Satisfiability Checking

Summary III

Prof. Dr. Erika Ábrahám

RWTH Aachen University
Informatik 2
LuFG Theory of Hybrid Systems

WS 19/20

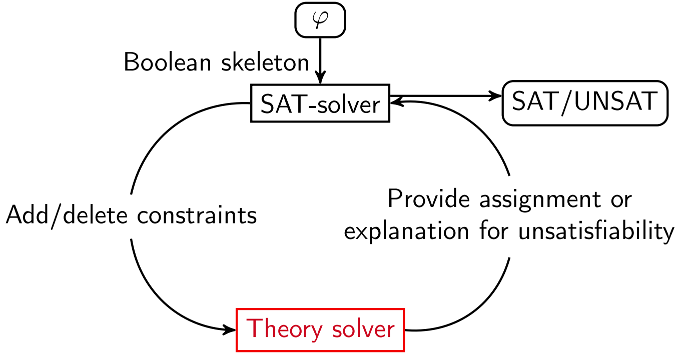
Non-linear real arithmetic

We consider input formulae φ from the theory of quantifier-free nonlinear real arithmetic (QFNRA):

$$
\begin{array}{lll}
p & := const \mid x \mid (p+p) \mid (p-p) \mid (p \cdot p) & \text{polynomials} \\
c & := p < 0 \mid p = 0 & \text{(polynomial) constraints} \\
\varphi & := c \mid (\varphi \wedge \varphi) \mid \neg\varphi & \text{QFNRA formulas}
\end{array}
$$

where constants *const* and variables x take real values from $\mathbb{R}$.

Connection to SMT



Solves $\begin{pmatrix} p_1 & \sim_1 & 0 \\ & \vdots & \\ p_m & \sim_m & 0 \end{pmatrix}$ where $\begin{matrix} p_i \in \mathbb{Z}[x_1, \ldots, x_n], \\ \sim_i \quad \in \{<, \leq, =, \neq, \geq, >\} \end{matrix}$ for $1 \leq i \leq m$.

1. Interval constraint propagation
2. Subtropical satisfiability
3. Virtual substitution
4. Cylindrical algebraic decomposition

Interval constraint propagation (ICP)

- Incomplete but cheap method.
- Basic idea:
  Start with a list containing a single initial box (value domain).
  Use the input constraints to contract a non-empty box from the list.
  If no contraction possible, split a non-empty box.
- Termination: all boxes are empty (UNSAT) or there is a sufficiently small non-empty box (possibly SAT).

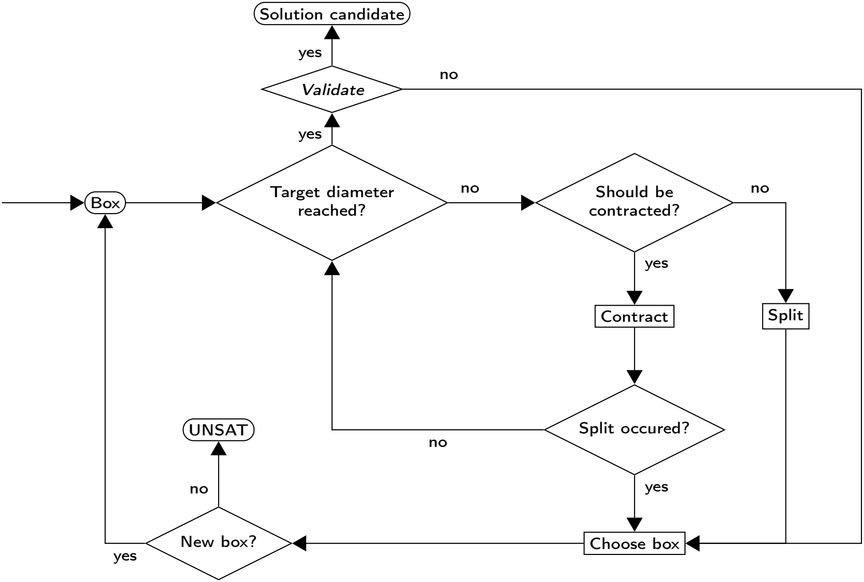
Interval constraint propagation (ICP)

- Incomplete but cheap method.
- Basic idea:
  Start with a list containing a single initial box (value domain).
  Use the input constraints to contract a non-empty box from the list.
  If no contraction possible, split a non-empty box.
- Termination: all boxes are empty (UNSAT) or there is a sufficiently small non-empty box (possibly SAT).

First contraction approach: Interval arithmetic

Second contraction approach: Interval Newton method

Algorithm overview



Contraction I: Preprocessing

- Set $C' := C$ and $C := \emptyset$.
- Repeat as long as C' is not empty:
  - Take a constraint $e_1 \sim e_2$, $\sim \in \{<, \leq, =, \geq, >\}$, from C'.
  - Bring $e_1 \sim e_2$ to the normal form $r_1 \cdot m_1 + \ldots + r_k \cdot m_k \sim 0$, where $r_i \in \mathbb{R}$ and m_i are monomials (either 1 or a product of variables) for each $i = 1, \ldots, k$.
  - Replace each non-linear monomial m_i in $r_1 \cdot m_1 + \ldots + r_k \cdot m_k \sim 0$ by a fresh variable h_i and add the result to C.
  - For each newly added variable h_i replacing m_i in the previous step, add an equation $h_i - m_i = 0$ to C, and initialize the bounds of h_i to the interval we get when we substitute the variable bounds in m_i and evaluate the result using interval arithmetic (note: the result will always be a single interval because there is no division or square root in m_i).

Contraction I: Interval arithmetic

- Step 1: Partially extend real arithmetic operations to $\mathbb{R} \cup \{-\infty, +\infty\}$.
- Step 2: Extend real arithmetic operations to intervals (interval arithmetic).

Definition (Interval arithmetic)

Assume real intervals $A = [\underline{A}, \overline{A}]$ and $B = [\underline{B}, \overline{B}]$.

$$
\begin{array}{ll}
A + B & = [\underline{A} + \underline{B}; \overline{A} + \overline{B}] \\
A - B & = [\underline{A} - \overline{B}; \overline{A} - \underline{B}] \\
A \cdot B & = [min(\underline{A} \cdot \underline{B}, \underline{A} \cdot \overline{B}, \overline{A} \cdot \underline{B}, \overline{A} \cdot \overline{B}); max(\underline{A} \cdot \underline{B}, \underline{A} \cdot \overline{B}, \overline{A} \cdot \underline{B}, \overline{A} \cdot \overline{B})] \\
A^2 & = (A \cdot A) \cap [0; +\infty) \\
A \div B & = A \cdot \frac{1}{B} = A \cdot [\frac{1}{\overline{B}}; \frac{1}{\underline{B}}] \text{ if } 0 \notin B \; \textit{(extended interval division if } 0 \in B\textit{)}
\end{array}
$$

Contraction I: Method

- Choose a constraint $c \in C$ and a variable x appearing in c.

