Decidability of Real arithmetic

Quantifier elimination in Real arithmetic

Upamanyu and Mohith

Is real arithmetic decidable?

Syntax of real arithmetic

- $FO(\mathbb{R}, +, -, \cdot, <, 0, 1)$, with the usual interpretations of $+, -, \cdot, <, 0, 1$.
- More formally:

  $(Term)\ t ::= x \in V \mid 0 \mid 1 \mid t_1 + t_2 \mid t_1 - t_2 \mid t_1 \cdot t_2$
  $(Atom)\ A ::= t_1 < t_2 \mid t_1 = t_2$
  $(Formula)\ \varphi ::= A \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x\varphi \mid \forall x\varphi$

Is real arithmetic decidable?

Syntax of real arithmetic

- $FO(\mathbb{R}, +, -, \cdot, <, 0, 1)$, with the usual interpretations of $+, -, \cdot, <, 0, 1$.
- More formally:

  $(Term)\ t ::= x \in V \mid 0 \mid 1 \mid t_1 + t_2 \mid t_1 - t_2 \mid t_1 \cdot t_2$
  $(Atom)\ A ::= t_1 < t_2 \mid t_1 = t_2$
  $(Formula)\ \varphi ::= A \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists x\varphi \mid \forall x\varphi$

Decidability

A theory is decidable if there is an algorithm that can decide in finite time whether a given formula is valid in the theory.

Tarski-Seidenberg theorem

Tarski-Seidenberg theorem

The first-order theory of reals admits quantifier elimination.

- Quantifier-free real arithmetic is decidable.
- Thus, it is sufficient to provide a proof of the above theorem. We will give a constructive proof, originally by Hörmander (1983).

Example

$\exists x : ax + b = 0$
becomes $(a = 0) \Rightarrow (b = 0)$ on quantifier elimination.

Reduction of formula to required form

All terms are polynomials

- If our formula has no variables, then we are done.
- Otherwise, each term has k variables. For any one variable x, it can be considered as a polynomial in x, with coefficients being polynomials in the remaining $k-1$ variables.
- So each atom is of the form $p(x) \bowtie 0$, where $\bowtie$ is one of $\leq, <, >, \geq, =, \neq$.

Lemma

It is sufficient to eliminate $\exists$ from a formula of the form

$$\exists x : p_1(x) \bowtie_1 0 \wedge \cdots \wedge p_n(x) \bowtie_n 0\,,$$

where $\bowtie_i$ is one of $\leq, <, >, \geq, =, \neq$.

Reduction of formula to required form

Proof of lemma

Suppose we can eliminate quantifiers from formulae of the given form. Then for any given formula with $n+1$ quantifiers,

- Consider the innermost quantifier along with the subformula φ in its body.
- If the quantifier is $\forall$, rewrite $\forall x \varphi(x)$ as $\neg(\exists x \neg\varphi(x))$.

Reduction of formula to required form

Proof of lemma

Suppose we can eliminate quantifiers from formulae of the given form. Then for any given formula with $n+1$ quantifiers,

- Consider the innermost quantifier along with the subformula φ in its body.
- If the quantifier is $\forall$, rewrite $\forall x \varphi(x)$ as $\neg(\exists x \neg\varphi(x))$.
- Convert the subformula to DNF (i.e. of the form $\varphi_1(x) \vee \cdots \vee \varphi_m(x)$, where each φ_i is a conjunction of atoms.)
- Distribute $\exists$ over the φ_i s, to get $(\exists x \varphi_1(x)) \vee \cdots \vee (\exists x \varphi_m(x))$.

Reduction of formula to required form

Proof of lemma

Suppose we can eliminate quantifiers from formulae of the given form. Then for any given formula with $n+1$ quantifiers,

- Consider the innermost quantifier along with the subformula φ in its body.
- If the quantifier is $\forall$, rewrite $\forall x \varphi(x)$ as $\neg(\exists x \neg\varphi(x))$.
- Convert the subformula to DNF (i.e. of the form $\varphi_1(x) \vee \cdots \vee \varphi_m(x)$, where each φ_i is a conjunction of atoms.)
- Distribute $\exists$ over the φ_i s, to get $(\exists x \varphi_1(x)) \vee \cdots \vee (\exists x \varphi_m(x))$.
- Eliminate quantifiers from each of these m subformulas, to get a formula with n quantifiers.

By induction, we are done.

Division algorithm (univariate)

Division

Given two polynomials $p(x)$ and $q(x)$ (with q nonzero), containing no other variables, we can divide p by q, to get polynomials s and r satisfying the equation

$$p(x) = s(x)q(x) + r(x)\,,$$

with $\deg(r) < \deg(q)$.

