Subtropical Satisfiability for SMT Solving

NASA Formal Methods Symposium (NFM 2023)

Jasper Nalbach, Erika Ábrahám

18 May, 2023

Satisfiability-modulo-theories (SMT)

Boolean satisfiability (SAT) ...

$$(a \vee \neg b) \wedge (\neg a \vee b \vee c)$$

Satisfiability-modulo-theories (SMT)

Boolean satisfiability (SAT) ...

$$(a \vee \neg b) \wedge (\neg a \vee b \vee c)$$

$$a = \textit{true}, \quad b = \textit{false}, \quad c = \textit{true}$$

Satisfiability-modulo-theories (SMT)

Boolean satisfiability (SAT) ...

$$(a \vee \neg b) \wedge (\neg a \vee b \vee c)$$

$$a = \textit{true}, \quad b = \textit{false}, \quad c = \textit{true}$$

... modulo quantifier-free non-linear real arithmetic (QFNRA)

$$(x - 2y > 0 \vee x^2 - 2 = 0) \wedge x^4 y + 2x^2 - 4 > 0$$

Satisfiability-modulo-theories (SMT)

Boolean satisfiability (SAT) ...

$$(a \vee \neg b) \wedge (\neg a \vee b \vee c)$$

$$a = \textit{true}, \quad b = \textit{false}, \quad c = \textit{true}$$

... modulo quantifier-free non-linear real arithmetic (QFNRA)

$$(x - 2y > 0 \vee x^2 - 2 = 0) \wedge x^4 y + 2x^2 - 4 > 0$$

$$x = \sqrt{2}, \quad y = 2$$

Satisfiability-modulo-theories (SMT)

Boolean satisfiability (SAT) ...

$$(a \vee \neg b) \wedge (\neg a \vee b \vee c)$$

$$a = \textit{true}, \quad b = \textit{false}, \quad c = \textit{true}$$

... modulo quantifier-free non-linear real arithmetic (QFNRA)

$$(x - 2y > 0 \vee x^2 - 2 = 0) \wedge x^4 y + 2x^2 - 4 > 0$$

$$x = \sqrt{2}, \quad y = 2$$

... modulo quantifier-free linear real arithmetic (QFLRA)

$$(x - 2y > 0 \vee x - 2 = 0) \wedge y + 2x - 4 > 0$$

Satisfiability-modulo-theories (SMT)

Boolean satisfiability (SAT) ...

$$(a \vee \neg b) \wedge (\neg a \vee b \vee c)$$

$$a = \textit{true}, \quad b = \textit{false}, \quad c = \textit{true}$$

... modulo quantifier-free non-linear real arithmetic (QFNRA)

$$(x - 2y > 0 \vee x^2 - 2 = 0) \wedge x^4 y + 2x^2 - 4 > 0$$

$$x = \sqrt{2}, \quad y = 2$$

... modulo quantifier-free linear real arithmetic (QFLRA)

$$(x - 2y > 0 \vee x - 2 = 0) \wedge y + 2x - 4 > 0$$

$$x = 2, \quad y = 2$$

DPLL(T)

$$\varphi = (x - 2y > 0 \vee x^2 - 2 = 0) \wedge x^4 y + 2x^2 - 4 > 0$$

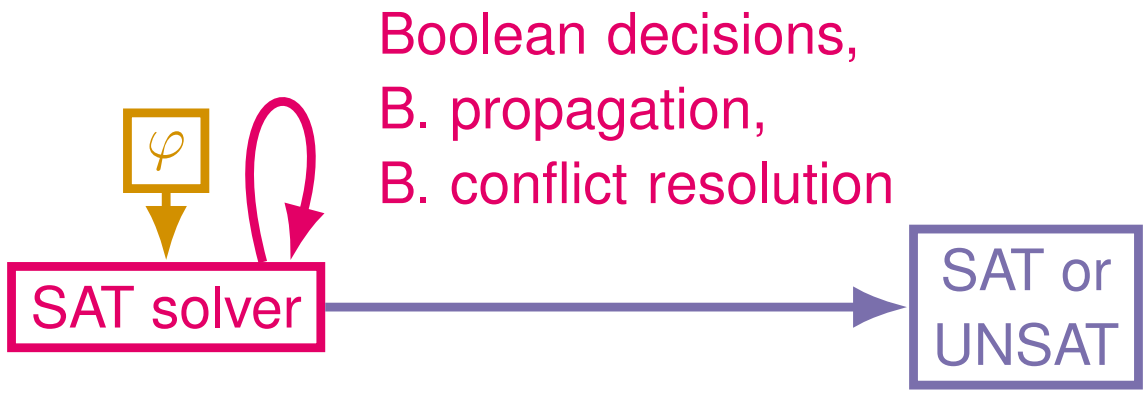
DPLL(T)

$$\varphi = (\underbrace{x - 2y > 0}_{b_1} \vee \underbrace{x^2 - 2 = 0}_{b_2}) \wedge \underbrace{x^4 y + 2x^2 - 4 > 0}_{b_3}$$

DPLL(T)

$$\varphi = (\underbrace{x - 2y > 0}_{b_1} \vee \underbrace{x^2 - 2 = 0}_{b_2}) \wedge \underbrace{x^4 y + 2x^2 - 4 > 0}_{b_3}$$

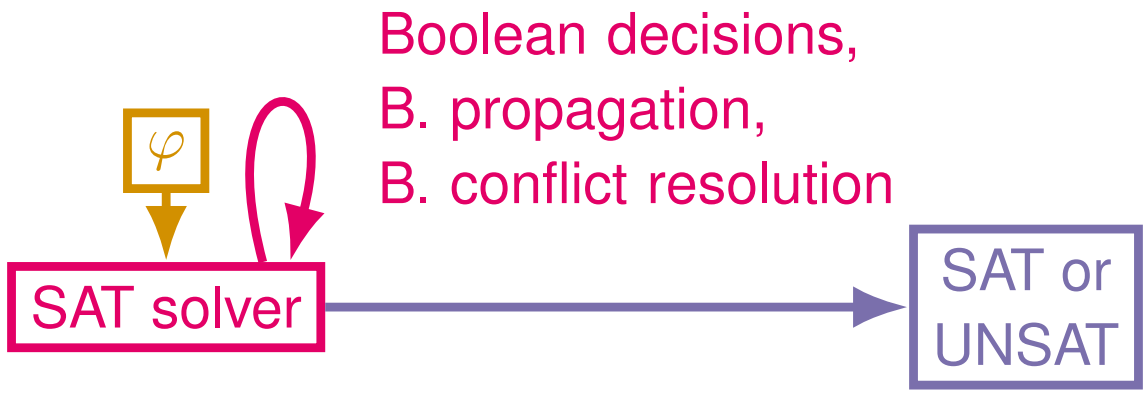
$(b_1 \vee b_2) \wedge b_3$



DPLL(T)