  We call such a pair (c, x) a contraction candidate (CC).
- Bring c to a form $x \sim e$, $\sim \in \{<, \leq, =, \geq, >\}$, where e does not contain x. (Note: due to preprocessing, if c is non-linear then it is of the form $h - m = 0$ with h a variable and m a monomial.)
- Replace all variables in e by their current bounds.
- Apply interval arithmetic to evaluate the right-hand-side (e with the variables substituted by their bounds) to a union of intervals.
- Make a case distinction for each interval B in that union.
- For each case, derive from the current bound A for x and the computed bound B for e a new bound on x, depending on the type of $\sim$, as follows:

$$
\begin{array}{lll}
x < e & \textit{if } \underline{A} \geq \overline{B} \textit{ then } \emptyset \textit{ else} & [\underline{A}, \min\{\overline{A}, \overline{B}\}] \\
x \leq e & & [\underline{A}, \min\{\overline{A}, \overline{B}\}] \\
x = e & & [\max\{\underline{A}, \underline{B}\}, \min\{\overline{A}, \overline{B}\}] \\
x \geq e & & [\max\{\underline{A}, \underline{B}\}, \overline{A}] \\
x > e & \textit{if } \overline{A} \leq \underline{B} \textit{ then } \emptyset \textit{ else} & [\max\{\underline{A}, \underline{B}\}, \overline{A}]
\end{array}
$$

Contraction II: Preprocessing

- This second method is called the interval Newton method.
- Also this second propagation method needs some lightweight preprocessing:
  - Transform each constraint $e_1 \sim e_2$ in C to $e_1 - e_2 \sim 0$.
  - For each inequation $p \sim 0$ with $\sim \in \{<, \leq, \geq, >\}$ in C, replace p by a fresh variable h, add an equation $h - p = 0$ to C, and initialize the bounds of h to the interval we get when we substitute the variable bounds in p and evaluate the result using interval arithmetic (note: the result will always be a single interval because there is no division or square root in p).
- After this preprocessing, the constraint set contains equations $p = 0$ stating that a polynomial equals to zero, and inequations of the form $x \sim 0$ with x a variable and $\sim \in \{<, \leq, \geq, >\}$.
- Assume in the following a constraint $c \in C$ and a variable x in c as a contraction candidate.

Contraction II: Method

If the constraint c is an inequation then it has the form $x \sim 0$ (where x is a variable). Contraction (assuming that the current interval for x is A):

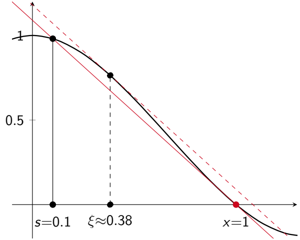
$$
\begin{array}{lll}
x < 0 & \textit{if } \underline{A} \geq 0 \textit{ then } \emptyset \textit{ else} & [\underline{A}, \min\{\overline{A}, 0\}] \\
x \leq 0 & & [\underline{A}, \min\{\overline{A}, 0\}] \\
x \geq 0 & & [\max\{\underline{A}, 0\}, \overline{A}] \\
x > 0 & \textit{if } \overline{A} \leq 0 \textit{ then } \emptyset \textit{ else} & [\max\{\underline{A}, 0\}, \overline{A}]
\end{array}
$$

Contraction II: Method

Assume now that the constraint c is an equation $f(x) = 0$ (with f being a polynomial).

Interval Newton method for the univariate case:

- Input:
  - interval A
  - univariate polynomial constraint $f(x) = 0$
  - sample point $s \in A$
- Output: contracted interval $A = s - \frac{f(s)}{f'(A)}$ (where $f'(x)$ is the first derivative of $f(x)$)



Contraction II: Componentwise multivariate interval Newton

Componentwise multivariate interval Newton:

- Input:
  - interval $A = A_1 \times \ldots A_n$
  - multivariate polynomial constraint $f(x_1, \ldots, x_n) = 0$
  - sample point $s = (s_1, \ldots, s_n) \in A$
  - variable x_j
- Output: contracted interval $A = s - \frac{f(A_1,\ldots,A_{j-1},s_j,A_{j+1},\ldots,A_n)}{\frac{\partial f}{\partial x_j}(A_1,\ldots,A_n)}$

Heuristics to choose CCs

- Relative contraction

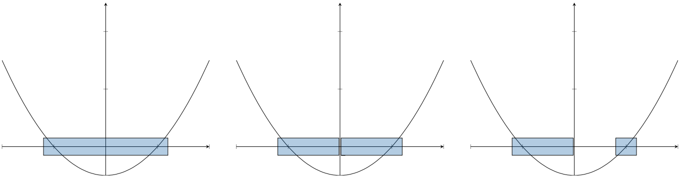
$$gain_{rel} = \frac{D_{old} - D_{new}}{D_{old}} = 1 - \frac{D_{new}}{D_{old}}$$

  is in general not predictable.
- Heuristics:
  - assign a weight $W_k^{(ij)} \in [0;1]$ to each CC
  - select the next contraction candidate with the highest weight
  - CCs with a weight less than some threshold ε are not considered for contraction
  - let $r_{k+1}^{(ij)} \in [0;1]$ be the achieved relative contraction
  - update weight:

$$W_{k+1}^{(ij)} = W_k^{(ij)} + \alpha(r_{k+1}^{(ij)} - W_k^{(ij)})$$

Assure termination

When the weight of all CCs is below the threshold we do not make progress
$\rightarrow$ split the box.



Handling linear constraints

ICP is not well-suited for linear problems (slow convergence).

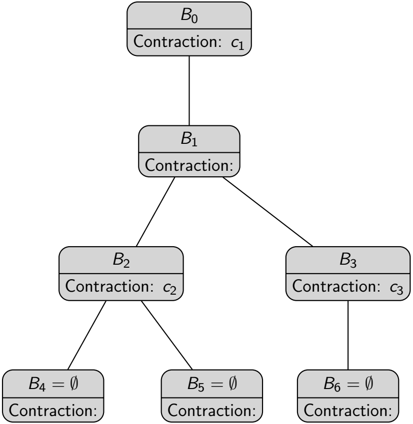
Make use of linear solvers (e.g. simplex) for linear constraints:

- Pre-process to separate linear and nonlinear constraints
- Use nonlinear constraints for contraction
- Validate resulting boxes against linear feasible region
- Box infeasible $\rightarrow$ add violated linear constraint for contraction

Incrementality and Explanations

We store the search history in a tree-structure. Each node stores information about one loop iteration:

- the box chosen and
- the constraint used for contraction if any.

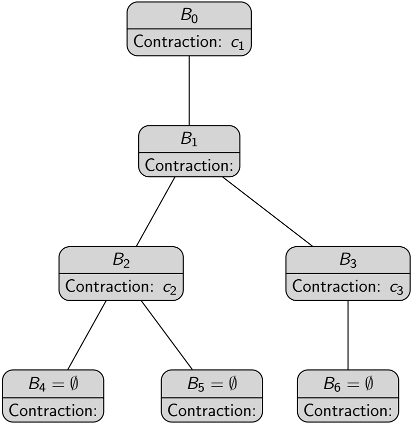


Incrementality and Explanations

We store the search history in a tree-structure. Each node stores information about one loop iteration:

- the box chosen and
- the constraint used for contraction if any.

Incrementality: Extend the tree.