Sign matrix (univariate)

Sign matrix

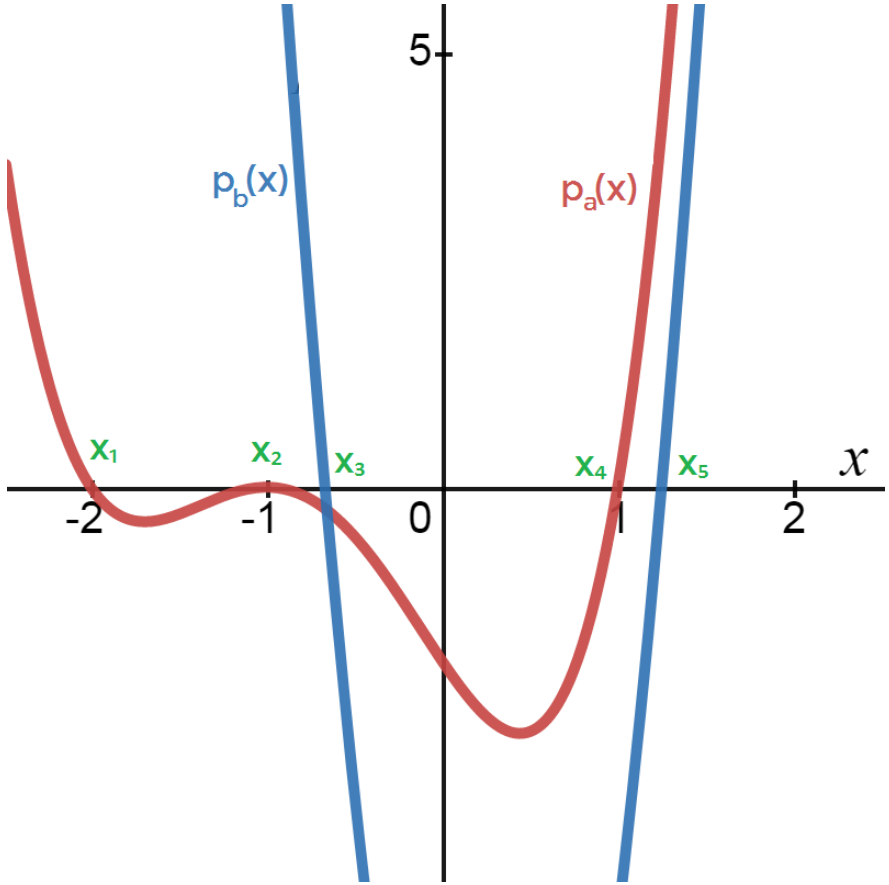
Given a set of polynomials $\{p_1(x), \ldots, p_n(x)\}$ in one variable x, with $x_1 < x_2 < \cdots < x_m$ being the list of points which are real roots of atleast one p_i, the reduced sign matrix M of these polynomials is as follows:

- Rows are indexed by p_i s.
- Columns indices are:
  $(-\infty, x_1), x_1, (x_1, x_2), x_2, \ldots, (x_{m-1}, x_m), x_m, (x_m, \infty)$.
- On the column indexed by the interval or point j,

$$M_{ij} = \begin{cases} + & \text{if } p_i(x) > 0 \text{ on j} \\ 0 & \text{if } p_i(x) = 0 \text{ on j} \\ - & \text{if } p_i(x) < 0 \text{ on j}\,. \end{cases}$$

Sign matrix example

$$p_a = x^4 + 3x^3 + x^2 - 3x - 2$$
$$p_b = 12x^2 - 7x - 10$$



		x_1		x_2		x_3		x_4		x_5	
p_a	$+$	0	$-$	0	$-$	$-$	$-$	0	$+$	$+$	$+$
p_b	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	0	$+$

Sign matrix to quantifier-free formula (univariate)

Quantifier-free formula from sign matrix

The sign matrix for a set of univariate polynomials $p_1(x), \ldots, p_n(x)$ is sufficient to convert any formula of the below form into a quantifier-free formula

$$\exists x : p_1(x) \bowtie_1 0 \ \wedge \cdots \wedge \ p_n(x) \bowtie_n 0$$

where $\bowtie_i$ can be one of $\leq, <, >, \geq, =, \neq$

Example:

$\exists x : p_a(x) < 0 \wedge p_b(x) = 0$
where
$p_a = x^4 + 3x^3 + x^2 - 3x - 2$ and
$p_b = 12x^2 - 7x - 10$

Sign matrix to quantifier-free formula (univariate)

		x_1		x_2		x_3		x_4		x_5	
p_a	$+$	0	$-$	0	$-$	$-$	$-$	0	$+$	$+$	$+$
p_b	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	0	$+$

Original formula: $\exists x : p_a(x) < 0 \wedge p_b(x) = 0$

New formula:

$(+1 = -1 \wedge +1 = 0) \quad \vee$

....

....