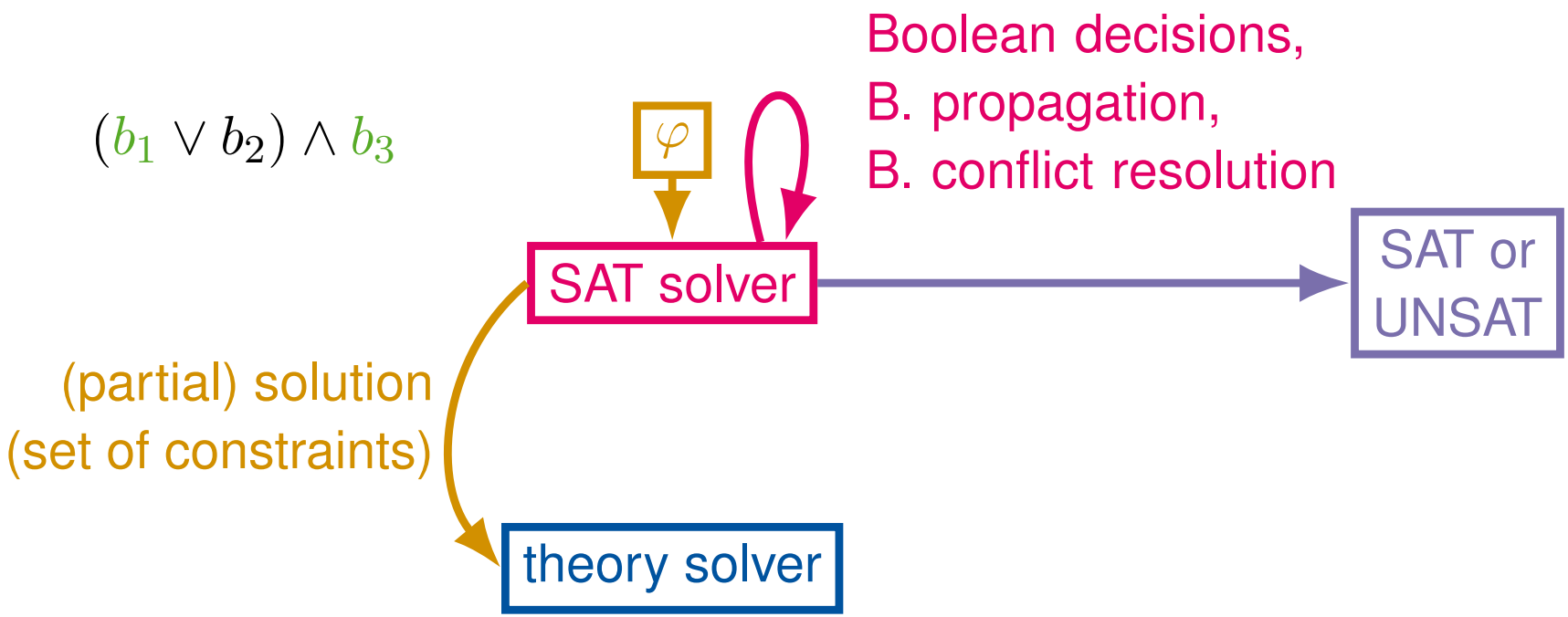
$$\varphi = (\underbrace{x - 2y > 0}_{b_1} \vee \underbrace{x^2 - 2 = 0}_{b_2}) \wedge \underbrace{x^4 y + 2x^2 - 4 > 0}_{b_3}$$

$(b_1 \vee b_2) \wedge b_3$



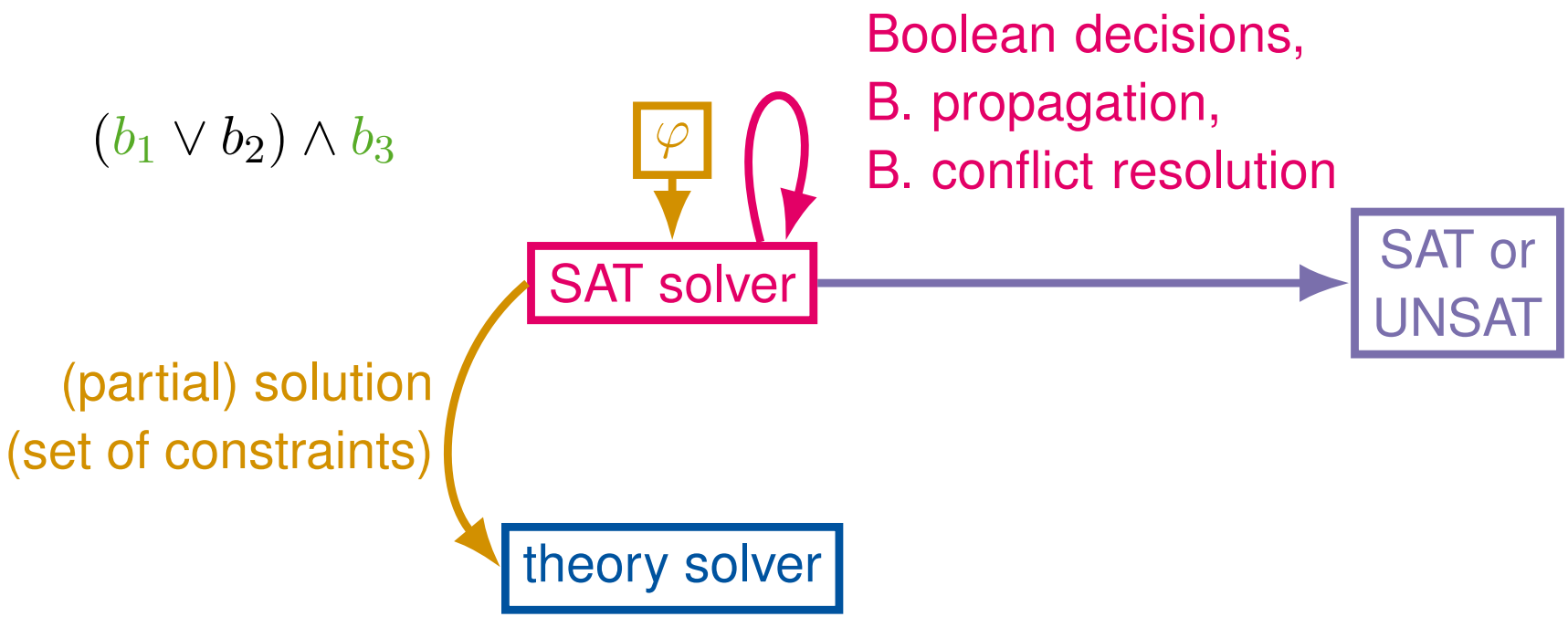
DPLL(T)