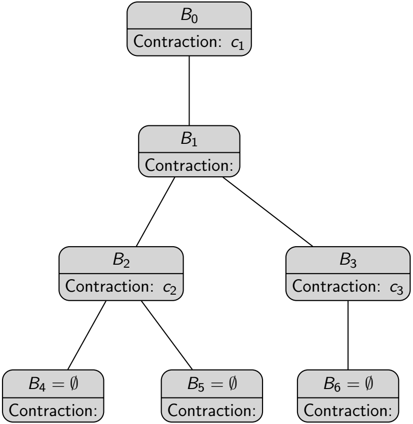
Incrementality and Explanations

We store the search history in a tree-structure. Each node stores information about one loop iteration:

- the box chosen and
- the constraint used for contraction if any.

Incrementality: Extend the tree.

Explanation: collect all constraints mentioned in the tree.



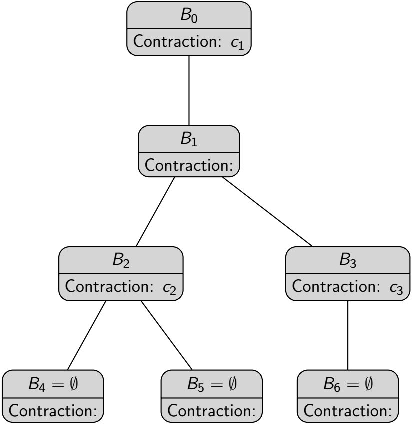
Incrementality and Explanations

We store the search history in a tree-structure. Each node stores information about one loop iteration:

- the box chosen and
- the constraint used for contraction if any.

Incrementality: Extend the tree.

Explanation: collect all constraints mentioned in the tree.



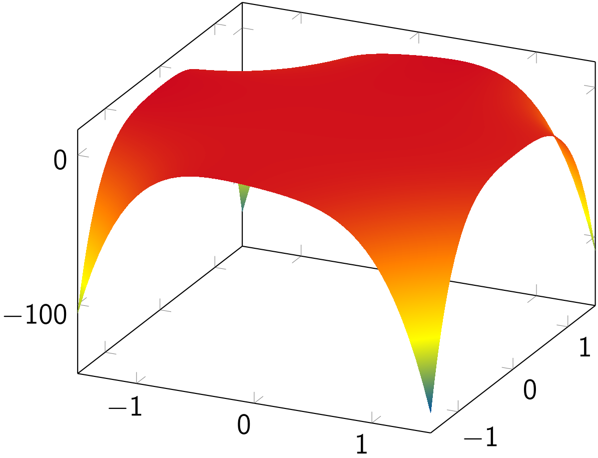
1. Interval constraint propagation

2. Subtropical satisfiability

3. Virtual substitution

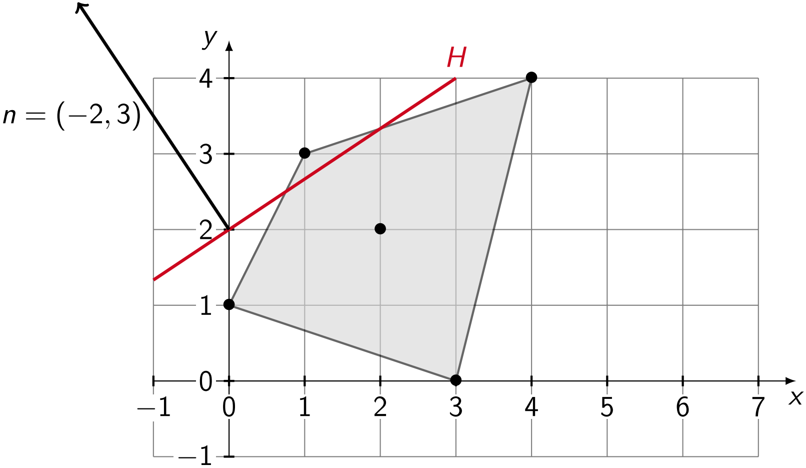
4. Cylindrical algebraic decomposition

Intuition



$$f(x, y) = y + 2xy^3 - 3x^2y^2 - x^3 - 4x^4y^4$$

Hyperplanes separating vertices of the Newton polytope



Virtual substitution

- Virtual substitution method: quantifier elimination procedure for real arithmetic formulas
- Here: only existential quantification, no free variables

$$\exists x_1. \dots \exists x_n . \varphi_n \quad \equiv \quad \exists x_1. \dots \exists x_{n-1} . \varphi_{n-1}$$

- Restriction: applicable only to variables that appear at most quadratic in the formula
- Basic idea: use solution equation to construct a finite set $T \subset \mathbb{R}$ of test candidates for x_n, and use virtual substitution to check whether one of the test candidates satisfies the formula:

$$\exists x_1. \dots \exists x_n . \varphi_n \quad \equiv \quad \exists x_1. \dots \exists x_{n-1} . \bigvee_{t \in T} \varphi_n[t /\!/ x_n].$$

Construction of the set of test candidates T

Given: A constraint $p \sim 0, \quad p = ax^2 + bx + c, \quad \sim \in \{=, <, >, \leq, \geq, \neq\}$.

Construction of the set of test candidates T

Given: A constraint $p \sim 0, \quad p = ax^2 + bx + c, \quad \sim \in \{=, <, >, \leq, \geq, \neq\}$.
Possible zeros of p (in x) are:

Construction of the set of test candidates T

Given: A constraint $p \sim 0$, $p = ax^2 + bx + c$, $\sim \in \{=, <, >, \leq, \geq, \neq\}$.
Possible zeros of p (in x) are:

Linear in x :	$x_0 = -\frac{c}{b}$	, if	$a = 0 \land b \neq 0$
Quadratic in x :	$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \land b^2 - 4ac \geq 0$
	$x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \land b^2 - 4ac > 0$

Construction of the set of test candidates T

Given: A constraint $p \sim 0$, $\quad p = ax^2 + bx + c$, $\quad \sim \in \{=, <, >, \leq, \geq, \neq\}$.
Possible zeros of p (in x) are:

Linear in x :	$x_0 = -\frac{c}{b}$	, if	$a = 0 \wedge b \neq 0$
Quadratic in x :	$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac \geq 0$
	$x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac > 0$

The finite endpoints of possible solution intervals of $p \sim 0$ are

Construction of the set of test candidates T

Given: A constraint $p \sim 0$, $p = ax^2 + bx + c$, $\sim \in \{=, <, >, \leq, \geq, \neq\}$.
Possible zeros of p (in x) are:

Linear in x :	$x_0 = -\frac{c}{b}$	, if	$a = 0 \wedge b \neq 0$
Quadratic in x :	$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac \geq 0$
	$x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac > 0$

The finite endpoints of possible solution intervals of $p \sim 0$ are the zeros of p (as the sign of p is invariant between its zeros).

Construction of the set of test candidates T

Given: A constraint $p \sim 0$, $p = ax^2 + bx + c$, $\sim \in \{=, <, >, \leq, \geq, \neq\}$.
Possible zeros of p (in x) are:

Linear in x :	$x_0 = -\frac{c}{b}$	, if	$a = 0 \wedge b \neq 0$
Quadratic in x :	$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac \geq 0$
	$x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac > 0$

The finite endpoints of possible solution intervals of $p \sim 0$ are the zeros of p (as the sign of p is invariant between its zeros).

