Formal proofs for Cylindrical Algebraic Coverings

... and other nonlinear reasoning techniques

Gereon Kremer

Some context

Assume

- you have an SMT solver (like `cvc5`)
- you support nonlinear arithmetic reasoning
- you produce formal proofs in the core solver ($\rightarrow$ Haniel's talk)
- you want to have proofs for theory reasoning

Overview

Proofs for incremental linearization

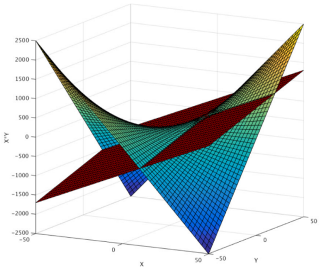
- Obtain a (linear) model
- Select a lemma schema
- Instantiate appropriately
- Refute the current (linear) model

[Cimatti et al. 2018]

Proofs for incremental linearization

- Obtain a (linear) model
- Select a lemma schema
- Instantiate appropriately
- Refute the current (linear) model

$$\begin{aligned}&(x \cdot y \geq b \cdot x + a \cdot y - a \cdot b)\\ \Leftrightarrow &(x \leq a \land y \leq b) \lor (x \geq a \land y \geq b)\end{aligned}$$



Source: [Cimatti et al. 2018]

[Cimatti et al. 2018]

Proofs for incremental linearization

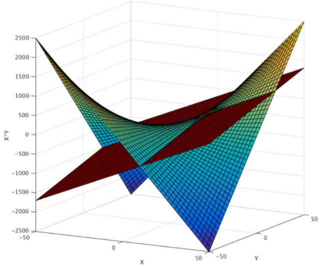
- Obtain a (linear) model
- Select a lemma schema
- Instantiate appropriately
- Refute the current (linear) model

- Most of them are even simpler:

$$(x = 0) \vee \neg(x = 0)$$

$$(x \cdot z + y \cdot z > 0)$$
$$\Rightarrow (k = x + y \wedge k \cdot z > 0)$$

$$(x \cdot y \geq b \cdot x + a \cdot y - a \cdot b)$$
$$\Leftrightarrow (x \leq a \wedge y \leq b) \vee (x \geq a \wedge y \geq b)$$



Source: [Cimatti et al. 2018]

[Cimatti et al. 2018]

Proofs for some other arithmetic methods

- Simplex
  - Input constraints $\bigwedge_j \sum_i c_{ij} \cdot x_i \bowtie_j c_{0j}$, $\bowtie_j \in \{>, \geq\}$
  - Farkas lemmas provides coefficients for $\sum_j s_j \sum_i c_{ij} \cdot x_i = 0$
  - But either $\sum_j c_{0j} > 0$ or some $\bowtie_j$ is strict
- Interval Constraint Propagation
  - Propagation: $(0 \leq x \leq 2) \wedge (y = x^2) \Rightarrow (0 \leq z \leq 4)$
  - Split: $(x < 7) \vee (x = 7) \vee (x > 7)$
- Incremental linearization for transcendental functions [Cimatti et al. 2018]
  - $exp(x) > 0$, $(x > 0) \Rightarrow (exp(x) > t + 1)$, $sin(x) = sin(-x)$, ...
  - Tangent lemmas based on Taylor approximations

Proofs for some other arithmetic methods

- Simplex
  - Input constraints $\bigwedge_j \sum_i c_{ij} \cdot x_i \bowtie_j c_{0j}$, $\bowtie_j \in \{>, \geq\}$
  - Farkas lemmas provides coefficients for $\sum_j s_j \sum_i c_{ij} \cdot x_i = 0$
  - But either $\sum_j c_{0j} > 0$ or some $\bowtie_j$ is strict
- Interval Constraint Propagation
  - Propagation: $(0 \leq x \leq 2) \wedge (y = x^2) \Rightarrow (0 \leq z \leq 4)$
  - Split: $(x < 7) \vee (x = 7) \vee (x > 7)$
- Incremental linearization for transcendental functions [Cimatti et al. 2018]
  - $exp(x) > 0$, $(x > 0) \Rightarrow (exp(x) > t + 1)$, $sin(x) = sin(-x)$, ...
  - Tangent lemmas based on Taylor approximations

For all of them:
Lemmas may be difficult to find, but are easy to prove.

Cylindrical Algebraic Decomposition

A brief digression...

Cylindrical Algebraic Decomposition

A very very very very abstract view:

$$\begin{array}{c} \text{Set of constraints} \\ \downarrow \\ \text{Set of polynomials} \\ \downarrow \\ \text{Set of solution candidates} \\ \downarrow \\ \text{SAT or UNSAT} \end{array}$$

Cylindrical Algebraic Decomposition

A very very very very abstract view:

Set of constraints

↓ Characterize truth-invariant cells

Set of polynomials

↓

Set of solution candidates

↓

SAT or UNSAT

Cylindrical Algebraic Decomposition

A very very very very abstract view:

Set of constraints

↓ Characterize truth-invariant cells

Set of polynomials

↓ Select one point per cell

Set of solution candidates

↓

SAT or UNSAT

Cylindrical Algebraic Decomposition

A very very very very abstract view:

$$\begin{array}{c} \text{Set of constraints} \\ \big\downarrow \quad \text{Characterize truth-invariant cells} \\ \text{Set of polynomials} \\ \big\downarrow \quad \text{Select one point per cell} \\ \text{Set of solution candidates} \\ \big\downarrow \quad \text{Search satisfiable point} \\ \text{SAT or UNSAT} \end{array}$$

Cylindrical Algebraic Decomposition

A very very very very abstract view:

Set of constraints

$\downarrow$ Characterize truth-invariant cells

Set of polynomials

$\downarrow$ Select one point per cell

Set of solution candidates

$\downarrow$ Search satisfiable point

SAT or UNSAT

What is the argument for answering UNSAT?

Proof sketch for CAD

We found no satisfiable sample point!

Proof sketch for CAD

We found no satisfiable sample point!

Why?

So what?

Proof sketch for CAD

We found no satisfiable sample point!

Why?

Prove evaluation correct
needs RAN reasoning

So what?

Proof sketch for CAD

We found no satisfiable sample point!

Why?

Prove evaluation correct
needs RAN reasoning

So what?

Set of candidates was sufficient!

Proof sketch for CAD

We found no satisfiable sample point!