$$\varphi = (\underbrace{x - 2y > 0}_{b_1} \vee \underbrace{x^2 - 2 = 0}_{b_2}) \wedge \underbrace{x^4 y + 2x^2 - 4 > 0}_{b_3}$$



DPLL(T)

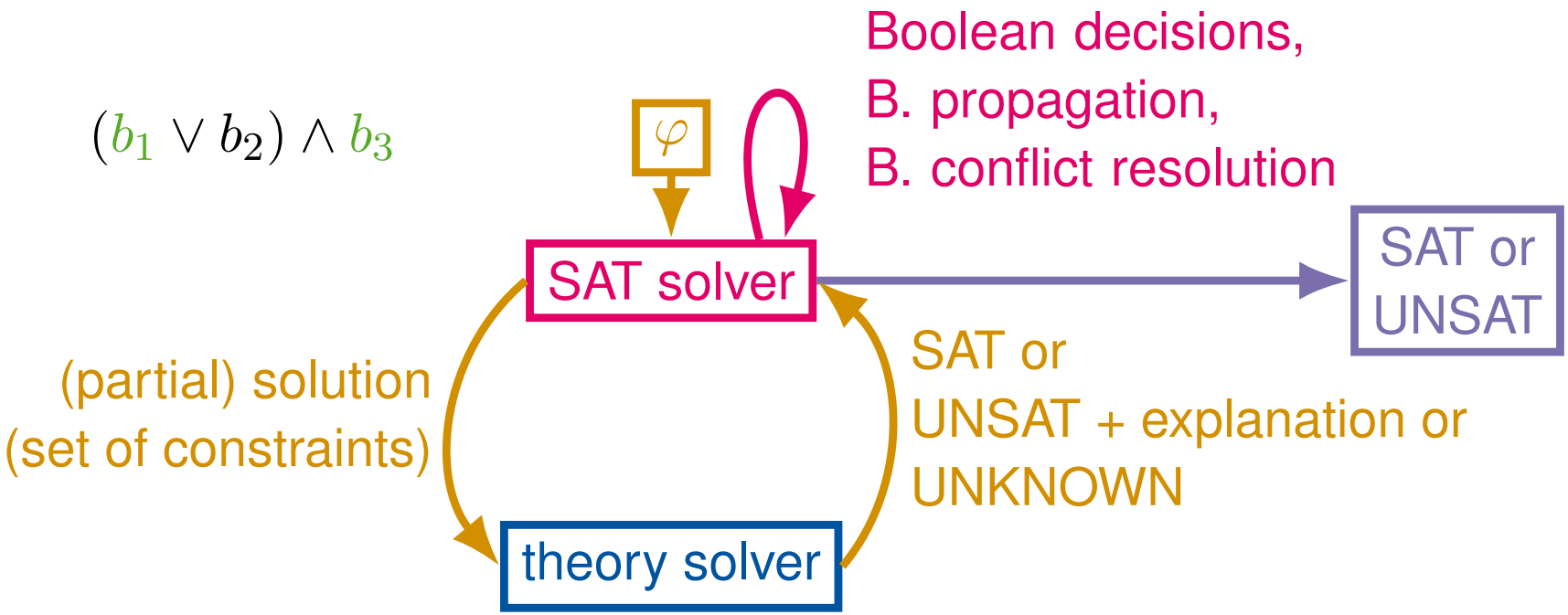
$$\varphi = (\underbrace{x - 2y > 0}_{b_1} \vee \underbrace{x^2 - 2 = 0}_{b_2}) \wedge \underbrace{x^4 y + 2x^2 - 4 > 0}_{b_3}$$



$$x - 2y > 0 \wedge x^4 y + 2x^2 - 4 > 0$$

DPLL(T)