Note: If p has no zeros then the possible solution interval is

Construction of the set of test candidates T

Given: A constraint $p \sim 0$, $p = ax^2 + bx + c$, $\sim \in \{=, <, >, \leq, \geq, \neq\}$.
Possible zeros of p (in x) are:

Linear in x :	$x_0 = -\frac{c}{b}$	, if	$a = 0 \wedge b \neq 0$
Quadratic in x :	$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac \geq 0$
	$x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac > 0$

The finite endpoints of possible solution intervals of $p \sim 0$ are the zeros of p (as the sign of p is invariant between its zeros).

Note: If p has no zeros then the possible solution interval is $(-\infty, \infty)$.

Construction of the set of test candidates T

Given: A constraint $p \sim 0$, $\quad p = ax^2 + bx + c$, $\quad \sim \in \{=, <, >, \leq, \geq, \neq\}$.
Possible zeros of p (in x) are:

Linear in x :	$x_0 = -\frac{c}{b}$	, if	$a = 0 \wedge b \neq 0$
Quadratic in x :	$x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac \geq 0$
	$x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$	, if	$a \neq 0 \wedge b^2 - 4ac > 0$

The finite endpoints of possible solution intervals of $p \sim 0$ are the zeros of p (as the sign of p is invariant between its zeros).

Note: If p has no zeros then the possible solution interval is $(-\infty, \infty)$.

Thus the possible solution intervals for x in $p \sim 0$ are:

constraints			possible solution intervals $(0 \leq i,\ j \leq 2,\ i \neq j)$				
$p = 0$				$[x_i,\ x_i]$			$(-\infty,\ \infty)$
$p < 0$	$p > 0$	$p \neq 0$	$(-\infty,\ x_i)$		$(x_i,\ x_j)$	$(x_i,\ \infty)$	$(-\infty,\ \infty)$
$p \leq 0$	$p \geq 0$		$(-\infty,\ x_i]$		$[x_i,\ x_j]$	$[x_i,\ \infty)$	$(-\infty,\ \infty)$

Construction of the set of test candidates T

constraints	possible solution intervals ($0 \leq i,\ j \leq 2,\ i \neq j$)				
$p = 0$		$[x_i,\ x_i]$			$(-\infty,\ \infty)$
$p < 0$ $p > 0$ $p \neq 0$	$(-\infty,\ x_i)$		$(x_i,\ x_j)$	$(x_i,\ \infty)$	$(-\infty,\ \infty)$
$p \leq 0$ $p \geq 0$	$(-\infty,\ x_i]$		$[x_i,\ x_j]$	$[x_i,\ \infty)$	$(-\infty,\ \infty)$

Construction of the set of test candidates T

constraints			possible solution intervals $(0 \leq i,\ j \leq 2,\ i \neq j)$				
$p = 0$				$[x_i,\ x_i]$			$(-\infty,\ \infty)$
$p < 0$	$p > 0$	$p \neq 0$	$(-\infty,\ x_i)$		$(x_i,\ x_j)$	$(x_i,\ \infty)$	$(-\infty,\ \infty)$
$p \leq 0$	$p \geq 0$		$(-\infty,\ x_i]$		$[x_i,\ x_j]$	$[x_i,\ \infty)$	$(-\infty,\ \infty)$

We need to pick one test candidate from each of those intervals.

Note: In general, the zeros x_i are not constants but might contain other variables. Especially, it implies that they cannot be ordered by their values.

Construction of the set of test candidates T

constraints	possible solution intervals ($0 \leq i,\ j \leq 2,\ i \neq j$)			
$p = 0$	$[x_i,\ x_i]$			$(-\infty,\ \infty)$
$p < 0$ $\quad p > 0$ $\quad p \neq 0$	$(-\infty,\ x_i)$	$(x_i,\ x_j)$	$(x_i,\ \infty)$	$(-\infty,\ \infty)$
$p \leq 0$ $\quad p \geq 0$	$(-\infty,\ x_i]$	$[x_i,\ x_j]$	$[x_i,\ \infty)$	$(-\infty,\ \infty)$

We need to pick one test candidate from each of those intervals.

Note: In general, the zeros x_i are not constants but might contain other variables. Especially, it implies that they cannot be ordered by their values.

As test candidates we take

Construction of the set of test candidates T

constraints	possible solution intervals $(0 \leq i,\ j \leq 2,\ i \neq j)$				
$p = 0$		$[x_i,\ x_i]$			$(-\infty,\ \infty)$
$p < 0 \quad p > 0 \quad p \neq 0$	$(-\infty,\ x_i)$		$(x_i,\ x_j)$	$(x_i,\ \infty)$	$(-\infty,\ \infty)$
$p \leq 0 \quad p \geq 0$	$(-\infty,\ x_i]$		$[x_i,\ x_j]$	$[x_i,\ \infty)$	$(-\infty,\ \infty)$

We need to pick one test candidate from each of those intervals.
Note: In general, the zeros x_i are not constants but might contain other variables. Especially, it implies that they cannot be ordered by their values.

As test candidates we take the smallest value from each of those possible solution intervals:

Construction of the set of test candidates T

constraints	possible solution intervals ($0 \leq i,\ j \leq 2,\ i \neq j$)				
$p = 0$		$[x_i,\ x_i]$			$(-\infty,\ \infty)$
$p < 0$ $\quad p > 0$ $\quad p \neq 0$	$(-\infty,\ x_i)$		$(x_i,\ x_j)$	$(x_i,\ \infty)$	$(-\infty,\ \infty)$
$p \leq 0$ $\quad p \geq 0$	$(-\infty,\ x_i]$		$[x_i,\ x_j]$	$[x_i,\ \infty)$	$(-\infty,\ \infty)$

We need to pick one test candidate from each of those intervals.
Note: In general, the zeros x_i are not constants but might contain other variables. Especially, it implies that they cannot be ordered by their values.

As test candidates we take the smallest value from each of those possible solution intervals:

- $p = 0,\ p \leq 0,\ p \geq 0$

constraints	possible solution intervals ($0 \leq i,\ j \leq 2,\ i \neq j$)			
$p = 0$	$[x_i,\ x_i]$			$(-\infty,\ \infty)$
$p < 0$ $\quad p > 0$ $\quad p \neq 0$	$(-\infty,\ x_i)$	$(x_i,\ x_j)$	$(x_i,\ \infty)$	$(-\infty,\ \infty)$
$p \leq 0$ $\quad p \geq 0$	$(-\infty,\ x_i]$	$[x_i,\ x_j]$	$[x_i,\ \infty)$	$(-\infty,\ \infty)$

We need to pick one test candidate from each of those intervals.
Note: In general, the zeros x_i are not constants but might contain other variables. Especially, it implies that they cannot be ordered by their values.