Why?

Prove evaluation correct
needs RAN reasoning

So what?

Set of candidates was sufficient!

How come?

Proof sketch for CAD

We found no satisfiable sample point!

Why?

Prove evaluation correct
needs RAN reasoning

So what?

Set of candidates was sufficient!

How come?

$\bigcup C = \mathbb{R}^n$
easy if you are careful
non-standard union reasoning

One candidate per cell
needs RAN reasoning

Cells truth-invariant

Proof sketch for CAD

We found no satisfiable sample point!

Why?

Prove evaluation correct
needs RAN reasoning

So what?

Set of candidates was sufficient!

How come?

$\bigcup C = \mathbb{R}^n$
easy if you are careful
non-standard union reasoning

One candidate per cell
needs RAN reasoning

Cells truth-invariant

They are **what**?

Proof sketch for CAD

We found no satisfiable sample point!

Why?

Prove evaluation correct
needs RAN reasoning

So what?

Set of candidates was sufficient!

How come?

$\bigcup C = \mathbb{R}^n$
easy if you are careful
non-standard union reasoning

One candidate per cell
needs RAN reasoning

Cells truth-invariant

They are **what**?

First you compute polynomials in this particular way. Then the theorem of Puiseux with parameters tells you where some factors of these do or do not vanish, and if the coefficients are (complexifications of) real analytic functions, the polynomials are a complex conjucation invariant. With the non-standard valuation for real root isolation to avoid nullification, ...

Proof sketch for CAD

We found no satisfiable sample point!

Why?

Prove evaluation correct
needs RAN reasoning

So what?

Set of candidates was sufficient!

How come?

$\bigcup C = \mathbb{R}^n$
easy if you are careful
non-standard union reasoning

One candidate per cell
needs RAN reasoning

Cells truth-invariant

They are **what**?

First you compute polynomials in this particular way. Then the theorem of Puiseux with parameters tells you where some factors of these do or do not vanish, and if the coefficients are (complexifications of) real analytic functions, the polynomials are a complex conjucation invariant. With the non-standard valuation for real root isolation to avoid nullification, ...

just trust "me"
with this huge monolithic proof

Some attempts

Formalizations within theorem provers:

- RAN reasoning in Coq and Isabelle/HOL
  [Cohen 2012] [Thiemann et al. 2016] [Joosten et al. 2020]
- Sturm sequences in Coq
  [Eberl 2015]
- Implementation of CAD in Coq (no proofs)
  [Mahboubi 2007]

Some attempts

Formalizations within theorem provers:

- RAN reasoning in Coq and Isabelle/HOL
  [Cohen 2012] [Thiemann et al. 2016] [Joosten et al. 2020]
- Sturm sequences in Coq
  [Eberl 2015]
- Implementation of CAD in Coq (no proofs)
  [Mahboubi 2007]

- What if our algorithms are different?
  Lazard valuation; Sturm's theorem vs. Descartes' rule of signs; ...
- Will anyone ever understand these proofs?
- What about other proof checkers?

Cylindrical Algebraic Coverings

Like CAD, but different.

- Regular CDCL(T)-style theory solver
- Algorithm similar to MCSAT / NLSAT
- Theory straight from CAD

[Ábrahám et al. 2021]

Cylindrical Algebraic Coverings

Like CAD, but different.

- Regular CDCL(T)-style theory solver
- Algorithm similar to MCSAT / NLSAT
- Theory straight from CAD

Why should we care?

[Ábrahám et al. 2021]

Cylindrical Algebraic Coverings

Like CAD, but different.

- Regular CDCL(T)-style theory solver
- Algorithm similar to MCSAT / NLSAT
- Theory straight from CAD

Why should we care?

QF_NRA	sat	unsat	solved
cvc5	5137	5596	10733
Yices2	4966	5450	10416
z3	5136	5207	10343
cvc5-old	3421	5376	8797

[Ábrahám et al. 2021]

Fundamental intuition

- Guess partial assignment

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

Fundamental intuition

- Guess partial assignment

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

- Refute partial assignment using intervals

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

Fundamental intuition

- Guess partial assignment

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

- Refute partial assignment using intervals

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

- Lift covering to lower dimension

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

Fundamental intuition

- Guess partial assignment

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

- Refute partial assignment using intervals

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

- Lift covering to lower dimension

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

- Eventually get satisfying assignment or a covering in first dimension

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

Cylindrical Algebraic Coverings in a nutshell

- Fix a variable ordering
- For the kth variable
  - Use constraints to exclude unsatisfiable intervals
  - Guess a value for the kth variable
  - Recurse to $k+1$st variable and obtain
    - a full variable assignment ($\rightarrow$ return SAT)
    - or a covering for the $k+1$st variable
  - Use CAD machinery to infer an interval from this covering
- Until the collected intervals form a covering for the kth variable

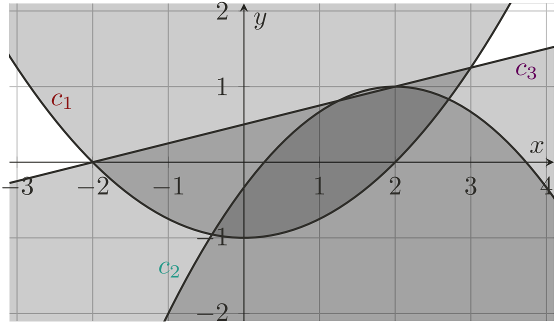
Called for the first variable, we get either

- a model, or
- a conflict (formulated as a covering).