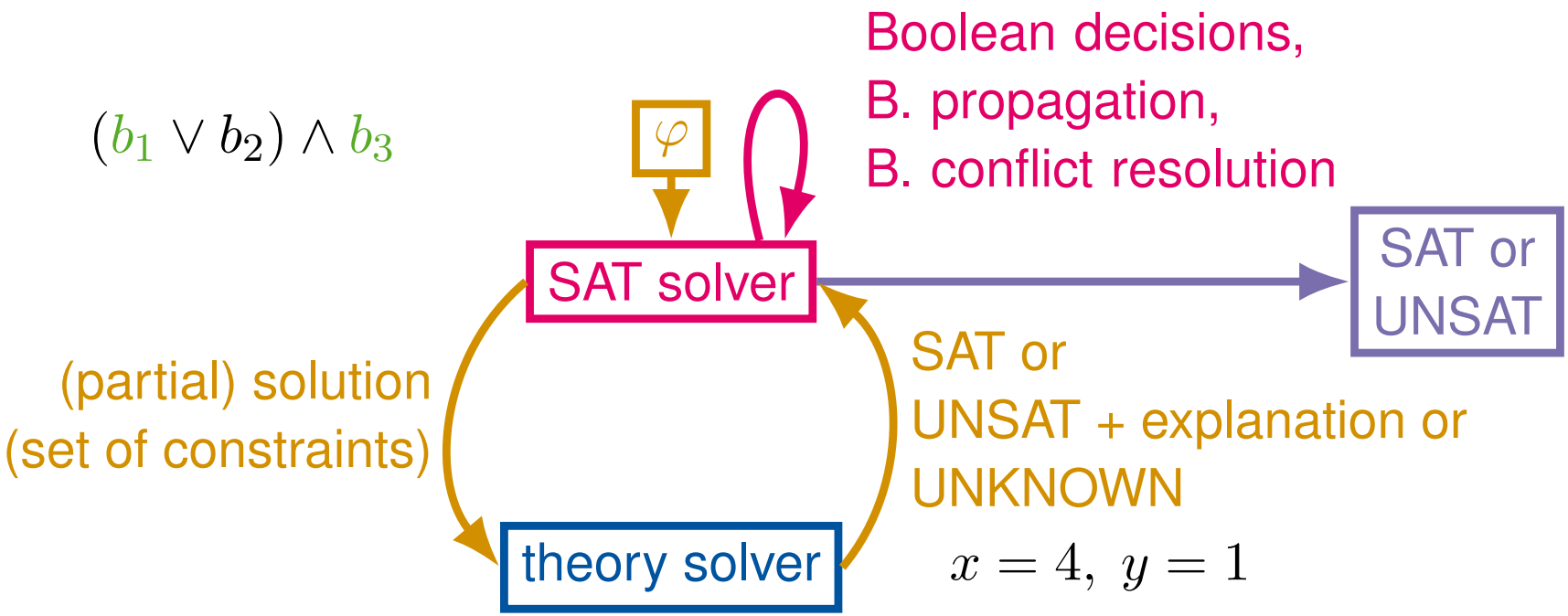
$$\varphi = (\underbrace{x - 2y > 0}_{b_1} \vee \underbrace{x^2 - 2 = 0}_{b_2}) \wedge \underbrace{x^4 y + 2x^2 - 4 > 0}_{b_3}$$



$$x - 2y > 0 \wedge x^4 y + 2x^2 - 4 > 0$$

DPLL(T)

$$\varphi = (\underbrace{x - 2y > 0}_{b_1} \vee \underbrace{x^2 - 2 = 0}_{b_2}) \wedge \underbrace{x^4 y + 2x^2 - 4 > 0}_{b_3}$$



$$x - 2y > 0 \wedge x^4 y + 2x^2 - 4 > 0$$

State-of-the-art QFNRA solving

- All complete QFNRA solvers are based on the doubly-exponential cylindrical algebraic decomposition (CAD)

State-of-the-art QFNRA solving

- All complete QFNRA solvers are based on the doubly-exponential cylindrical algebraic decomposition (CAD)
- Combination with incomplete methods such as virtual substitution [Weispfenning 1997] or incremental linearization [Cimatti et al. 2018]

State-of-the-art QFNRA solving

- All complete QFNRA solvers are based on the doubly-exponential cylindrical algebraic decomposition (CAD)
- Combination with incomplete methods such as virtual substitution [Weispfenning 1997] or incremental linearization [Cimatti et al. 2018]
- Some solvers do not use DPLL(T) but MCSAT [De Moura, Jovanović 2013] (which extends DPLL for theories directly)

State-of-the-art QFNRA solving

- All complete QFNRA solvers are based on the doubly-exponential cylindrical algebraic decomposition (CAD)
- Combination with incomplete methods such as virtual substitution [Weispfenning 1997] or incremental linearization [Cimatti et al. 2018]
- Some solvers do not use DPLL(T) but MCSAT [De Moura, Jovanović 2013] (which extends DPLL for theories directly)
- Linear real arithmetic is efficiently solved using simplex in practise

Subtropical satisfiability [Sturm 2015, Fontaine et al 2017]

- Is an incomplete method

Subtropical satisfiability [Sturm 2015, Fontaine et al 2017]

- Is an incomplete method
- which finds solutions for sets of inequations or a single equation,

Subtropical satisfiability [Sturm 2015, Fontaine et al 2017]

- Is an incomplete method
- which finds solutions for sets of inequations or a single equation,
- by encoding them in QFLRA,

Subtropical satisfiability [Sturm 2015, Fontaine et al 2017]

- Is an incomplete method
- which finds solutions for sets of inequations or a single equation,
- by encoding them in QFLRA,
- is simple and fast.

Solutions for a single inequation [Sturm 2015, Fontaine et al 2017]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} > 0$$

Solutions for a single inequation [Sturm 2015, Fontaine et al 2017]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} > 0$$

Idea: choose direction $(n_x, n_y) \in \mathbb{Q}^2$ and $a \in \mathbb{Q}$ large enough such that $p(a^{n_x}, a^{n_y}) > 0$

Solutions for a single inequation [Sturm 2015, Fontaine et al 2017]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} > 0$$

Idea: choose direction $(n_x, n_y) \in \mathbb{Q}^2$ and $a \in \mathbb{Q}$ large enough such that $p(a^{n_x}, a^{n_y}) > 0$

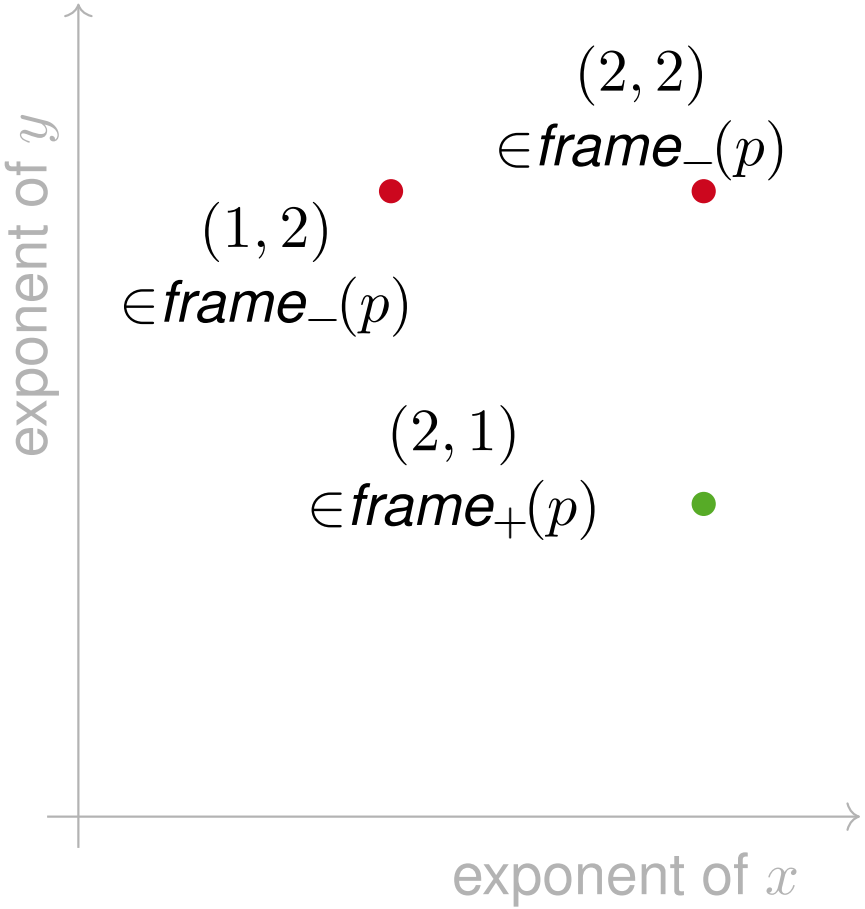
$2x^2y^1$ dominates p at $(a^{n_x}, a^{n_y}) \in \mathbb{Q}^2 \implies p(a^{n_x}, a^{n_y}) > 0$