....

	x_1		x_2		x_3		x_4		x_5	
p_a	$+$	0	$-$	0	$-$	$-$	$-$	0	$+$	$+$	$+$
p_b	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	0	$+$

Original formula: $\exists x : p_a(x) < 0 \wedge p_b(x) = 0$

New formula:

$$
\begin{aligned}
&(+1 = -1 \wedge +1 = 0) \vee \\
&\quad (0 = -1 \wedge +1 = 0) \vee \\
&(-1 = -1 \wedge +1 = 0) \vee \\
&\quad (0 = -1 \wedge +1 = 0) \vee \\
&(-1 = -1 \wedge +1 = 0) \vee \\
&(-1 = -1 \wedge 0 = 0) \vee \\
&(-1 = -1 \wedge -1 = 0) \vee \\
&\quad (0 = -1 \wedge -1 = 0) \vee \\
&(+1 = -1 \wedge -1 = 0) \vee \\
&(+1 = -1 \wedge 0 = 0) \vee \\
&(+1 = -1 \wedge +1 = 0) \vee
\end{aligned}
$$

Sign matrix to quantifier-free formula (univariate)

		x_1		x_2		x_3		x_4		x_5	
p_a	$+$	0	$-$	0	$-$	$-$	$-$	0	$+$	$+$	$+$
p_b	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	0	$+$

Another example:

if we instead had

$$\exists x : p_a(x) \geq 0 \wedge p_b(x) < 0$$

the quantifier-free formula would be:

$$\begin{aligned}
&((+1 = 0 \vee +1 \;\; = +1) \wedge +1 \;\; = -1) \;\; \vee \\
&\quad ((0 = 0 \vee 0 \;\; = +1) \wedge +1 \;\; = -1) \;\; \vee \\
&((-1 = 0 \vee -1 \;\; = +1) \wedge +1 \;\; = -1) \;\; \vee \\
&\ldots. \\
&\ldots.
\end{aligned}$$

Building the Sign Matrix

Algorithm: Build from 'smaller' one

Given $p, p_1, p_2, \ldots, p_n$, this algorithm constructs its sign matrix from the sign matrix of $p', p_1, \ldots, p_n, r_0, r_1, \ldots, r_n$
where p' is the derivative of p and r_i is the remainder obtained when p is divided by p_i, in other words $p(x) = s(x) \cdot p_i(x) + r_i(x)$.
(Note: r_0 is the remainder when p is divided by p'. Think of p' as p_0)

Example:

$p = x^4 + 3x^3 + x^2 - 3x - 2$ and
$p_1 = 12x^2 - 7x - 10$

Building the Sign Matrix from a 'smaller' one

$$p = x^4 + 3x^3 + x^2 - 3x - 2 \qquad p' = 4x^3 + 9x^2 + 2x - 3$$

$$p_1 = 12x^2 - 7x - 10 \qquad r_0 = -\frac{19}{16}x^2 - \frac{-21}{8}x - \frac{23}{16}$$

$$r_1 = \frac{3931}{1728}x + \frac{1097}{864}$$

	x_1		x_2		x_3		x_4		x_5		x_6		x_7	
p'	−	0	+	+	+	0	−	−	−	−	−	0	+	+	+
p_1	+	+	+	+	+	+	+	0	−	−	−	−	−	0	+
r_0	−	−	−	0	+	0	−	−	−	−	−	−	−	−	−
r_1	−	−	−	−	−	−	−	−	−	0	+	+	+	+	+

Building the Sign Matrix from a 'smaller' one

$$p = x^4 + 3x^3 + x^2 - 3x - 2 \qquad p' = 4x^3 + 9x^2 + 2x - 3$$

$$p_1 = 12x^2 - 7x - 10 \qquad r_0 = -\frac{19}{16}x^2 - \frac{-21}{8}x - \frac{23}{16}$$

$$r_1 = \frac{3931}{1728}x + \frac{1097}{864}$$

	x_1		x_2		x_3		x_4		x_5		x_6		x_7	
p'	−	0	+	+	+	0	−	−	−	−	−	0	+	+	+
p_1	+	+	+	+	+	+	+	0	−	−	−	−	−	0	+
r_0	−	−	−	0	+	0	−	−	−	−	−	−	−	−	−
r_1	−	−	−	−	−	−	−	−	−	0	+	+	+	+	+

Split the sign matrix into two equally sized parts, one for the $p, p_1, \ldots, p_n$ and one for the $r_0, r_1, \ldots, r_n$.

Building the Sign Matrix from a 'smaller' one

- Split the sign matrix into two equally sized parts, one for the $p, p_1, \ldots, p_n$ and one for the $r_0, r_1, \ldots, r_n$
- Add a new row for p

	x_1			x_2			x_3			x_4			x_5			x_6			x_7		
p'	−	0	+	+	+	0	−	−	−	−	−	0	+	+	+						
p_1	+	+	+	+	+	+	+	0	−	−	−	−	−	0	+						
p																					

	x_1			x_2			x_3			x_4			x_5			x_6			x_7		
r_0	−	−	−	0	+	0	−	−	−	−	−	−	−	−	−						
r_1	−	−	−	−	−	−	−	−	−	0	+	+	+	+	+						

Building the Sign Matrix from a 'smaller' one

- We can now infer the sign of $p(x_i)$ for each point x_i that is a root of one of the polynomials p_k, as follows:

  Recall, $p(x) = s_k(x) \cdot p_k(x) + r_k(x)$.
  Thus, if $p_k(x_i) = 0$, then $p(x_i) = 0 + r_k(x_i)$.
  $\text{sign}(p(x_i)) = \text{sign}(r_k(x_i))$

	x_1			x_2			x_3			x_4			x_5			x_6			x_7		
p'	−	**0**	+	+	+	**0**	−	−	−	−	−	**0**	+	+	+						
p_1	+	+	+	+	+	+	+	**0**	−	−	−	−	−	**0**	+						
p		−				0		−				−		+							

	x_1			x_2			x_3			x_4			x_5			x_6			x_7		
r_0	−	−	−	0	+	0	−	−	−	−	−	−	−	−								
r_1	−	−	−	−	−	−	−	−	−	0	+	+	+	+	+							

Building the Sign Matrix from a 'smaller' one

- Throw away second sign matrix
- 'Condense' the first matrix by removing the points that are not roots of one of $p', p_1, \ldots, p_n$

	x_1		x_2		x_3		x_4		x_5		x_6		x_7	
p'	$-$	0	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$				0		$-$				$-$		$+$	

Building the Sign Matrix from a 'smaller' one

- Throw away second sign matrix
- 'Condense' the first matrix by removing the points that are not roots of one of $p', p_1, \ldots, p_n$

	x_1		x_2		x_3		x_4		x_5		x_6		x_7	
p'	$-$	0	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$				0		$-$				$-$		$+$	

	x_1		x_2		x_3		x_4		x_5	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$		$+$	

Building the Sign Matrix from a 'smaller' one

- If $p(x_i) \cdot p(x_{i+1}) < 0$, by Intermediate Value theorem, p has a root in the interval (x_i, x_{i+1}).

  Replace column labeled by (x_i, x_{i+1}) with 3 copies of itself, where the middle column is labeled by the root of p

		x_1		x_2		x_3		x_4		x_5	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$		$+$	

		x_1		x_2		x_3		x_4		x_{new}			x_5	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$					$+$	

Building the Sign Matrix from a 'smaller' one

- If $p(x_i) \cdot p(x_{i+1}) < 0$, by Intermediate Value theorem, p has a root in the interval (x_i, x_{i+1}).

  Replace column labeled by (x_i, x_{i+1}) with 3 copies of itself, where the middle column is labeled by the root of p

		x_1		x_2		x_3		x_4		x_5	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$		$+$	

		x_1		x_2		x_3		x_4		x_{new}		x_5	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$		0		$+$	

Building the Sign Matrix from a 'smaller' one

Pause and ponder

- Can p have two roots in (x_i, x_{i+1}) ?

		x_1		x_2		x_3		x_4		x_5		x_6	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$		0		$+$	

Building the Sign Matrix from a 'smaller' one

Pause and ponder

- Can p have two roots in (x_i, x_{i+1}) ?
- Can $p(x_i) = p(x_{i+1}) = 0$?

		x_1		x_2		x_3		x_4		x_5		x_6	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$		0		$+$	

Building the Sign Matrix from a 'smaller' one

Pause and ponder

- Can p have two roots in (x_i, x_{i+1}) ?
- Can $p(x_i) = p(x_{i+1}) = 0$?
- Given $p(x_i) > 0$ and $p(x_{i+1}) = 0$, can p have a root in (x_i, x_{i+1}) ?

	x_1		x_2		x_3		x_4		x_5		x_6	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$		0		$+$	

Building the Sign Matrix from a 'smaller' one

Pause and ponder

- Can p have two roots in (x_i, x_{i+1}) ?
- Can $p(x_i) = p(x_{i+1}) = 0$?
- Given $p(x_i) > 0$ and $p(x_{i+1}) = 0$, can p have a root in (x_i, x_{i+1}) ?
- Given $p(x_i) < 0$ and $p(x_{i+1}) = 0$, can p have a root in (x_i, x_{i+1}) ?

	x_1		x_2		x_3		x_4		x_5		x_6	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$		0		$+$	

Building the Sign Matrix from a 'smaller' one

- If $p(x_i) \geq 0$ and $p(x_{i+1}) \geq 0$, then add sign of p on (x_i, x_{i+1}) as $+$

	x_1		x_2		x_3		x_4		x_5		x_6	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$		0		$-$		$-$		0	$+$	$+$	

Building the Sign Matrix from a 'smaller' one

- If $p(x_i) \geq 0$ and $p(x_{i+1}) \geq 0$, then add sign of p on (x_i, x_{i+1}) as $+$
- If $p(x_i) \leq 0$ and $p(x_{i+1}) \leq 0$, then add sign of p on (x_i, x_{i+1}) as $-$

	x_1		x_2		x_3		x_4		x_5		x_6	
p'	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$
p		$-$	$-$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$	

Building the Sign Matrix from a 'smaller' one

- Add a new row at either ends of the table and label them as $-\infty$ and $+\infty$