As test candidates we take the smallest value from each of those possible solution intervals:

- $p = 0,\ p \leq 0,\ p \geq 0$
  1. Zeros of the polynomial p
  2. $-\infty$ $\quad$ ($:=$ sufficiently small value)

Construction of the set of test candidates T

constraints	possible solution intervals ($0 \leq i,\ j \leq 2,\ i \neq j$)			
$p = 0$	$[x_i,\ x_i]$			$(-\infty,\ \infty)$
$p < 0$ $\quad p > 0$ $\quad p \neq 0$	$(-\infty,\ x_i)$	$(x_i,\ x_j)$	$(x_i,\ \infty)$	$(-\infty,\ \infty)$
$p \leq 0$ $\quad p \geq 0$	$(-\infty,\ x_i]$	$[x_i,\ x_j]$	$[x_i,\ \infty)$	$(-\infty,\ \infty)$

We need to pick one test candidate from each of those intervals.
Note: In general, the zeros x_i are not constants but might contain other variables. Especially, it implies that they cannot be ordered by their values.

As test candidates we take the smallest value from each of those possible solution intervals:

- $p = 0,\ p \leq 0,\ p \geq 0$
  1. Zeros of the polynomial p
  2. $-\infty$ $\quad$ (:= sufficiently small value)
- $p < 0,\ p > 0,\ p \neq 0$

Construction of the set of test candidates T

constraints	possible solution intervals ($0 \le i,\ j \le 2,\ i \ne j$)			
$p = 0$		$[x_i,\ x_i]$		$(-\infty,\ \infty)$
$p < 0$ $p > 0$ $p \ne 0$	$(-\infty,\ x_i)$	$(x_i,\ x_j)$	$(x_i,\ \infty)$	$(-\infty,\ \infty)$
$p \le 0$ $p \ge 0$	$(-\infty,\ x_i]$	$[x_i,\ x_j]$	$[x_i,\ \infty)$	$(-\infty,\ \infty)$

We need to pick one test candidate from each of those intervals.
Note: In general, the zeros x_i are not constants but might contain other variables. Especially, it implies that they cannot be ordered by their values.

As test candidates we take the smallest value from each of those possible solution intervals:

- $p = 0,\ p \le 0,\ p \ge 0$
  1. Zeros of the polynomial p
  2. $-\infty$ ($:=$ sufficiently small value)
- $p < 0,\ p > 0,\ p \ne 0$
  1. Zeros of the polynomial p plus an infinitesimal ϵ
  2. $-\infty$

Construction of the set of test candidates T

Example: $\exists y \exists x : \ (y = 0 \ \vee \ y^2 + 1 < 0) \ \wedge \ x - 3 \leq 0 \ \wedge \ xy + 1 < 0$

$$\overset{\text{eliminate x}}{\equiv}$$

Construction of the set of test candidates T

Example: $\exists y \exists x : \; (y = 0 \;\vee\; y^2 + 1 < 0) \;\wedge\; x - 3 \leq 0 \;\wedge\; xy + 1 < 0$

Test candidates

eliminate x

$\equiv$

$$\begin{aligned}
\exists y : \quad & (\,(y = 0 \;\vee\; y^2 + 1 < 0) \;\wedge\; x - 3 \leq 0 \;\wedge\; xy + 1 < 0\,)[-\infty /\!/ \mathbf{x}] \\
\vee \quad & (\,(y = 0 \;\vee\; y^2 + 1 < 0) \;\wedge\; x - 3 \leq 0 \;\wedge\; xy + 1 < 0\,)[\mathbf{3} /\!/ \mathbf{x}] \quad) \\
\vee \; (\; \mathbf{y} \neq \mathbf{0} \;\wedge\; & (\,(y = 0 \;\vee\; y^2 + 1 < 0) \;\wedge\; x - 3 \leq 0 \;\wedge\; xy + 1 < 0\,)[-\tfrac{\mathbf{1}}{\mathbf{y}} + \epsilon /\!/ \mathbf{x}] \quad)
\end{aligned}$$

Side condition

Construction of the set of test candidates T

Example: $\exists y \exists x : \ (y = 0 \ \vee \ y^2 + 1 < 0) \ \wedge \ x - 3 \leq 0 \ \wedge \ xy + 1 < 0$

Test candidates

$\overset{\text{eliminate x}}{\equiv}$

$$\begin{aligned} \exists y : \quad & (\ (y = 0 \ \vee \ y^2 + 1 < 0) \ \wedge \ x - 3 \leq 0 \ \wedge \ xy + 1 < 0\ )[-\infty /\!/ \mathbf{x}] \\ \vee \quad & (\ (y = 0 \ \vee \ y^2 + 1 < 0) \ \wedge \ x - 3 \leq 0 \ \wedge \ xy + 1 < 0\ )[\mathbf{3} /\!/ \mathbf{x}] \quad) \\ \vee \ (\ \mathbf{y \neq 0} \ \wedge \quad & (\ (y = 0 \ \vee \ y^2 + 1 < 0) \ \wedge \ x - 3 \leq 0 \ \wedge \ xy + 1 < 0\ )[-\tfrac{\mathbf{1}}{\mathbf{y}} + \epsilon /\!/ \mathbf{x}] \quad) \end{aligned}$$

Side condition

$\overset{\text{eliminate y}}{\equiv}$

...

Virtual substitution of a variable by a test candidate

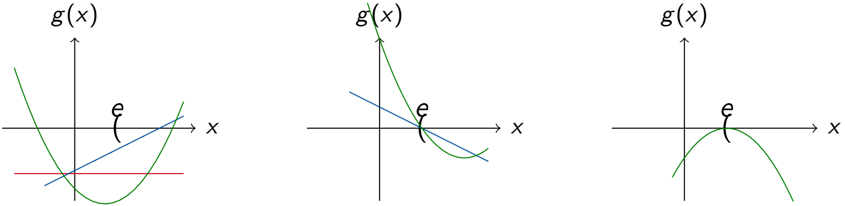
Example: $(g(x)=0)[\frac{q+r\sqrt{t}}{s}/\!/x]$

1. Substitute $\frac{q+r\sqrt{t}}{s}$ for x in $g(x)=0$ in the common way.
2. Transform the result to $\frac{\hat{q}+\hat{r}\sqrt{t}}{\hat{s}}=0$ where $\hat{q}$, $\hat{r}$, and $\hat{s}$ are polynomials.
3. $$\begin{aligned} & \frac{\hat{q}+\hat{r}\sqrt{t}}{\hat{s}}=0 && \Leftrightarrow \quad \hat{q}+\hat{r}\sqrt{t}=0 \\ \Leftrightarrow \quad & \hat{q}\hat{r}\leq 0 \ \wedge \ \|\hat{q}\|=\|\hat{r}\sqrt{t}\| && \Leftrightarrow \quad \hat{q}\hat{r}\leq 0 \wedge \hat{q}^2-\hat{r}^2 t=0 \end{aligned}$$

Result: $(g(x)=0)[\frac{q+r\sqrt{t}}{s}/\!/x] = (\hat{q}\hat{r}\leq 0 \wedge \hat{q}^2-\hat{r}^2 t = 0)$

Virtual substitution of a variable by a test candidate

Example: $(g(x) < 0)[e + \epsilon /\!/ x]$



Case 1 Case 2 Case 3

Result:

$$\underbrace{g[e/\!/x] < 0}_{\text{Case 1}} \vee \underbrace{g[e/\!/x] = 0 \wedge g'[e/\!/x] < 0}_{\text{Case 2}} \vee \underbrace{g[e/\!/x] = 0 \wedge g'[e/\!/x] = 0 \wedge g''[e/\!/x] < 0}_{\text{Case 3}}$$

Cylindrical algebraic decomposition: Idea

- Assume a set P of polynomials in n variables together with a sign condition for each polynomial in P.
- The cylindrical algebraic decomposition (CAD) method produces a decomposition of $\mathbb{R}^n$ into a finite number of P-sign-invariant regions (CAD cells).
- Take an arbitrary element (sample point) from each of the CAD cells.
- If all sign conditions are satisfied for at least one sample point then the problem is satisfiable.
- Otherwise the problem is unsatisfiable.