[Ábrahám et al. 2021]

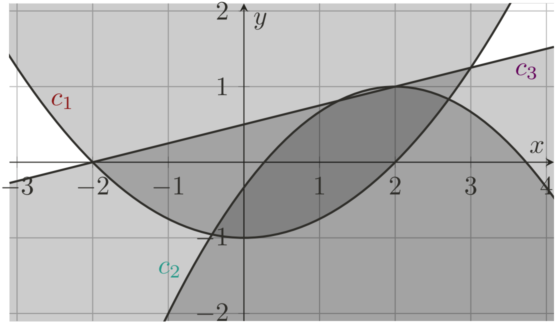
An example

$$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 3 \cdot y > 5 - (x-2)^2 \qquad c_3 : 4 \cdot y > x + 2$$



An example

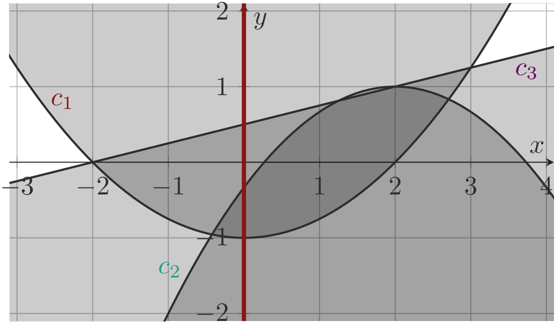
$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 3 \cdot y > 5 - (x-2)^2 \qquad c_3 : 4 \cdot y > x + 2$



No constraint for x

An example

$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 3 \cdot y > 5 - (x-2)^2 \qquad c_3 : 4 \cdot y > x + 2$

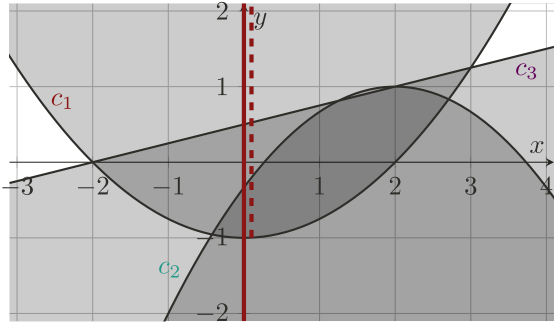


No constraint for x

Guess $x \mapsto 0$

An example

$c_1 : 4 \cdot y < x^2 - 4$ $\quad c_2 : 3 \cdot y > 5 - (x-2)^2$ $\quad c_3 : 4 \cdot y > x + 2$



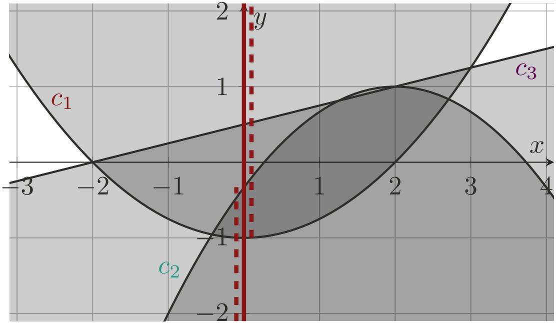
No constraint for x

Guess $x \mapsto 0$

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

An example

$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 3 \cdot y > 5 - (x-2)^2 \qquad c_3 : 4 \cdot y > x + 2$



No constraint for x

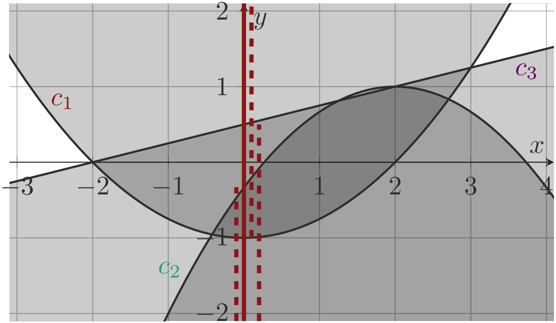
Guess $x \mapsto 0$

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

$c_2 \to y \notin (-\infty, -0.33)$

An example

$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 3 \cdot y > 5 - (x-2)^2 \qquad c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

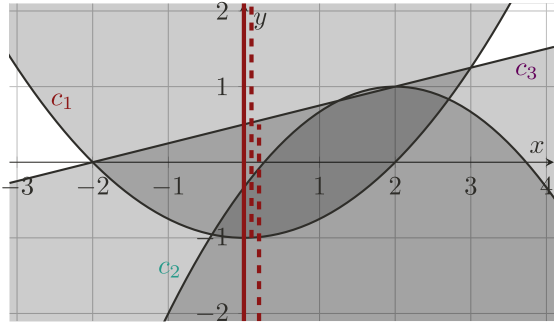
$c_1 \rightarrow y \notin (-1, \infty)$

$c_2 \rightarrow y \notin (-\infty, -0.33)$

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

An example

$c_1 : 4 \cdot y < x^2 - 4$ $\quad$ $c_2 : 3 \cdot y > 5 - (x-2)^2$ $\quad$ $c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

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

$c_2 \to y \notin (-\infty, -0.33)$

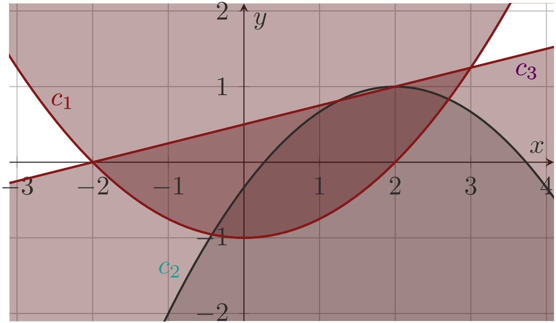
$c_3 \to y \notin (-\infty, 0.5)$

Construct covering

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

An example

$c_1 : 4 \cdot y < x^2 - 4 \qquad c_2 : 3 \cdot y > 5 - (x-2)^2 \qquad c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

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

$c_2 \to y \notin (-\infty, -0.33)$

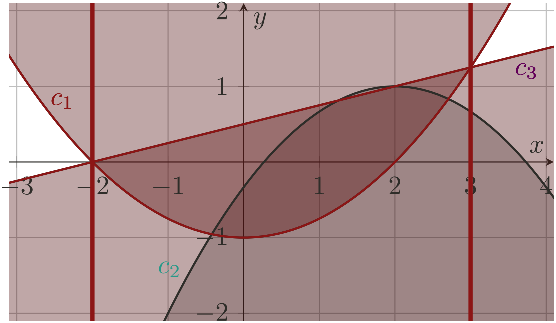
$c_3 \to y \notin (-\infty, 0.5)$

Construct covering

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

An example

$c_1 : 4 \cdot y < x^2 - 4$ $\quad c_2 : 3 \cdot y > 5 - (x-2)^2$ $\quad c_3 : 4 \cdot y > x + 2$