Solutions for a single inequation [Sturm 2015, Fontaine et al 2017]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} > 0$$

Idea: choose direction $(n_x, n_y) \in \mathbb{Q}^2$ and $a \in \mathbb{Q}$ large enough such that $p(a^{n_x}, a^{n_y}) > 0$

$2x^2y^1$ dominates p at $(a^{n_x}, a^{n_y}) \in \mathbb{Q}^2 \implies p(a^{n_x}, a^{n_y}) > 0$

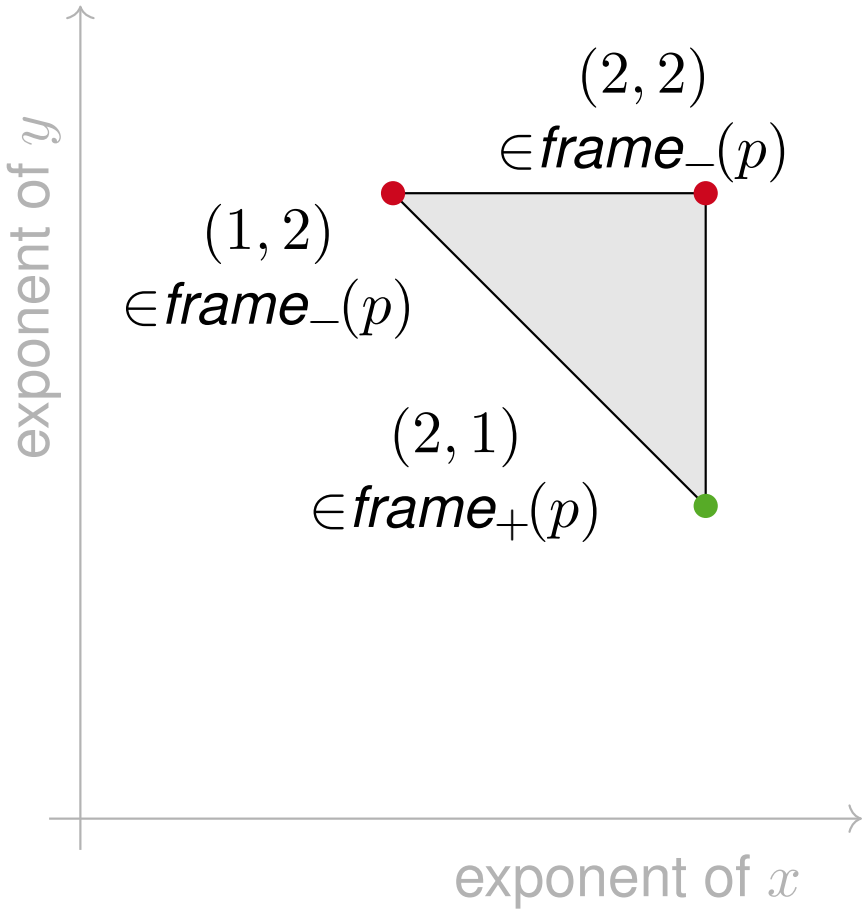


Solutions for a single inequation [Sturm 2015, Fontaine et al 2017]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} > 0$$

Idea: choose direction $(n_x, n_y) \in \mathbb{Q}^2$ and $a \in \mathbb{Q}$ large enough such that $p(a^{n_x}, a^{n_y}) > 0$

$2x^2y^1$ dominates p at $(a^{n_x}, a^{n_y}) \in \mathbb{Q}^2 \implies p(a^{n_x}, a^{n_y}) > 0$