Delineability

Let $R \subseteq \mathbb{R}^{n-1}$ be a region and $P = \{p_1, \ldots, p_m\} \subset \mathbb{Z}[x_1, \ldots, x_n]$, where $m \geq 1$ and $n \geq 2$.

Intuition: If P is delineable on R then the real roots of P vary continuously over R, while maintaining their number and order.

Definition

P is delineable on R if for $1 \leq i, j \leq m$ with $i \neq j$ and for all $a \in R$:

1. the number of roots of $p_i(a)$ is constant,
2. the number of different roots of $p_i(a)$ is constant,
3. the number of common roots of $p_i(a)$ and $p_j(a)$ is constant.

Cylindrical algebraic decomposition

Let $P = (p_1, \ldots, p_m) \in \mathbb{Z}[x_1, \ldots, x_n]^m$ and $\mathcal{C} \subseteq 2^{\mathbb{R}^n}$ finite with $m, n \geq 1$.

Definition

$\mathcal{C}$ is called cylindrical algebraic decomposition (CAD) of $\mathbb{R}^n$ for P if the following holds:

1. $\bigcup \mathcal{C} = \mathbb{R}^n$,
2. $C \cap C' = \emptyset$ for all $C, C' \in \mathcal{C}$ with $C \neq C'$,
3. If $n = 1$, then every $C \in \mathcal{C}$ is a P-sign invariant region.
4. If $n > 1$ then there exists a CAD $\mathcal{C}'$ of $\mathbb{R}^{n-1}$ such that for every $C \in \mathcal{C}$ there is a $C' \in \mathcal{C}'$ such that the projection of C to the first $n-1$ dimensions is C'.

An element $C \in \mathcal{C}$ is called a cell.

Cylindrical algebraic decomposition

Let $P = (p_1, \ldots, p_m) \in \mathbb{Z}[x_1, \ldots, x_n]^m$ and $\mathcal{C} \subseteq 2^{\mathbb{R}^n}$ finite with $m, n \geq 1$.

Definition

$\mathcal{C}$ is called cylindrical algebraic decomposition (CAD) of $\mathbb{R}^n$ for P if the following holds:

1. $\bigcup \mathcal{C} = \mathbb{R}^n$,
2. $C \cap C' = \emptyset$ for all $C, C' \in \mathcal{C}$ with $C \neq C'$,
3. If $n = 1$, then every $C \in \mathcal{C}$ is a P-sign invariant region.
4. If $n > 1$ then there exists a CAD $\mathcal{C}'$ of $\mathbb{R}^{n-1}$ such that for every $C \in \mathcal{C}$ there is a $C' \in \mathcal{C}'$ such that the projection of C to the first $n-1$ dimensions is C'.

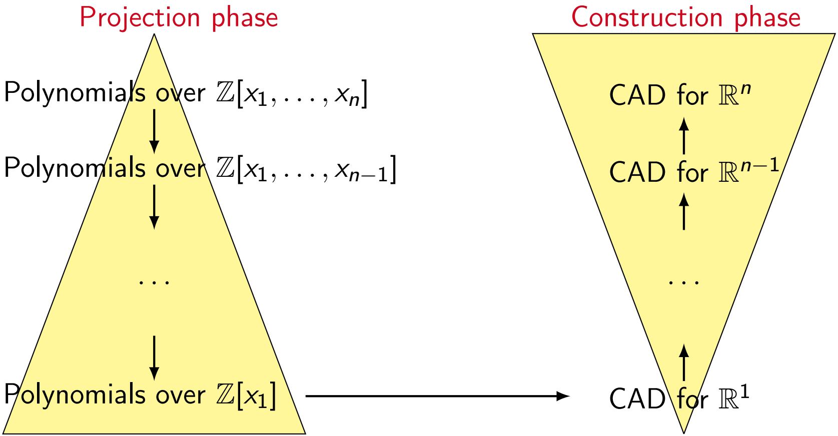
An element $C \in \mathcal{C}$ is called a cell.

Remark

One sample point per cell is sufficient in order to represent a CAD.

CAD for $\mathbb{R}^n$

A CAD for a set of polynomials from $\mathbb{Z}[x_1, \ldots, x_n]$ splits $\mathbb{R}^n$ into sign-invariant regions.



CAD projection

Let $P = \{p_1, \ldots, p_m\} \in \mathbb{Z}[x_1, \ldots, x_n]$ where $n \geq 2$ and $m \geq 1$.

Definition

A mapping

$$\mathrm{proj} : 2^{\mathbb{Z}[x_1,\ldots,x_n]} \longrightarrow 2^{\mathbb{Z}[x_1,\ldots,x_{n-1}]}$$

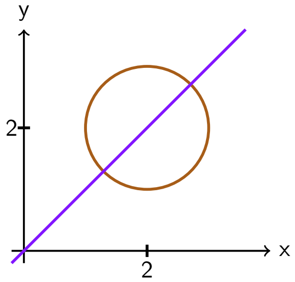
is called a CAD-Projection, if any region $R \subseteq \mathbb{R}^{n-1}$ is $\mathrm{proj}(P)$-sign invariant *iff* R is P-delineable.

Remark

- Usually, $|\mathrm{proj}(P)| = |P|^2$. Thus, projecting recursively up to the univariate case is in $\mathcal{O}(|P|^{2^{n-1}})$.