No constraint for x

Guess $x \mapsto 0$

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

$c_2 \to y \notin (-\infty, -0.33)$

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

Construct covering

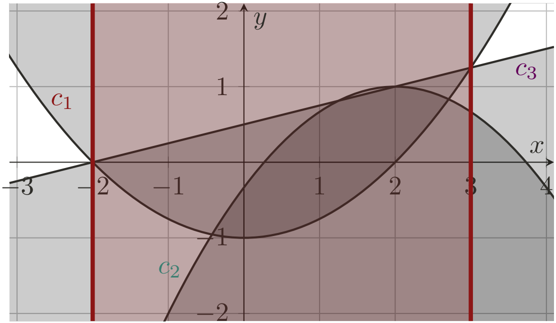
$(-\infty, 0.5), (-1, \infty)$

Construct interval for x

$x \notin (-2, 3)$

An example

$c_1 : 4 \cdot y < x^2 - 4$ $\quad$ $c_2 : 3 \cdot y > 5 - (x-2)^2$ $\quad$ $c_3 : 4 \cdot y > x + 2$



No constraint for x
Guess $x \mapsto 0$
$c_1 \to y \notin (-1, \infty)$
$c_2 \to y \notin (-\infty, -0.33)$
$c_3 \to y \notin (-\infty, 0.5)$
Construct covering
$(-\infty, 0.5), (-1, \infty)$
Construct interval for x
$x \notin (-2, 3)$
New guess for x

The main algorithm

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($(s_1, \ldots, s_{i-1}, s_i), O$)
    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]

The main algorithm

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

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]

The 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]

The main algorithm

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** $(\text{SAT}, (s_1, \ldots, s_{i-1}, s_i))$
  $(f, O) :=$ `get_unsat_cover`$((s_1, \ldots, s_{i-1}, s_i))$
  if $f = \text{SAT}$ **then** **return** (SAT, O)
  else if $f = \text{UNSAT}$ **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 $(\text{UNSAT}, \mathbb{I})$

Real root isolation over a partial sample point

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

Recurse to next variable

[Ábrahám et al. 2021]

The main algorithm

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($(s_1, \ldots,$
    $J :=$ 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

[Ábrahám et al. 2021]

The 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

Extract interval from polynomials

[Ábrahám et al. 2021]

The main algorithm

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 $\mathbb{R} \setminus I$

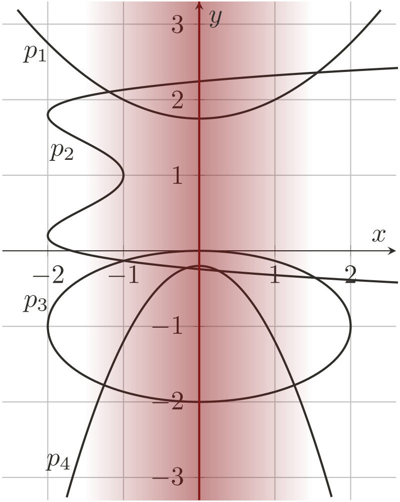
Recurse to next variable

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

Extract interval from polynomials

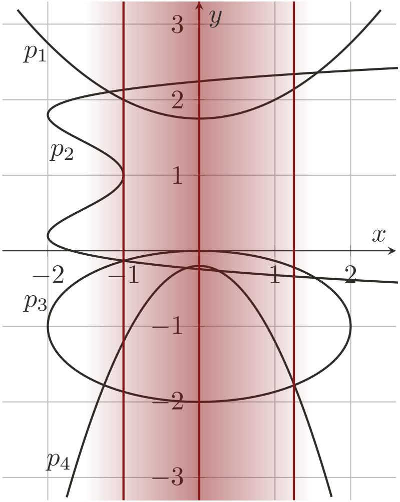
[Ábrahám et al. 2021]

construct_characterization



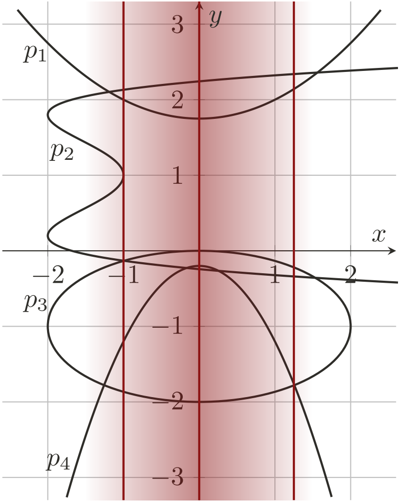
Identify region around sample

construct_characterization



Identify region around sample

construct_characterization

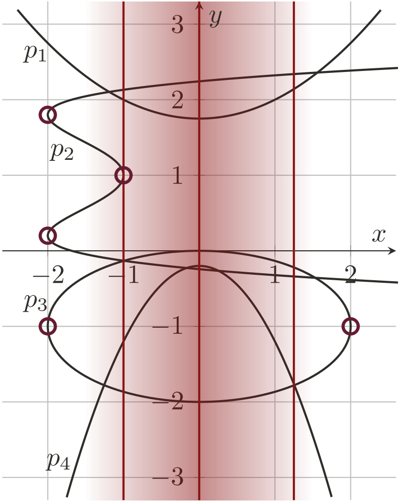


Identify region around sample

CAD projection:

- Discriminants (and coefficients)
- Resultants

construct_characterization

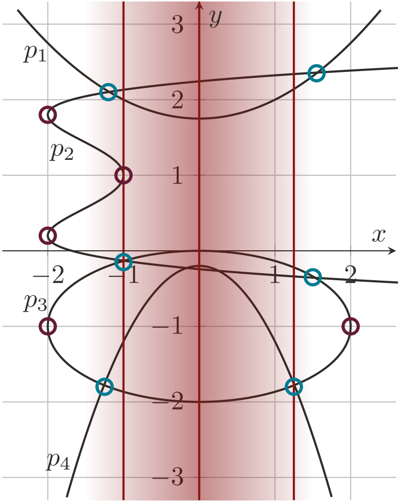


Identify region around sample

CAD projection:

- Discriminants (and coefficients)
- Resultants

construct_characterization

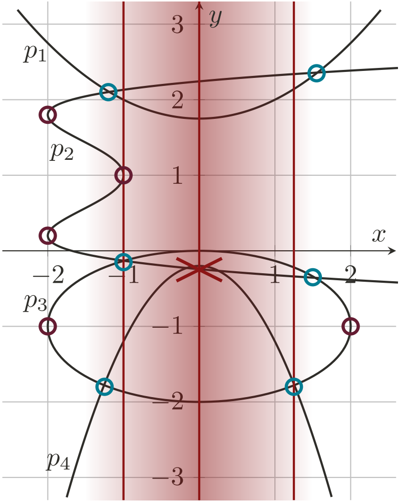


Identify region around sample

CAD projection:

- Discriminants (and coefficients)
- Resultants

construct_characterization



Identify region around sample

CAD projection:

- Discriminants (and coefficients)
- Resultants

Improvement over CAD:

- Resultants between neighbouring intervals only!