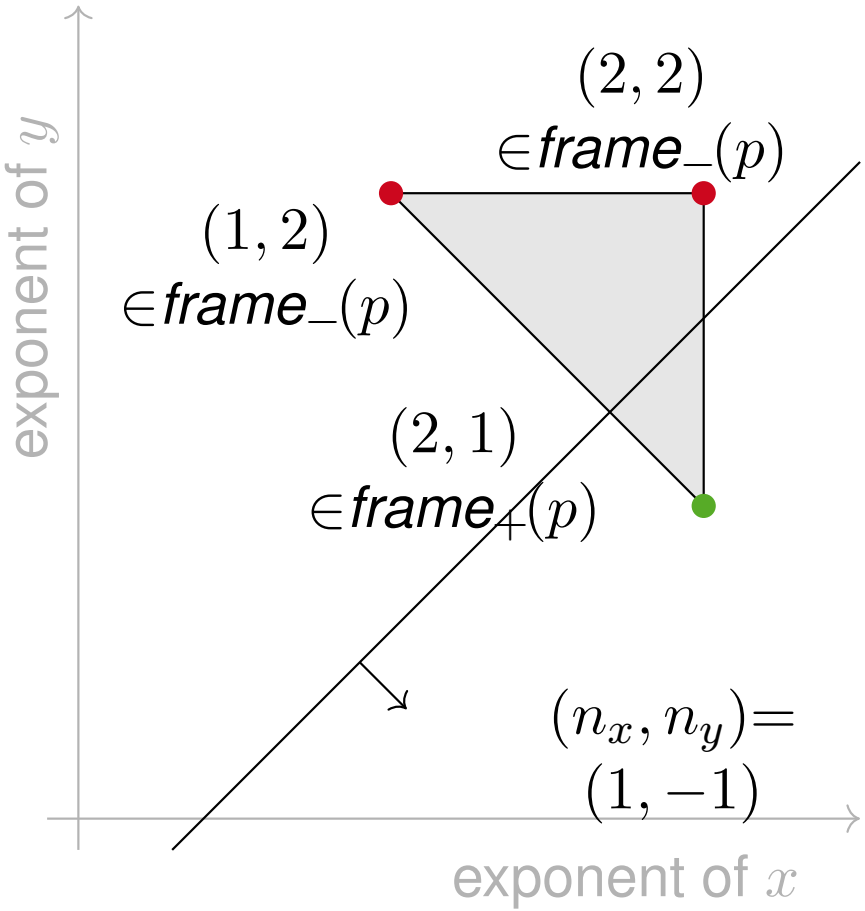


Solutions for a single inequation [Sturm 2015, Fontaine et al 2017]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} > 0$$

Idea: choose direction $(n_x, n_y) \in \mathbb{Q}^2$ and $a \in \mathbb{Q}$ large enough such that $p(a^{n_x}, a^{n_y}) > 0$

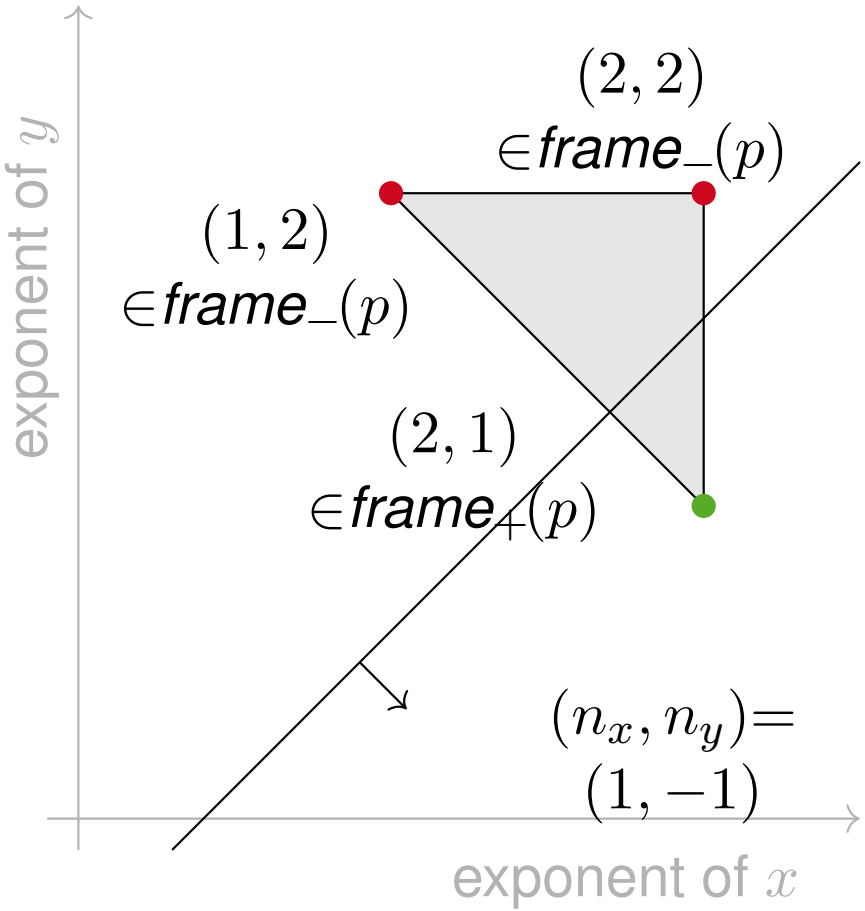
$2x^2y^1$ dominates p at $(a^{n_x}, a^{n_y}) \in \mathbb{Q}^2 \implies p(a^{n_x}, a^{n_y}) > 0$



Solutions for a single inequation [Sturm 2015, Fontaine et al 2017]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} > 0$$

$$\begin{aligned} &\mathcal{ST}_{p>0}(n_x, n_y) := \\ &\exists b.\ 2n_x + 1n_y > b \\ &\quad \wedge 1n_x + 2n_y < b \\ &\quad \wedge 2n_x + 2n_y < b \end{aligned}$$



$\rightarrow$ solve efficiently using simplex in practise!

Solutions for a single inequation [Sturm 2015, Fontaine et al 2017]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} > 0$$

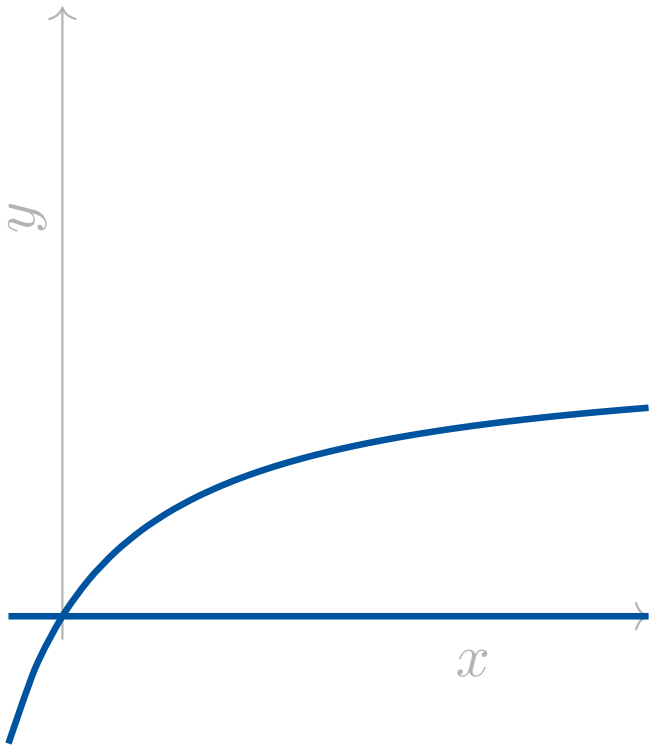
$$\begin{aligned}
&\mathcal{ST}_{p>0}(n_x, n_y) := \\
\exists b.\ & 2n_x + 1n_y > b \\
\wedge & 1n_x + 2n_y < b \\
\wedge & 2n_x + 2n_y < b
\end{aligned}$$

$\rightarrow$ solve efficiently using simplex in practise!