Example: CAD projection

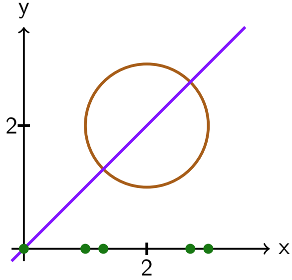
$$P = \begin{pmatrix} (x-2)^2 + \\ (y-2)^2 - 1, \\ \\ x - y \end{pmatrix}$$



$$\begin{aligned} \mathrm{proj}(P) \quad = \quad \{ & x^2 - 4x + 3, \\ & -4x + x^2 + \tfrac{7}{2}, \\ & x^4 - 8x^3 + 30x^2 - 56x + 49, \\ & x^2 - 4x + 7, \\ & x \} \end{aligned}$$

Example: CAD projection

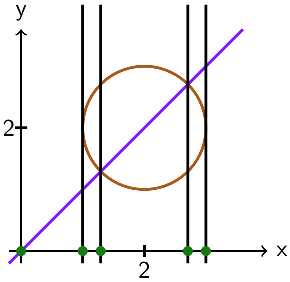
$$P = \begin{pmatrix} (x-2)^2 + \\ (y-2)^2 - 1, \\ \\ x - y \end{pmatrix}$$



$$\begin{aligned} \mathrm{proj}(P) \quad = \quad & \{x^2 - 4x + 3, & \{1,\ 3\} \\ & -4x + x^2 + \tfrac{7}{2}, & \{2 - \tfrac{\sqrt{2}}{2},\ 2 + \tfrac{\sqrt{2}}{2}\} \\ & x^4 - 8x^3 + 30x^2 - 56x + 49, & \{\} \\ & x^2 - 4x + 7, & \{\} \\ & x\} & \{0\} \end{aligned}$$

Example: CAD projection

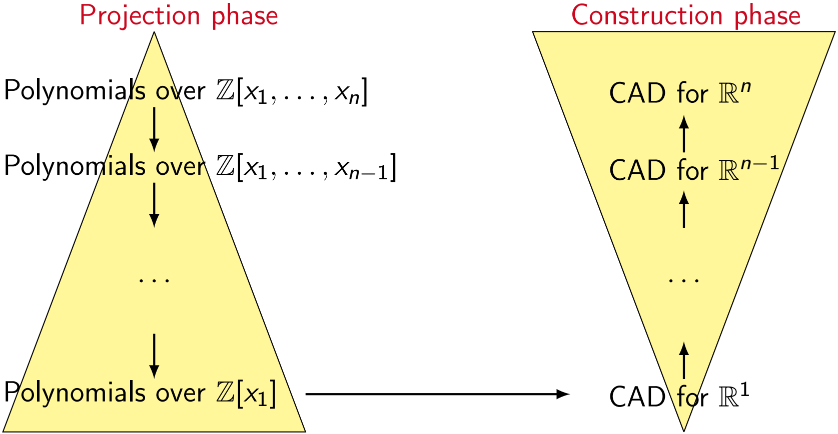
$$P = \begin{pmatrix} (x-2)^2 + \\ (y-2)^2 - 1, \\ \\ x - y \end{pmatrix}$$



$$\begin{aligned} \text{proj}(P) \quad = \quad \{ & x^2 - 4x + 3, & \{1,\ 3\} \\ & -4x + x^2 + \tfrac{7}{2}, & \{2 - \tfrac{\sqrt{2}}{2},\ 2 + \tfrac{\sqrt{2}}{2}\} \\ & x^4 - 8x^3 + 30x^2 - 56x + 49, & \{\} \\ & x^2 - 4x + 7, & \{\} \\ & x\} & \{0\} \end{aligned}$$

CAD for $\mathbb{R}^n$

A CAD for a set of polynomials from $\mathbb{Z}[x_1, \ldots, x_n]$
splits $\mathbb{R}^n$ into sign-invariant regions.

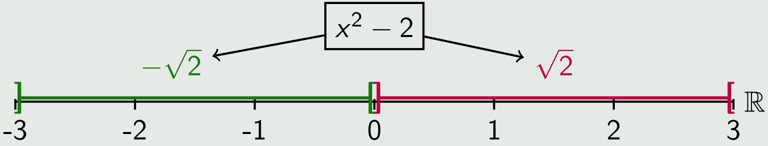


Representing real roots (real algebraic numbers)

Interval representation

$$\Big(\ \underbrace{p,}_{\in \mathbb{Z}[x]}\quad \underbrace{(\ l,\quad r\ )}_{\text{exactly one real root of } p \text{ in } (l,r)}\ \Big)$$

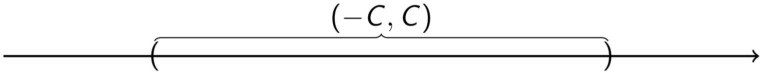
Example



Real root isolation in $\mathbb{R}$

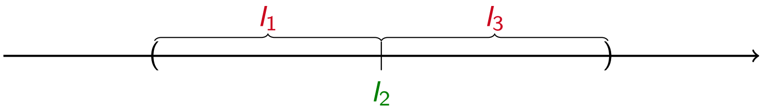
Real root isolation in $\mathbb{R}$

- Assume a set $P = \{p_1 \sim_1 0, \ldots, p_k \sim_k\}$ of univariate polynomial constraints with $p_i \in \mathbb{Z}[x]$ and $\sim_i \in \{<, \leq, =, \neq, \geq, >\}$.
- Cauchy bound $\Rightarrow$ Interval $(-C, C)$ containing all real roots of $p_1, \ldots, p_k$.
- Sturm sequence $\Rightarrow$ count the real roots of each p_i in an interval.
- Split C until each sub-interval I contains at most one real root.



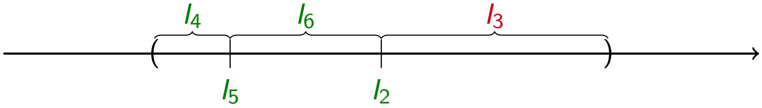
Real root isolation in $\mathbb{R}$

- Assume a set $P = \{p_1 \sim_1 0, \ldots, p_k \sim_k\}$ of univariate polynomial constraints with $p_i \in \mathbb{Z}[x]$ and $\sim_i \in \{<, \leq, =, \neq, \geq, >\}$.
- Cauchy bound $\Rightarrow$ Interval $(-C, C)$ containing all real roots of $p_1, \ldots, p_k$.
- Sturm sequence $\Rightarrow$ count the real roots of each p_i in an interval.
- Split C until each sub-interval I contains at most one real root.



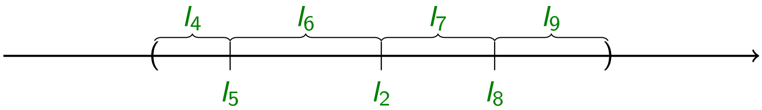
Real root isolation in $\mathbb{R}$

- Assume a set $P = \{p_1 \sim_1 0, \ldots, p_k \sim_k\}$ of univariate polynomial constraints with $p_i \in \mathbb{Z}[x]$ and $\sim_i \in \{<, \leq, =, \neq, \geq, >\}$.
- Cauchy bound $\Rightarrow$ Interval $(-C, C)$ containing all real roots of $p_1, \ldots, p_k$.
- Sturm sequence $\Rightarrow$ count the real roots of each p_i in an interval.
- Split C until each sub-interval I contains at most one real root.



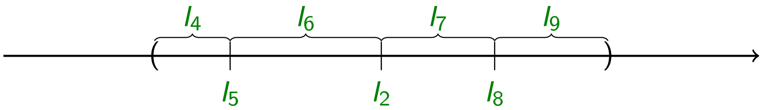
Real root isolation in $\mathbb{R}$

- Assume a set $P = \{p_1 \sim_1 0, \ldots, p_k \sim_k\}$ of univariate polynomial constraints with $p_i \in \mathbb{Z}[x]$ and $\sim_i \in \{<, \leq, =, \neq, \geq, >\}$.
- Cauchy bound $\Rightarrow$ Interval $(-C, C)$ containing all real roots of $p_1, \ldots, p_k$.
- Sturm sequence $\Rightarrow$ count the real roots of each p_i in an interval.
- Split C until each sub-interval I contains at most one real root.



