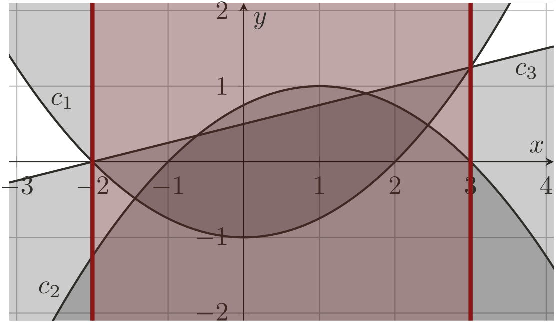
$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$



No constraint for x
Guess $x \mapsto 0$
$c_1 \to y \notin (-1, \infty)$
$c_2 \to y \notin (-\infty, 0.75)$
$c_3 \to y \notin (-\infty, 0.5)$
Construct covering
$(-\infty, 0.5), (-1, \infty)$
Construct interval for x
$x \notin (-2, 3)$

An example

$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 4 \cdot y > 4 - (x-1)^2 \qquad c_3 : 4 \cdot y > x + 2$$



No constraint for x
Guess $x \mapsto 0$
$c_1 \to y \notin (-1, \infty)$
$c_2 \to y \notin (-\infty, 0.75)$
$c_3 \to y \notin (-\infty, 0.5)$
Construct covering
$(-\infty, 0.5), (-1, \infty)$
Construct interval for x
$x \notin (-2, 3)$
New guess for x

Cylindrical Algebraic Coverings – main algorithm

```
function get_unsat_cover((s_1, ..., s_{i-1}))

𝕀 := get_unsat_intervals(s)
while ⋃_{I∈𝕀} I ≠ ℝ do
  s_i := sample_outside(𝕀)
  if i = n then return (SAT, (s_1, ..., s_{i-1}, s_i))
  (f, O) := get_unsat_cover((s_1, ..., s_{i-1}, s_i))
  if f = SAT then return (SAT, O)
  else if f = UNSAT then
    R := construct_characterization((s_1, ..., s_{i-1}, s_i), O)
    J := interval_from_characterization((s_1, ..., s_{i-1}), s_i, R)
    𝕀 := 𝕀 ∪ {J}
return (UNSAT, 𝕀)
```

[Ábrahám et al. 2021] [Kremer et al. 2021]

Cylindrical Algebraic Coverings – main algorithm

```
function get_unsat_cover((s_1, ..., s_{i-1}))
```

Real root isolation over a partial sample point

$\mathbb{I}$:= `get_unsat_intervals(s)`
while $\bigcup_{I \in \mathbb{I}} I \neq \mathbb{R}$ **do**
  s_i := `sample_outside`($\mathbb{I}$)
  if $i = n$ **then return** $(\mathrm{SAT}, (s_1, \ldots, s_{i-1}, s_i))$
  (f, O) := `get_unsat_cover`($(s_1, \ldots, s_{i-1}, s_i)$)
  if $f = \mathrm{SAT}$ **then return** (SAT, O)
  else if $f = \mathrm{UNSAT}$ **then**
    R := `construct_characterization`($(s_1, \ldots, s_{i-1}, s_i), O$)
    J := `interval_from_characterization`($(s_1, \ldots, s_{i-1}), s_i, R$)
    $\mathbb{I} := \mathbb{I} \cup \{J\}$
return $(\mathrm{UNSAT}, \mathbb{I})$

[Ábrahám et al. 2021] [Kremer et al. 2021]

Cylindrical Algebraic Coverings – main algorithm

```
function get_unsat_cover((s_1, ..., s_{i-1}))

𝕀 := get_unsat_intervals(s)
while ⋃_{I∈𝕀} I ≠ ℝ do
  s_i := sample_outside(𝕀)
  if i = n then return (SAT, (s_1, ..., s_{i-1}, s_i))
  (f, O) := get_unsat_cover((s_1, ..., s_{i-1}, s_i))
  if f = SAT then return (SAT, O)
  else if f = UNSAT then
    R := construct_characterization((s_1, ..., s_{i-1}, s_i), O)
    J := interval_from_characterization((s_1, ..., s_{i-1}), s_i, R)
    𝕀 := 𝕀 ∪ {J}
return (UNSAT, 𝕀)
```