	$-\infty$		x_1		x_2		x_3		x_4		x_5		x_6		$+\infty$
p'	$-$	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$
p			$-$	$-$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$		

Building the Sign Matrix from a 'smaller' one

- Add a new row at either ends of the table and label them as $-\infty$ and $+\infty$
- Infer the sign of p at $-\infty$ and $+\infty$ by looking at the degree of p and the sign of leading coefficient of p

Recall,

$$p = x^4 + 3x^3 + x^2 - 3x - 2$$

	$-\infty$		x_1		x_2		x_3		x_4		x_5		x_6		$+\infty$
p'	$-$	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$
p	$+$		$-$	$-$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$		$+$

Building the Sign Matrix from a 'smaller' one

- Fill in the 2 intervals, $(-\infty, x_1), (x_n, +\infty)$ as before.

	$-\infty$		x_{new}		x_1		x_2		x_3		x_4		x_5		x_6		$+\infty$
p'	$-$	$-$	$-$	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$	$+$
p_1	$+$	$+$	$+$	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$
p	$+$	$+$	0	$-$	$-$	$-$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$

Building the Sign Matrix from a 'smaller' one

- Fill in the 2 intervals, $(-\infty, x_1), (x_n, +\infty)$ as before.
- Throw away the first and last column

		x_1			x_2			x_3			x_4			x_5			x_6			x_7	
p'	$-$	$-$	$-$	0	$+$	0	$-$	$-$	$-$	0	$+$	$+$	$+$	$+$	$+$						
p_1	$+$	$+$	$+$	$+$	$+$	$+$	$+$	0	$-$	$-$	$-$	$-$	$-$	0	$+$						
p	$+$	0	$-$	$-$	$-$	0	$-$	$-$	$-$	$-$	$-$	0	$+$	$+$	$+$						

Building the Sign Matrix from a 'smaller' one

- Fill in the 2 intervals, $(-\infty, x_1), (x_n, +\infty)$ as before.
- Throw away the first and last column
- Remove the row labeled by p'
- 'Condense' the matrix to remove points that are not roots of one of $p, p_1, \ldots, p_n$

	x_1			x_2		x_3				x_4		x_5			x_6			x_7		
p'	−	−	−	0	+	0	−	−	−	0	+	+	+	+	+					
p_1	+	+	+	+	+	+	+	0	−	−	−	−	−	0	+					
p	+	0	−	−	−	0	−	−	−	−	−	0	+	+	+					

Building the Sign Matrix from a 'smaller' one

- Fill in the 2 intervals, $(-\infty, x_1), (x_n, +\infty)$ as before.
- Throw away the first and last column
- Remove the row labeled by p'
- 'Condense' the matrix to remove points that are not roots of one of $p, p_1, \ldots, p_n$

		x_1			x_2			x_3			x_4			x_5	
p_1	+	+	+	+	+	0	−	−	−	0	+				
p	+	0	−	0	−	−	−	0	+	+	+				

This completes the construction

Building Sign Matrix

Using the algorithm we just described, we can build the sign matrix of $p_1, p_2, \ldots, p_n$ as follows:

- If all p_i have degree 0, i.e. $p_i = c_i$ for all i, then the sign matrix is

p_1	$\text{sign}(c_1)$
p_2	$\text{sign}(c_2)$
.	.
.	.
p_n	$\text{sign}(c_n)$

Building Sign Matrix

Using the algorithm we just described, we can build the sign matrix of $p_1, p_2, \ldots, p_n$ as follows:

- If all p_i have degree 0, i.e. $p_i = c_i$ for all i, then the sign matrix is

p_1	$\text{sign}(c_1)$
p_2	$\text{sign}(c_2)$
.	.
.	.
p_n	$\text{sign}(c_n)$

- Else, use recursion to find the sign matrix of $p', p_2, p_3, \ldots, p_n, r_0, r_2, r_3, \ldots, r_n$.
  Using this and the previous algorithm, find the sign matrix of $p_1, p_2, \ldots, p_n$

Building Sign Matrix

Using the algorithm we just described, we can build the sign matrix of $p_1, p_2, \ldots, p_n$ as follows:

- If all p_i have degree 0, i.e. $p_i = c_i$ for all i, then the sign matrix is

p_1	$\text{sign}(c_1)$
p_2	$\text{sign}(c_2)$
.	.
.	.
p_n	$\text{sign}(c_n)$

- Else, use recursion to find the sign matrix of
  $p', p_2, p_3, \ldots, p_n, r_0, r_2, r_3, \ldots, r_n$.
  Using this and the previous algorithm, find the sign matrix of
  $p_1, p_2, \ldots, p_n$

Termination?