- $\mathcal{ST}_{p<0}, \mathcal{ST}_{p\geq 0}, \mathcal{ST}_{p\leq 0}, \mathcal{ST}_{p\neq 0}$ can be derived easily

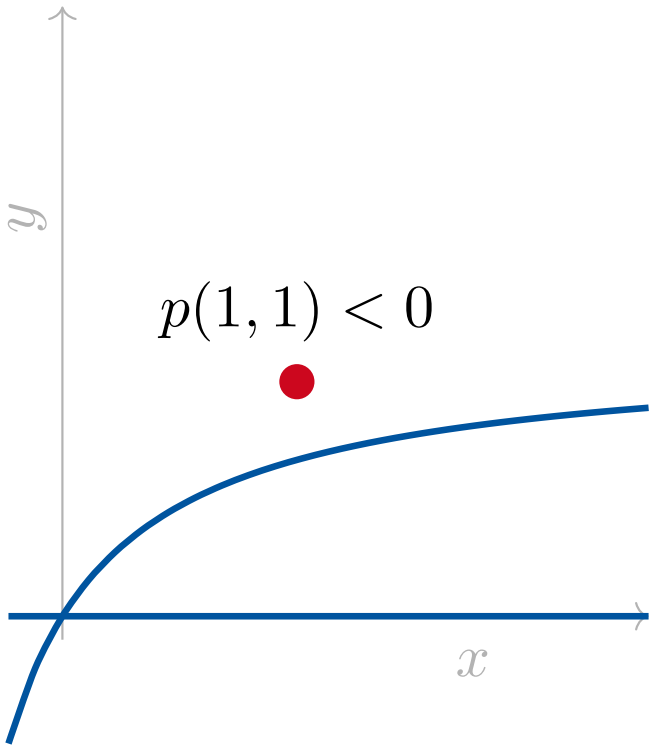
Solutions for a single equation [Sturm 2015]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} = 0$$



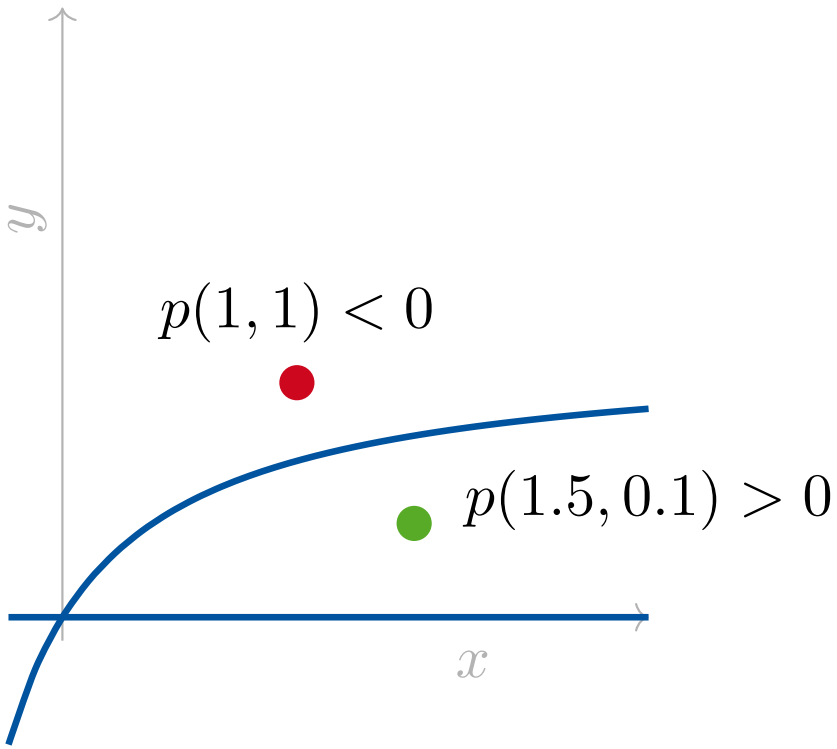
Solutions for a single equation [Sturm 2015]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} = 0$$



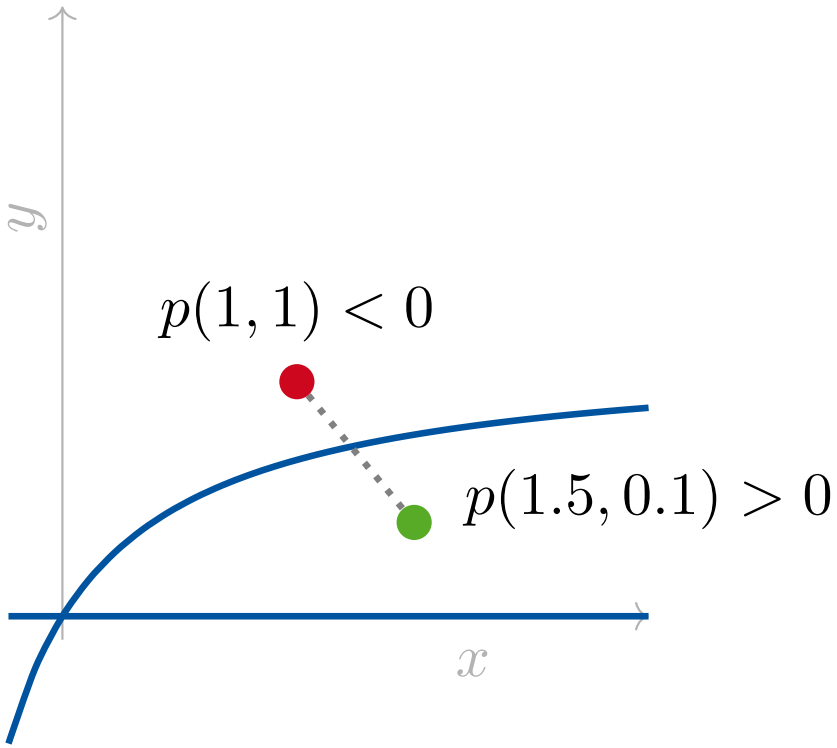
Solutions for a single equation [Sturm 2015]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} = 0$$