Proofs for Cylindrical Algebraic Coverings

Are proofs trivial now?

[Abrahám et al. 2021]

Proofs for Cylindrical Algebraic Coverings

Are proofs trivial now?

No, but:

- Proof is much more constructive
- Hard reasoning is local to a partial assignment
- Feels more natural

[Abrahám et al. 2021]

Proofs for Cylindrical Algebraic Coverings

Are proofs trivial now?

No, but:

- Proof is much more constructive
- Hard reasoning is local to a partial assignment
- Feels more natural

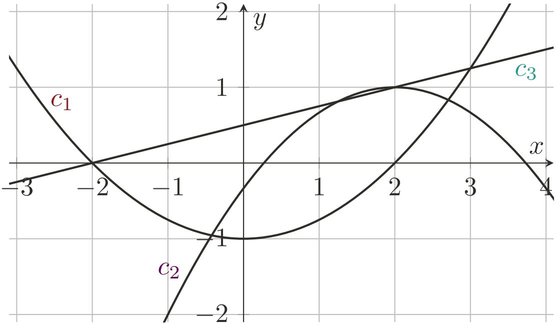
Basically, we claim:

- Algorithm is a proof sketch (CAD is not)
- Proof steps are reasonably intuitive (CAD is not)

[Abrahám et al. 2021]

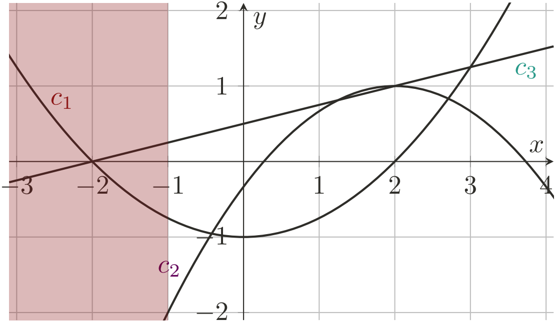
An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



An example

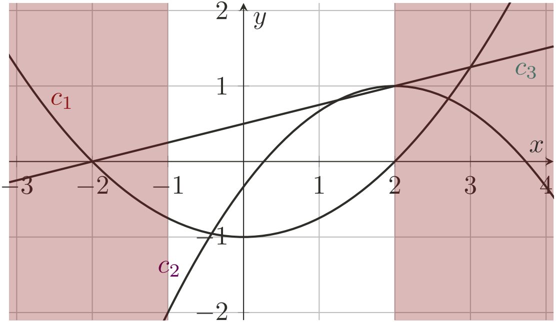
$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



$c_4 \to x \notin (-\infty, -1]$

An example

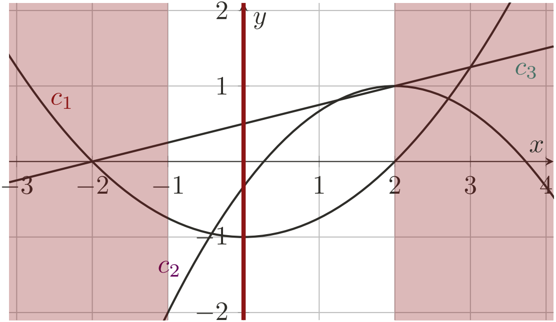
$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



$$c_4 \to x \notin (-\infty, -1]$$
$$c_5 \to x \notin [2, \infty)$$

An example

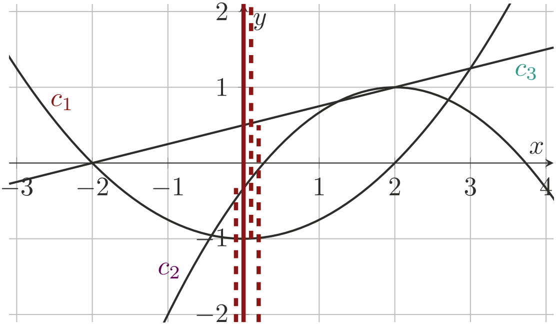
$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$

An example

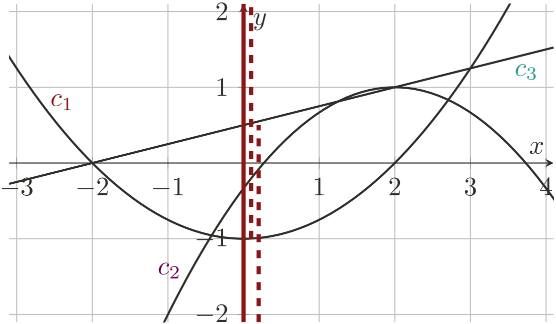
$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \to y \notin (-1, \infty)$
$\quad c_2 \to y \notin (-\infty, -0.33)$
$\quad c_3 \to y \notin (-\infty, 0.5)$

An example

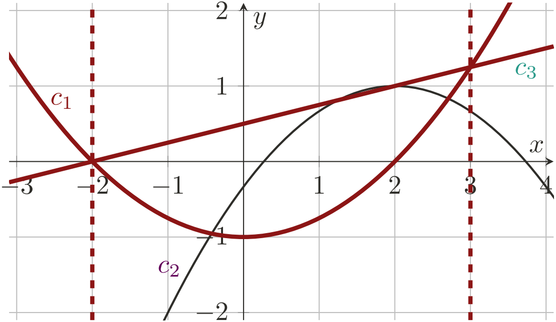
$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \to y \notin (-1, \infty)$
$\quad \cancel{c_2 \to y \notin (-\infty, -0.33)}$
$\quad c_3 \to y \notin (-\infty, 0.5)$