Building Sign Matrix-Termination

Termination using König's lemma

Given $p_1, \ldots, p_n$ we set up a finitely branching tree as follows:

- Let the root of the tree be a polynomial which has degree strictly larger than the degree of all p_i
- let $p_1, \ldots, p_n$ be the children of the root node.
- We build the rest of the tree inductively as follows:
  - let $s_1, \ldots, s_m$ be the leaves of the current tree.
  - If $\deg(s_i) = 0$ for all i, then we stop here and return this as final tree.
  - If $\deg(s_i) > 0$ for some i,find s_j which has maximum degree and add $s_j', r_0, r_1, .., r_{j-1}, r_{j+1}, .., r_m$ as the children of s_j (*Finitely branching*).

Key observation: degree of child $<$ degree of parent

Building Sign Matrix-Termination

Termination using König's lemma

Now after we have built the tree, assume that our algorithm doesn't terminate. This means our tree is infinite.
But by König's lemma, we will have an infinite branch. But note that as we go down a branch, the degree of the polynomials are strictly decreasing
$\Rightarrow$ infinite decent in $\mathbb{N}$
$\Rightarrow$ contradiction

Thus, our algorithm terminates.

Building Sign Matrix-Termination

Termination using König's lemma

Now after we have built the tree, assume that our algorithm doesn't terminate. This means our tree is infinite.
But by König's lemma, we will have an infinite branch. But note that as we go down a branch, the degree of the polynomials are strictly decreasing
$\Rightarrow$ infinite decent in $\mathbb{N}$
$\Rightarrow$ contradiction

Thus, our algorithm terminates.

This finishes quantifier elimination for univariate polynomials.

Can we extend this to the case where there are more than 1 variables in the formula? (Multivariate case)

Multivariate case

Issues with multiple variables

In the general case, where we have a formula of the form

$$\exists x : \alpha_1(x, y_1, \ldots, y_m) \wedge \cdots \wedge \alpha_n(x, y_1, \ldots, y_m),$$

where each α_i is of the form $p_i(x, y_1, \ldots, y_m) \bowtie_i 0$, and the y_j s are free variables, we cannot use the same procedure.

Example

Consider the formula $\exists x : yx > 0$. We cannot directly construct the sign matrix here, as the sign of yx depends on the value of y. In this case, we can solve this by splitting into three cases (depending on the sign of y), and creating a sign matrix for each separately.

Let $\phi_+(y)$, $\phi_0(y)$ and $\phi_-(y)$ be the resulting quantifier-free fromulae from each of these matrices (where ϕ_+ corresponds to the case where $y > 0$, etc.)

Multivariate case

Example

Then $\exists x : yx > 0$ can be rewritten as:
$(y > 0 \wedge \phi_+(y)) \vee$
$(y = 0 \wedge \phi_0(y)) \vee$
$(y < 0 \wedge \phi_-(y))$.
Specifically, in this case, we have the following sign matrices:

$y > 0$

	x_1		
$p(x)$	-1	0	$+1$

$\phi_+(y)$:
$(-1 = +1) \vee$
$(0 = +1) \vee$
$(+1 = +1)$

$y = 0$

$p(x)$	0

$\phi_0(y)$:
$(0 = +1)$

$y < 0$

	x_1		
$p(x)$	$+1$	0	-1

$\phi_+(y)$:
$(+1 = +1) \vee$
$(0 = +1) \vee$
$(-1 = +1)$

Multivariate case

Example

Then $\exists x : yx > 0$ is equivalent to:
$(y > 0 \wedge ((-1 = +1) \vee (0 = +1) \vee (+1 = +1))) \vee$
$(y = 0 \wedge (0 = +1)) \vee$
$(y < 0 \wedge ((+1 = +1) \vee (0 = +1) \vee (-1 = +1)))$.

In general, the process requires many more cases and subcases.

Division algorithm (multivariate)

Pseudo-division

Given two polynomials $p(x)$ and $q(x)$, which may contain other variables in addition to x, we can 'divide' p by q, to get polynomials s and r satisfying the equation

$$a^k p(x) = s(x)q(x) + r(x)\,,$$

with $\deg_x(r) < \deg_x(q)$. (Here, a is the leading coefficient of $q(x)$)

Division algorithm (multivariate)

Algorithm

Let $\deg_x(q) = n$, and $\deg_x(p) = m$.

- If $n > m$, then setting $s(x) \equiv 0$ and $r(x) = p(x)$ works.

Division algorithm (multivariate)

Algorithm

Let $\deg_x(q) = n$, and $\deg_x(p) = m$.

- If $n > m$, then setting $s(x) \equiv 0$ and $r(x) = p(x)$ works.
- Consider the case where $n \leq m$.
- Take the (nonzero) leading coefficients of $q(x)$ and $p(x)$ to be a and b, respectively, so that $q(x) = ax^n + q_0(x)$, and $p(x) = bx^m + p_0(x)$. Then we have

$$ap(x) = bx^{m-n}q(x) + (ap_0(x) - bx^{m-n}q_0(x)).$$

Division algorithm (multivariate)

Algorithm

Let $\deg_x(q) = n$, and $\deg_x(p) = m$.