Real root isolation over a partial sample point

Select sample from $\mathbb{R} \setminus I$

[Ábrahám et al. 2021] [Kremer et al. 2021]

Cylindrical Algebraic Coverings – main algorithm

```
function get_unsat_cover((s_1, ..., s_{i-1}))

𝕀 := get_unsat_intervals(s)
while ⋃_{I∈𝕀} I ≠ ℝ do
  | s_i := sample_outside(𝕀)
  | if i = n then return (SAT, (s_1, ..., s_{i-1}, s_i))
  | (f, O) := get_unsat_cover((s_1, ..., s_{i-1}, s_i))
  | if f = SAT then return (SAT, O)
  | else if f = UNSAT then
  |   | R := construct_characterization((s_1, ..., s_{i-1}, s_i), O)
  |   | J := interval_from_characterization((s_1, ..., s_{i-1}), s_i, R)
  |   | 𝕀 := 𝕀 ∪ {J}
return (UNSAT, 𝕀)
```

Real root isolation over a partial sample point

Select sample from $\mathbb{R} \setminus I$

Recurse to next variable

[Ábrahám et al. 2021] [Kremer et al. 2021]

Cylindrical Algebraic Coverings – main algorithm

```
function get_unsat_cover((s_1, ..., s_{i-1}))

𝕀 := get_unsat_intervals(s)
while ⋃_{I∈𝕀} I ≠ ℝ do
  s_i := sample_outside(𝕀)
  if i = n then return (SAT, (s_1, ..., s_{i-1}, s_i))
  (f, O) := get_unsat_cover((s_1, ..., s_{i-1}, s_i))
  if f = SAT then return (SAT, O)
  else if f = UNSAT then
    R := construct_characterization((s_1, ..., s
    J := interval_from_characterization((s_1, 
    𝕀 := 𝕀 ∪ {J}
return (UNSAT, 𝕀)
```

Real root isolation over a partial sample point

Select sample from $\mathbb{R} \setminus I$

Recurse to next variable

CAD-style projection: Roots of polynomials restrict where covering is still applicable

[Ábrahám et al. 2021] [Kremer et al. 2021]

function get_unsat_cover($(s_1, \ldots, s_{i-1})$)

$\mathbb{I}$:= get_unsat_intervals(s)
while $\bigcup_{I \in \mathbb{I}} I \neq \mathbb{R}$ **do**
  s_i := sample_outside($\mathbb{I}$)
  if $i = n$ **then** **return** $(\mathrm{SAT}, (s_1, \ldots, s_{i-1}, s_i))$
  (f, O) := get_unsat_cover($(s_1, \ldots, s_{i-1}, s_i)$)
  if $f = \mathrm{SAT}$ **then** **return** (SAT, O)
  else if $f = \mathrm{UNSAT}$ **then**
    R := construct_characterization($(s_1, \ldots, s$
    J := interval_from_characterization($(s_1,$
    $\mathbb{I} := \mathbb{I} \cup \{J\}$
return $(\mathrm{UNSAT}, \mathbb{I})$

Real root isolation over a partial sample point

Select sample from $\mathbb{R} \setminus I$

Recurse to next variable

CAD-style projection: Roots of polynomials restrict where covering is still applicable

Extract interval from polynomials

[Ábrahám et al. 2021] [Kremer et al. 2021]