Real root isolation in $\mathbb{R}$

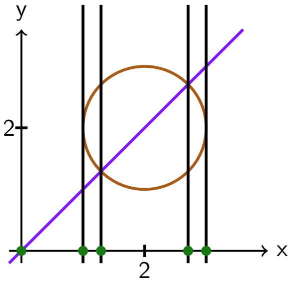
- Assume a set $P = \{p_1 \sim_1 0, \ldots, p_k \sim_k\}$ of univariate polynomial constraints with $p_i \in \mathbb{Z}[x]$ and $\sim_i \in \{<, \leq, =, \neq, \geq, >\}$.
- Cauchy bound $\Rightarrow$ Interval $(-C, C)$ containing all real roots of $p_1, \ldots, p_k$.
- Sturm sequence $\Rightarrow$ count the real roots of each p_i in an interval.
- Split C until each sub-interval I contains at most one real root.



CAD for $\mathbb{R}$ with respect to $p_1, \ldots, p_k$:
$[(p_i, I_j), (p_i, I_j)]$ for each I_j containing a real root of a p_i and open intervals between them.

Example: CAD projection

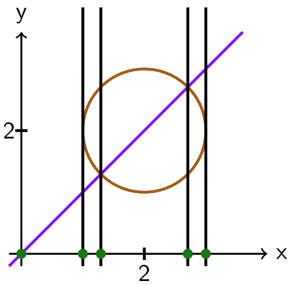
$$P = \begin{pmatrix} (x-2)^2 + \\ (y-2)^2 - 1, \\ \\ x - y \end{pmatrix}$$



$$\begin{aligned} \mathrm{proj}(P) \quad = \quad \{ & x^2 - 4x + 3, & \{1,\ 3\} \\ & -4x + x^2 + \tfrac{7}{2}, & \{2 - \tfrac{\sqrt{2}}{2},\ 2 + \tfrac{\sqrt{2}}{2}\} \\ & x^4 - 8x^3 + 30x^2 - 56x + 49, & \{\} \\ & x^2 - 4x + 7, & \{\} \\ & x\} & \{0\} \end{aligned}$$

Example: CAD projection

$$P = \begin{pmatrix} (x-2)^2 + \\ (y-2)^2 - 1, \\ \\ x - y \end{pmatrix}$$



$$\begin{array}{rcll} \mathrm{proj}(P) & = & \{x^2 - 4x + 3, & \{1,\ 3\} \\ & & -4x + x^2 + \frac{7}{2}, & \{2 - \frac{\sqrt{2}}{2},\ 2 + \frac{\sqrt{2}}{2}\} \\ & & x^4 - 8x^3 + 30x^2 - 56x + 49, & \{\} \\ & & x^2 - 4x + 7, & \{\} \\ & & x\} & \{0\} \end{array}$$

1-dimensional CAD (dimension x):

$$\underbrace{(-\infty, 1), \{1\}, (1, 2-\frac{\sqrt{2}}{2}), \{2-\frac{\sqrt{2}}{2}\}, (2-\frac{\sqrt{2}}{2}, 2+\frac{\sqrt{2}}{2}), \{2+\frac{\sqrt{2}}{2}\}, (2+\frac{\sqrt{2}}{2}, 3), \{3\}, (3, \infty)}_{samples \rightarrow Z_1}$$

The CAD sample construction in a nutshell

$\rightarrow P_n \subseteq \mathbb{Z}[x_1, \ldots, x_n]$

eliminate x_n $\downarrow$

$P_{n-1} \subseteq \mathbb{Z}[x_1, \ldots, x_{n-1}]$

eliminate x_{n-1} $\downarrow$

$\vdots$

eliminate x_2 $\downarrow$

$P_1 \subseteq \mathbb{Z}[x_1]$

The CAD sample construction in a nutshell

$\rightarrow P_n \subseteq \mathbb{Z}[x_1, \ldots, x_n]$

eliminate x_n $\downarrow$

$P_{n-1} \subseteq \mathbb{Z}[x_1, \ldots, x_{n-1}]$

eliminate x_{n-1} $\downarrow$

$\vdots$

eliminate x_2 $\downarrow$

$P_1 \subseteq \mathbb{Z}[x_1]$

roots of P_1 $\rightarrow$

$Z_1 \subseteq \mathbb{R}$

$\uparrow$ roots of $\{p(z) \mid p \in P_2, z \in Z_1\}$

$Z_1 \times Z_2 \subseteq \mathbb{R}^2$

$\uparrow$ roots of $\{p(z) \mid p \in P_3, z \in Z_1 \times Z_2\}$

$\vdots$

$\uparrow$ roots of $\{p(z) | p \in P_n, z \in Z_1 \times \cdots \times Z_{n-1}\}$

$Z_1 \times \cdots \times Z_n \subseteq \mathbb{R}^n$

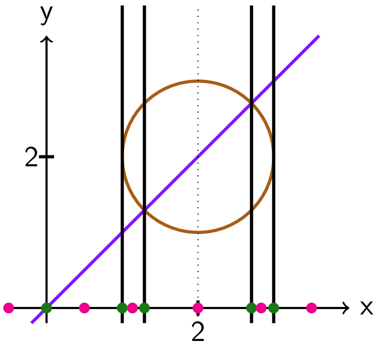
Example: CAD sample construction

$$P = \begin{pmatrix} (x-2)^2 + \\ (y-2)^2 - 1, \\ \\ x - y \end{pmatrix}$$

Samples for $\mathrm{proj}(P)$:

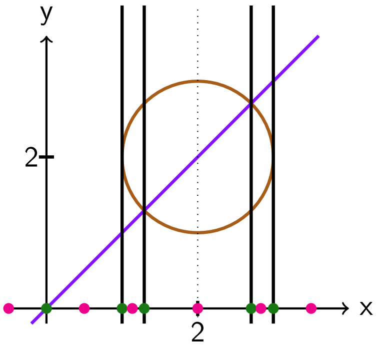
$\{0, 1, 2 - \frac{\sqrt{2}}{2},\ 2 + \frac{\sqrt{2}}{2},\ 3\}$

$\{-0.5,\ 0.5,\ 1.135,$
$2,\ 2.835,\ 3.5\}$



Example: CAD sample construction

$$P = \begin{pmatrix} (x-2)^2 + \\ (y-2)^2 - 1, \\ \\ x - y \end{pmatrix}$$



Samples for $\mathrm{proj}(P)$:

$\{0, 1, 2 - \frac{\sqrt{2}}{2}, \ 2 + \frac{\sqrt{2}}{2}, \ 3\}$

$\{-0.5, \ 0.5, \ 1.135,$
$2, \ 2.835, \ 3.5\}$

Example sample constructions

- $(2-2)^2 + (y-2)^2 - 1$ has zeros 1 and 3.
- $2 - y$ has zero 2.
- Two-dimensional samples are $(2, s)$, one s taken from the each of $(-\infty, 1), \{1\}, (1, 2), \{2\}, (2, 3), \{3\}, (3, \infty)$.