- If $n > m$, then setting $s(x) \equiv 0$ and $r(x) = p(x)$ works.
- Consider the case where $n \leq m$.
- Take the (nonzero) leading coefficients of $q(x)$ and $p(x)$ to be a and b, respectively, so that $q(x) = ax^n + q_0(x)$, and $p(x) = bx^m + p_0(x)$. Then we have

$$ap(x) = bx^{m-n}q(x) + (ap_0(x) - bx^{m-n}q_0(x)).$$

- Note that $r'(x) = ap_0(x) - bx^{m-n}q_0(x)$ has lower degree than $p(x)$. Now we can repeat the step above, with $r'(x)$ taking the place of $p(x)$. This gives us

$$ar'(x) = s'(x)q(x) + r''(x).$$

Division algorithm (multivariate)

Algorithm (cont.)

- Composing the previous two steps gives us

$$a^2 p(x) = \left[abx^{m-n} q(x) + s'(x)\right] q(x) + r''(x)\,.$$

- Continue this process by dividing the remainder polynomial by q at each step. As the degree of the remainder decreases at each step, it eventually becomes less than n, and the algorithm terminates.
- If it terminates after k divisions, we have

$$a^k p(x) = s_1(x) q(x) + r_1(x)\,,$$

with $\deg_x(r_1) < \deg_x(q)$.

Division algorithm (multivariate)

Algorithm (cont.)

- If we consider univariate polynomials (so that a is a nonzero constant), dividing throughout by a^k gets us

$$p(x) = s(x)q(x) + r(x)\,,$$

  which is the required result.
- However, if p and q are multivariate, then a might be a polynomial in the other variables. In this case, we leave the result as it is.

Sign-preserving pseudo-division algorithm

Pseudo-division

Given two polynomials $p(x)$ and $q(x)$, which may contain other variables in addition to x, we can 'divide' p by q, to get polynomials s and r satisfying the equation

$$a^k p(x) = s(x)q(x) + r(x)\,,$$

with $\deg_x(r) < \deg_x(q)$. (Here, a is the leading coefficient of $q(x)$)

In the univariate case, we had no prefactor of a^k, so at the roots of q, $sgn(p) = sgn(r)$.
For the multivariate case, this only holds at points where $a^k > 0$. (Note that a is a polynomial in the y_i s.) So we need to modify the algorithm a bit.

Sign-preserving pseudo-division algorithm

Sign-preserving pseudo-division algorithm

a is the leading coefficient of $q(x)$. Depending on the values of $y_1, \ldots, y_m$, we have the following cases:

- If $a = 0$, then a is not the leading coefficient of q, and so we move to the next coefficient and recurse. (Edge case - all coefficients are zero)
- If $a \neq 0$, then we perform pseudo-division to get $a^k p = s \cdot q + r'$.

Sign-preserving pseudo-division algorithm

Sign-preserving pseudo-division algorithm

a is the leading coefficient of $q(x)$. Depending on the values of $y_1, \ldots, y_m$, we have the following cases:

- If $a = 0$, then a is not the leading coefficient of q, and so we move to the next coefficient and recurse. (Edge case - all coefficients are zero)
- If $a \neq 0$, then we perform pseudo-division to get $a^k p = s \cdot q + r'$.
- If $a^k < 0$ (i.e. if $a < 0$ and k is odd), we multiply both sides by a to get $a^{k+1} p = as \cdot q + ar'$. Now $a^{k+1} > 0$, so we can set $r = ar'$.
- Otherwise ($a > 0$, or k even), simply set $r = r'$.

Sign-preserving pseudo-division algorithm

Sign-preserving pseudo-division algorithm

a is the leading coefficient of $q(x)$. Depending on the values of $y_1, \ldots, y_m$, we have the following cases:

- If $a = 0$, then a is not the leading coefficient of q, and so we move to the next coefficient and recurse. (Edge case - all coefficients are zero)
- If $a \neq 0$, then we perform pseudo-division to get $a^k p = s \cdot q + r'$.
- If $a^k < 0$ (i.e. if $a < 0$ and k is odd), we multiply both sides by a to get $a^{k+1} p = as \cdot q + ar'$. Now $a^{k+1} > 0$, so we can set $r = ar'$.
- Otherwise ($a > 0$, or k even), simply set $r = r'$.
- In both these above cases, the prefactor of $p(x)$ is positive, so we have $sgn(p(x_i)) = sgn(r(x_i))$ at all roots x_i of $q(x)$. So r is a sign-preserving remainder on 'division' of p by q.

Sign-preserving pseudo-division algorithm

Edge case

While performing case splits, there is a case wherein we assume all the coefficients of q are zero. Here we simply return *None* which will will be handled separately when are building the sign matrix.

Building a sign matrix

- Given $p, p_1, p_2, \ldots, p_n$, we can construct their sign matrices from the sign matrices of $p', p_1, \ldots, p_n, r_0, r_1, \ldots, r_n$, in the same manner as earlier.
- The only difference is that now we have multiple cases, corresponding to multiple choices for the signs of the coefficients. Under the assumptions of each case, we can create separate sign matrices for $p, p_1, p_2, \ldots, p_n$.