An example

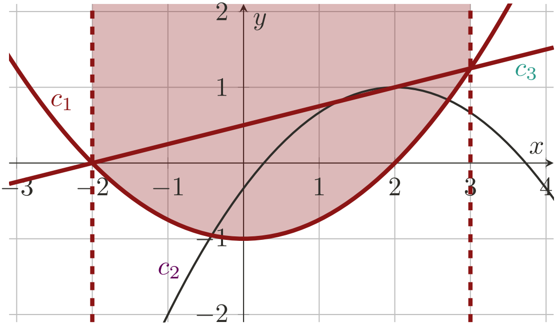
$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \to y \notin (-1, \infty)$
$\quad \cancel{c_2 \to y \notin (-\infty, -0.33)}$
$\quad c_3 \to y \notin (-\infty, 0.5)$

An example

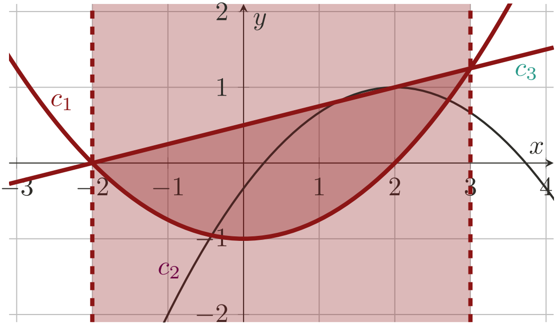
$c_1 : 4 \cdot y < x^2 - 4$ $c_2 : 3 \cdot y > 5 - (x-2)^2$ $c_3 : 4 \cdot y > x + 2$ $c_4 : x > -1$ $c_5 : x < 2$



$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$c_1 \to y \notin (-1, \infty)$
$\cancel{c_2 \to y \notin (-\infty, -0.33)}$
$c_3 \to y \notin (-\infty, 0.5)$

An example

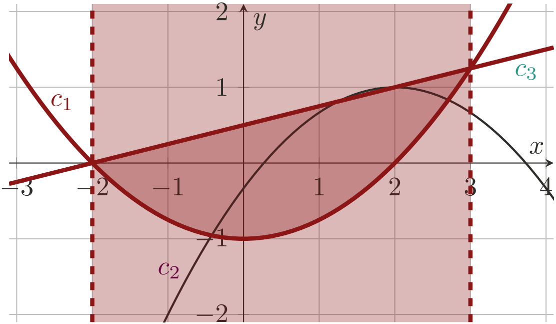
$c_1 : 4 \cdot y < x^2 - 4$ $c_2 : 3 \cdot y > 5 - (x-2)^2$ $c_3 : 4 \cdot y > x + 2$ $c_4 : x > -1$ $c_5 : x < 2$



$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$c_1 \to y \notin (-1, \infty)$
$\cancel{c_2 \to y \notin (-\infty, -0.33)}$
$c_3 \to y \notin (-\infty, 0.5)$

An example

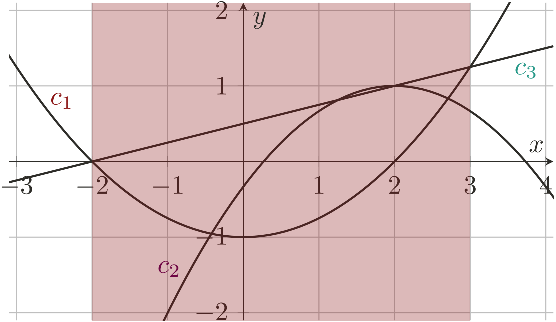
$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \to y \notin (-1, \infty)$
$\quad \cancel{c_2 \to y \notin (-\infty, -0.33)}$
$\quad c_3 \to y \notin (-\infty, 0.5)$
$\quad y \notin \mathbb{R}$

An example

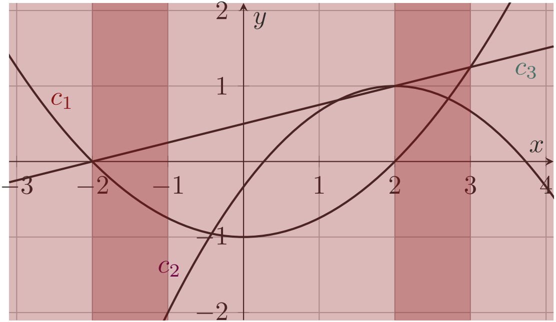
$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



$c_4 \rightarrow x \notin (-\infty, -1]$
$c_5 \rightarrow x \notin [2, \infty)$
$x \mapsto 0$
$c_1 \rightarrow y \notin (-1, \infty)$
$\cancel{c_2 \rightarrow y \notin (-\infty, -0.33)}$
$c_3 \rightarrow y \notin (-\infty, 0.5)$
$y \notin \mathbb{R}$
$c_1, c_2 \rightarrow x \notin (-2, 3)$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$



$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \to y \notin (-1, \infty)$
$\quad \cancel{c_2 \to y \notin (-\infty, -0.33)}$
$\quad c_3 \to y \notin (-\infty, 0.5)$
$\quad y \notin \mathbb{R}$
$c_1, c_2 \to x \notin (-2, 3)$
$x \notin \mathbb{R}$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$

$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \to y \notin (-1, \infty)$
$\quad \cancel{c_2 \to y \notin (-\infty, -0.33)}$
$\quad c_3 \to y \notin (-\infty, 0.5)$
$\quad y \notin \mathbb{R}$
$c_1, c_2 \to x \notin (-2, 3)$
$x \notin \mathbb{R}$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$

RECURSE

$$
\begin{aligned}
& c_4 \rightarrow x \notin (-\infty, -1] \\
& c_5 \rightarrow x \notin [2, \infty) \\
& x \mapsto 0 \\
& \quad c_1 \rightarrow y \notin (-1, \infty) \\
& \quad \cancel{c_2 \rightarrow y \notin (-\infty, -0.33)} \\
& \quad c_3 \rightarrow y \notin (-\infty, 0.5) \\
& \quad y \notin \mathbb{R} \\
& c_1, c_2 \rightarrow x \notin (-2, 3) \\
& x \notin \mathbb{R}
\end{aligned}
$$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$

RECURSE

DIRECT c_4
$\Rightarrow \neg(x \leq -1)$

$c_4 \rightarrow x \notin (-\infty, -1]$
$c_5 \rightarrow x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \rightarrow y \notin (-1, \infty)$
$\quad \cancel{c_2 \rightarrow y \notin (-\infty, -0.33)}$
$\quad c_3 \rightarrow y \notin (-\infty, 0.5)$
$\quad y \notin \mathbb{R}$
$c_1, c_2 \rightarrow x \notin (-2, 3)$
$x \notin \mathbb{R}$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$