Some implementation details

- Heavily based on LibPoly [Jovanovic et al. 2017]
- Implements stuff beyond [Ábrahám et al. 2021]:
  - Different projection operators (McCallum, Lazard)
  - Lazard's lifting [Lazard 1994] [Kremer et al. 2021] using CoCoALib [Abbott et al. 2018]
  - Different variable orderings inspired by [England et al. 2014]
  - Generates infeasible subsets
    Store assertions with every interval
  - Supports mixed-integer problems using naive B&B-style intervals
  - Generation of formal proof skeletons
    Helps understanding, not detailed enough for automated verification
  - Experimental support for incremental checks
    No performance benefit observed, lives in a branch
- Arbitrary theory combination
  Real algebraic numbers are first-class citizens of `cvc5`

Experiments

	QF_NRA	sat	unsat	solved
→	cvc5	**5137**	**5596**	**10733**
	Yices2	4966	5450	10416
	z3	5136	5207	10343
→	cvc5.cov	5001	5077	10078
	SMT-RAT	4828	5038	9866
	veriT	4522	5034	9556
	MathSAT	3645	5357	9002
→	cvc5.inclin	3421	5376	8797

Incremental linearization – extensions

Also supports extended operators in the same style:

- transcendentals ($\pi, \sin, \cos, \tan, \ldots$)
- exponentials ($\exp$)
- bitwise and on integers (`IAND`, `bvand` in arithmetic)
- power of two (`POW2`, bit shift in arithmetic)

Easily integrates other solving techniques

- does one or more of the following:
  - generate a (preferably) linear lemma that rejects the current model
  - finds a proper model
- implemented: ICP-style propagations
- ideas: GB-style conflicts, subtropical satisfiability, ...

Conclusion

- combines linearization and coverings
- conceptually simple strategy
- easily integrates other techniques
- there is more to do...

Any questions?

References I

- John Abbott, Anna M. Bigatti, and Elisa Palezzato. "New in CoCoA-5.2.4 and CoCoALib-0.99600 for SC-Square". In: SC^2. FLoC. July 2018. URL: http://ceur-ws.org/Vol-2189/paper4.pdf.
- Erika Ábrahám, James H. Davenport, Matthew England, and Gereon Kremer. "Deciding the Consistency of Non-Linear Real Arithmetic Constraints with a Conflict Driven Search Using Cylindrical Algebraic Coverings". In: Journal of Logical and Algebraic Methods in Programming 119 (2021), p. 100633. DOI: 10.1016/j.jlamp.2020.100633.
- Alessandro Cimatti, Alberto Griggio, Ahmed Irfan, Marco Roveri, and Roberto Sebastiani. "Incremental Linearization for Satisfiability and Verification Modulo Nonlinear Arithmetic and Transcendental Functions". In: ACM Transactions on Computational Logic 19 (3 2018), 19:1–19:52. DOI: 10.1145/3230639.
- Matthew England, Russell Bradford, James H. Davenport, and David Wilson. "Choosing a Variable Ordering for Truth-Table Invariant Cylindrical Algebraic Decomposition by Incremental Triangular Decomposition". In: ICMS. 2014. DOI: 10.1007/978-3-662-44199-2_68.
- Dejan Jovanovic and Bruno Dutertre. "LibPoly: A Library for Reasoning about Polynomials". In: SMT. CAV. 2017. URL: http://ceur-ws.org/Vol-1889/paper3.pdf.
- Gereon Kremer, Erika Ábrahám, Matthew England, and James H. Davenport. "On the Implementation of Cylindrical Algebraic Coverings for Satisfiability Modulo Theories Solving". In: SYNASC. 2021. DOI: 10.1109/SYNASC54541.2021.00018.
- Gereon Kremer and Jens Brandt. "Implementing arithmetic over algebraic numbers A tutorial for Lazard's lifting scheme in CAD". In: SYNASC. 2021. DOI: 10.1109/SYNASC54541.2021.00013.
- Daniel Lazard. "An Improved Projection for Cylindrical Algebraic Decomposition". In: Algebraic Geometry and its Applications. 1994. Chap. 29, pp. 467–476. DOI: 10.1007/978-1-4612-2628-4_29.