Building a sign matrix

- Given $p, p_1, p_2, \ldots, p_n$, we can construct their sign matrices from the sign matrices of $p', p_1, \ldots, p_n, r_0, r_1, \ldots, r_n$, in the same manner as earlier.
- The only difference is that now we have multiple cases, corresponding to multiple choices for the signs of the coefficients. Under the assumptions of each case, we can create separate sign matrices for $p, p_1, p_2, \ldots, p_n$.
- At the end, this leaves us with a large number of sign matrices. Each of them can be converted into a quantifier-free formula, and put in conjunction with the assumptions for that case. The disjunction of all these formulae is the required end-result.

Building a sign matrix

- Given $p, p_1, p_2, \ldots, p_n$, we can construct their sign matrices from the sign matrices of $p', p_1, \ldots, p_n, r_0, r_1, \ldots, r_n$, in the same manner as earlier.
- The only difference is that now we have multiple cases, corresponding to multiple choices for the signs of the coefficients. Under the assumptions of each case, we can create separate sign matrices for $p, p_1, p_2, \ldots, p_n$.
- At the end, this leaves us with a large number of sign matrices. Each of them can be converted into a quantifier-free formula, and put in conjunction with the assumptions for that case. The disjunction of all these formulae is the required end-result.
- We can use the same argument from earlier to say that a sign matrix for any $p_1, p_2, \ldots, p_n$, can be constructed in finite time.

Building a sign matrix

Edge cases

- If r_0 is *None*, then all the coefficients of p' were assumed to be zero. This would mean p is a constant, hence we simply find sign of the coefficient of x^0 and append a constant row to the sign matrix of $p_1, \ldots, p_n$ and label the row as p.
- If r_i is *None* where $i \geq 1$, then all the coefficients of p_i were assumed to be zero. This would mean p_i is a constant, hence it has no roots and doesn't help in building the sign matrix. Thus we simply ignore p_i and build the sign matrix for $p_1, .., p_{i-1}, p_{i+1}, .., p_n, r_0, .., r_{i-1}, r_{i+1}, .., r_n$ from which we build sign matrix of $p, p_1, .., p_{i-1}, p_{i+1}, .., p_n$ to which we can simply add the constant row corresponding p_i.

An example

Example

Eliminating $\exists$ from the formula $\exists x : y(x^2 + 1) > 0$.

- We have $p = y(x^2 + 1)$, and $p' = 2xy$.
- On pseudo-division of p by p', we get $(2y) \cdot y(x^2 + 1) = yx \cdot 2yx + 2y^2$.
- Divide into cases based on the sign of $2y$:

  - Case 1: if $2y > 0$, then $r_0 = 2y^2$.
  - Case 2: if $2y < 0$, then $r_0 = 2y \cdot 2y^2 = 4y^3$.
  - Case 3: if $2y = 0$, then we move to the next coefficient. However, this is the last coefficient, so p' is the zero polynomial, and so p is constant over the real line.

An example

Case 1

- To get the sign matrix of p, we need the sign-matrix of $2xy$ and $2y^2$.

An example

Case 1

- To get the sign matrix of p, we need the sign-matrix of $2xy$ and $2y^2$.
- We have $(2xy)' = 2y$. On pseudo-division, this gives us $(2y) \cdot 2xy = (2yx) \cdot 2y + 0$.
- Similarly, on pseudo-dividing $2xy$ by $2y^2$, we get $(2y^2) \cdot 2xy = (2yx) \cdot 2y^2 + 0$.

An example

Case 1

- To get the sign matrix of p, we need the sign-matrix of $2xy$ and $2y^2$.
- We have $(2xy)' = 2y$. On pseudo-division, this gives us $(2y) \cdot 2xy = (2yx) \cdot 2y + 0$.
- Similarly, on pseudo-dividing $2xy$ by $2y^2$, we get $(2y^2) \cdot 2xy = (2yx) \cdot 2y^2 + 0$.
- So to find the sign matrix of $2xy$ and $2y^2$, we need to find the sign matrix of $\{2y^2, 2y, 0\}$ (there will be 3×3 cases, depending on the signs of $2y^2$ and $2y$).

Case 2 is dealt with in a similar way.

An example

So our resulting formula is a disjunction of clauses, with each clause corresponding to different cases. For example, part of the formula arising from subcases of Case 1 (where $y > 0$) is:

$((2y > 0 \wedge 2y^2 > 0 \wedge y > 0) \wedge (+1 = +1)) \vee$

$((2y < 0 \wedge 2y^2 > 0 \wedge y > 0) \wedge (-1 = +1)) \vee$

$\ldots$

$\ldots$

$\ldots$

References

- McLaughlin S., Harrison J. (2005). A Proof-Producing Decision Procedure for Real Arithmetic.
- Schoutens, H. (2010). Muchnik's Proof of Tarski-Seidenberg.