RECURSE

DIRECT c_4
$\Rightarrow \neg(x \leq -1)$

DIRECT c_5
$\Rightarrow \neg(x \geq 2)$

$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \to y \notin (-1, \infty)$
$\quad \cancel{c_2 \to y \notin (-\infty, -0.33)}$
$\quad c_3 \to y \notin (-\infty, 0.5)$
$\quad y \notin \mathbb{R}$
$c_1, c_2 \to x \notin (-2, 3)$
$x \notin \mathbb{R}$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$

RECURSE

- DIRECT c_4 $\Rightarrow \neg(x \leq -1)$
- RECURSE $x \mapsto 0$
- DIRECT c_5 $\Rightarrow \neg(x \geq 2)$

$$
\begin{array}{l}
c_4 \rightarrow x \notin (-\infty, -1] \\
c_5 \rightarrow x \notin [2, \infty) \\
x \mapsto 0 \\
\quad c_1 \rightarrow y \notin (-1, \infty) \\
\quad \cancel{c_2 \rightarrow y \notin (-\infty, -0.33)} \\
\quad c_3 \rightarrow y \notin (-\infty, 0.5) \\
\quad y \notin \mathbb{R} \\
c_1, c_2 \rightarrow x \notin (-2, 3) \\
x \notin \mathbb{R}
\end{array}
$$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$

RECURSE

DIRECT c_4
$\Rightarrow \neg(x \leq -1)$

RECURSE $x \mapsto 0$

DIRECT c_5
$\Rightarrow \neg(x \geq 2)$

DIRECT $c_1, x \mapsto 0$
$\Rightarrow \neg(y > -1)$

DIRECT $c_3, x \mapsto 0$
$\Rightarrow \neg(y < 0.5)$

DIRECT $c_2, x \mapsto 0$
$\Rightarrow \neg(y < -0.33)$

$c_4 \to x \notin (-\infty, -1]$
$c_5 \to x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \to y \notin (-1, \infty)$
$\quad \cancel{c_2 \to y \notin (-\infty, -0.33)}$
$\quad c_3 \to y \notin (-\infty, 0.5)$
$\quad y \notin \mathbb{R}$
$c_1, c_2 \to x \notin (-2, 3)$
$x \notin \mathbb{R}$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$

RECURSE

DIRECT c_4 $\Rightarrow \neg(x \leq -1)$

RECURSE $x \mapsto 0$

DIRECT c_5 $\Rightarrow \neg(x \geq 2)$

DIRECT $c_1, x \mapsto 0$ $\Rightarrow \neg(y > -1)$

DIRECT $c_3, x \mapsto 0$ $\Rightarrow \neg(y < 0.5)$

$c_4 \rightarrow x \notin (-\infty, -1]$
$c_5 \rightarrow x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \rightarrow y \notin (-1, \infty)$
$\quad \cancel{c_2 \rightarrow y \notin (-\infty, -0.33)}$
$\quad c_3 \rightarrow y \notin (-\infty, 0.5)$
$\quad y \notin \mathbb{R}$
$c_1, c_2 \rightarrow x \notin (-2, 3)$
$x \notin \mathbb{R}$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$

RECURSE

DIRECT c_4 $\Rightarrow \neg(x \leq -1)$

RECURSE c_1, c_3 $\Rightarrow \neg(-2 < x < 3)$

DIRECT c_5 $\Rightarrow \neg(x \geq 2)$

DIRECT $c_1, (-2 < x < 3)$ $\Rightarrow \neg(y > -1)$

DIRECT $c_3, (-2 < x < 3)$ $\Rightarrow \neg(y < 0.5)$

$c_4 \rightarrow x \notin (-\infty, -1]$
$c_5 \rightarrow x \notin [2, \infty)$
$x \mapsto 0$
$\quad c_1 \rightarrow y \notin (-1, \infty)$
$\quad \cancel{c_2 \rightarrow y \notin (-\infty, -0.33)}$
$\quad c_3 \rightarrow y \notin (-\infty, 0.5)$
$\quad y \notin \mathbb{R}$
$c_1, c_2 \rightarrow x \notin (-2, 3)$
$x \notin \mathbb{R}$

An example

$c_1 : 4 \cdot y < x^2 - 4 \quad c_2 : 3 \cdot y > 5 - (x-2)^2 \quad c_3 : 4 \cdot y > x + 2 \quad c_4 : x > -1 \quad c_5 : x < 2$

RECURSE c_1, c_3, c_4, c_5
$\Rightarrow \neg(true)$

- DIRECT c_4
  $\Rightarrow \neg(x \leq -1)$
- RECURSE c_1, c_3
  $\Rightarrow \neg(-2 < x < 3)$
  - DIRECT $c_1, (-2 < x < 3)$
    $\Rightarrow \neg(y > -1)$
  - DIRECT $c_3, (-2 < x < 3)$
    $\Rightarrow \neg(y < 0.5)$
- DIRECT c_5
  $\Rightarrow \neg(x \geq 2)$

$c_4 \rightarrow x \notin (-\infty, -1]$
$c_5 \rightarrow x \notin [2, \infty)$
$x \mapsto 0$
  $c_1 \rightarrow y \notin (-1, \infty)$
  $\cancel{c_2 \rightarrow y \notin (-\infty, -0.33)}$
  $c_3 \rightarrow y \notin (-\infty, 0.5)$
  $y \notin \mathbb{R}$
$c_1, c_2 \rightarrow x \notin (-2, 3)$
$x \notin \mathbb{R}$

How it looks in cvc5*

```
(SCOPE (c1 c3 c4 c5)
  (ARITH_NL_CAD_RECURSIVE
    (SCOPE ((ROOT_PREDICATE k=1 (<= x 0) (+ 1 (* 1 x))))
      (ARITH_NL_CAD_DIRECT
        (ASSUME (ROOT_PREDICATE k=1 (<= x 0) (+ 1 (* 1 x))))
        (ASSUME c4)))
    (SCOPE ((ROOT_PREDICATE k=1 (>= x 0) (+ (- 2) (* 1 x))))
      (ARITH_NL_CAD_DIRECT
        (ASSUME (ROOT_PREDICATE k=1 (>= x 0) (+ (- 2) (* 1 x))))
        (ASSUME c5)))
    (SCOPE ((ROOT_PREDICATE k=1 (> x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2))))
            (ROOT_PREDICATE k=2 (< x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
      (ARITH_NL_CAD_RECURSIVE
        (ASSUME (ROOT_PREDICATE k=1 (> x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
        (ASSUME (ROOT_PREDICATE k=2 (< x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
        (SCOPE ((ROOT_PREDICATE k=1 (<= y 0) (+ (- 2) (* (- 1) x) (* 4 y))))
          (ARITH_NL_CAD_DIRECT
            (ASSUME (ROOT_PREDICATE k=1 (> x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
            (ASSUME (ROOT_PREDICATE k=2 (< x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
            (ASSUME (ROOT_PREDICATE k=1 (<= y 0) (+ (- 2) (* (- 1) x) (* 4 y))))
            (ASSUME c3)))
        (SCOPE ((ROOT_PREDICATE k=1 (>= y 0) (+ 4 (* (- 1) (^ x 2)) (* 4 y))))
          (ARITH_NL_CAD_DIRECT
            (ASSUME (ROOT_PREDICATE k=1 (> x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
            (ASSUME (ROOT_PREDICATE k=2 (< x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
            (ASSUME (ROOT_PREDICATE k=1 (>= y 0) (+ 4 (* (- 1) (^ x 2)) (* 4 y))))
            (ASSUME c1))))))
```

Hard reasoning is still there... but localized

```
(SCOPE ((ROOT_PREDICATE k=1 (<= y 0) (+ (- 2) (* (- 1) x) (* 4 y))))
  (ARITH_NL_CAD_DIRECT
    (ASSUME (ROOT_PREDICATE k=1 (> x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
    (ASSUME (ROOT_PREDICATE k=2 (< x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
    (ASSUME (ROOT_PREDICATE k=1 (<= y 0) (+ (- 2) (* (- 1) x) (* 4 y))))
    (ASSUME c3 )))
```

Hard reasoning is still there... but localized

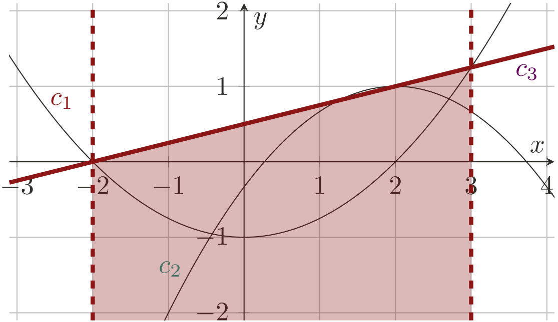
```
(SCOPE ((ROOT_PREDICATE k=1 (<= y 0) (+ (- 2) (* (- 1) x) (* 4 y))))
  (ARITH_NL_CAD_DIRECT
    (ASSUME (ROOT_PREDICATE k=1 (> x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
    (ASSUME (ROOT_PREDICATE k=2 (< x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
    (ASSUME (ROOT_PREDICATE k=1 (<= y 0) (+ (- 2) (* (- 1) x) (* 4 y))))
    (ASSUME c3 )))
```

$$(c_3 \wedge x > \mathrm{Root}_1(x^2 - x - 6) \wedge x < \mathrm{Root}_2(x^2 - x - 6)) \Rightarrow \neg(y \leq \mathrm{Root}_1(4y - x - 2))$$

Hard reasoning is still there... but localized

```
(SCOPE ((ROOT_PREDICATE k=1 (<= y 0) (+ (- 2) (* (- 1) x) (* 4 y))))
  (ARITH_NL_CAD_DIRECT
    (ASSUME (ROOT_PREDICATE k=1 (> x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
    (ASSUME (ROOT_PREDICATE k=2 (< x 0) (+ (- 6) (* (- 1) x) (* 1 (^ x 2)))))
    (ASSUME (ROOT_PREDICATE k=1 (<= y 0) (+ (- 2) (* (- 1) x) (* 4 y))))
    (ASSUME c3 )))
```

$$(c_3 \wedge x > \mathrm{Root}_1(x^2 - x - 6) \wedge x < \mathrm{Root}_2(x^2 - x - 6)) \Rightarrow \neg(y \leq \mathrm{Root}_1(4y - x - 2))$$



Conclusion

- Theory lemmas are usually easy to prove
- CAD-based lemmas are hard to prove
- Coverings provide a proof skeleton

Conclusion

- Theory lemmas are usually easy to prove
- CAD-based lemmas are hard to prove
- Coverings provide a proof skeleton

Open questions:

- How can we make these proofs more accessible?
- Which parts of CAD theory are really necessary for proofs?
- What can we reasonably expect proof checkers to know?
- Are automatically verifiable CAD-based proofs feasible? When?

Conclusion

- Theory lemmas are usually easy to prove
- CAD-based lemmas are hard to prove
- Coverings provide a proof skeleton

Open questions:

- How can we make these proofs more accessible?
- Which parts of CAD theory are really necessary for proofs?
- What can we reasonably expect proof checkers to know?
- Are automatically verifiable CAD-based proofs feasible? When?

Any ideas?

References I

- Erika Abrahám, James H Davenport, Matthew England, and Gereon Kremer. “Proving UNSAT in SMT: The Case of Quantifier Free Non-Linear Real Arithmetic”. In: arXiv preprint arXiv:2108.05320 (2021).
- 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.
- Cyril Cohen. “Construction of Real Algebraic Numbers in Coq”. In: 2012, pp. 67–82.
- Manuel Eberl. “A Decision Procedure for Univariate Real Polynomials in Isabelle/HOL”. In: 2015, pp. 75–83. DOI: 10.1145/2676724.2693166. URL: https://doi.org/10.1145/2676724.2693166.
- Sebastiaan J. C. Joosten, René Thiemann, and Akihisa Yamada. “A Verified Implementation of Algebraic Numbers in Isabelle/HOL”. In: Journal of Automated Reasoning 64 (2020), pp. 363–389. DOI: 10.1007/s10817-018-09504-w.
- Assia Mahboubi. “Implementing the cylindrical algebraic decomposition within the Coq system”. In: Mathematical Structures in Computer Science 17 (1 2007), pp. 99–127. DOI: 10.1017/S096012950600586X.

References II

- René Thiemann and Akihisa Yamada. "Algebraic Numbers in Isabelle/HOL". In: 2016, pp. 391–408.