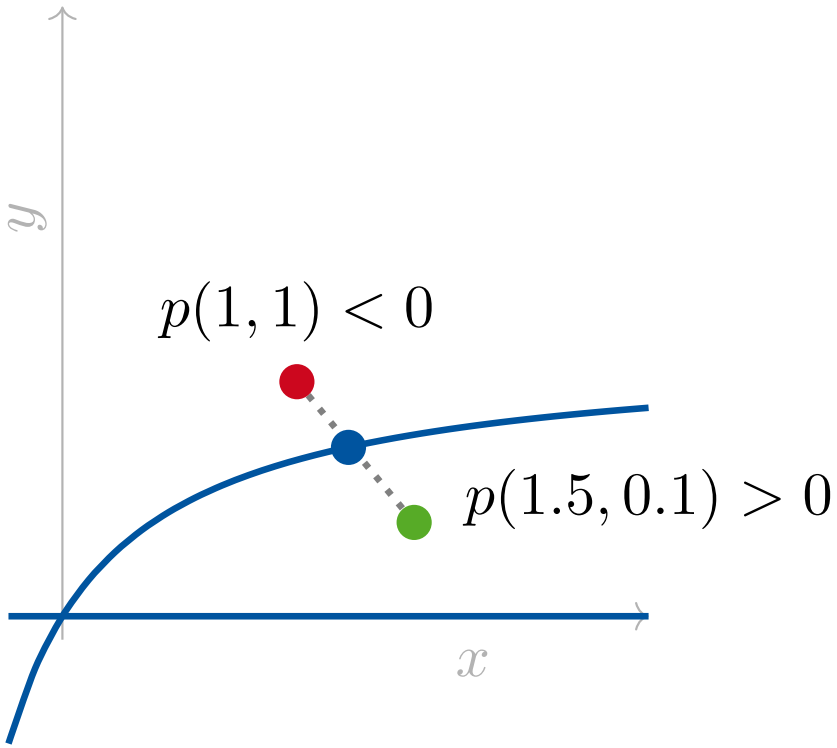
Solutions for a single equation [Sturm 2015]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} = 0$$



Solutions for a single equation [Sturm 2015]

$$\underbrace{-7x^2y^2 - 5x^1y^2 + 2x^2y^1}_{p} = 0$$



Generalization to formulas

How to make use of subtropical for general QFNRA formulas?

Transformation to single equation [Seidenberg 1954]

Idea: Transform a QFNRA formula φ to a single equation $Tr(\varphi)$ and apply subtropical to $Tr(\varphi)$

Transformation to single equation [Seidenberg 1954]

Idea: Transform a QFNRA formula φ to a single equation $Tr(\varphi)$ and apply subtropical to $Tr(\varphi)$

Transformation (excerpt):

$$Tr(p \geq 0) := p - (y_{p \geq 0})^2 = 0$$

$$Tr(\bigvee_{i=1}^{n} p_i = 0) := \prod_{i=1}^{n} p_i = 0$$

$$Tr(\bigwedge_{i=1}^{n} p_i = 0) := \sum_{i=1}^{n} (p_i)^2 = 0$$

Transformation to single equation [Seidenberg 1954]

Idea: Transform a QFNRA formula φ to a single equation $Tr(\varphi)$ and apply subtropical to $Tr(\varphi)$

Transformation (excerpt):

$$Tr(p \geq 0) := p - (y_{p \geq 0})^2 = 0$$

$$Tr(\bigvee_{i=1}^{n} p_i = 0) := \prod_{i=1}^{n} p_i = 0$$

$$Tr(\bigwedge_{i=1}^{n} p_i = 0) := \sum_{i=1}^{n} (p_i)^2 = 0$$

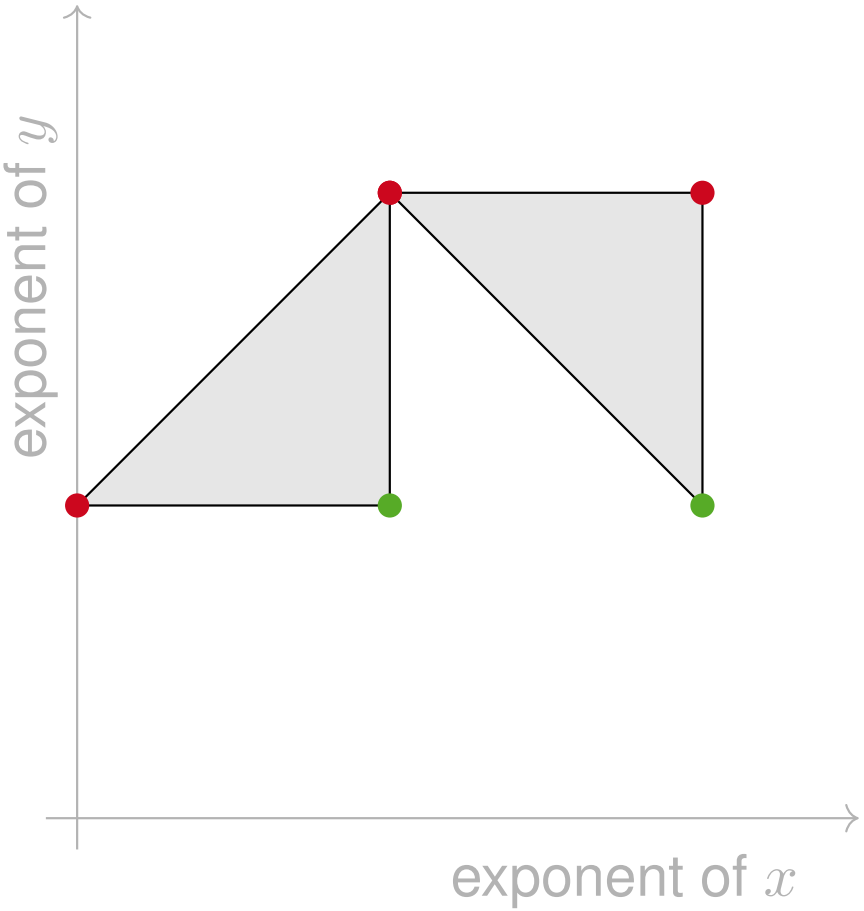
Subtropical is not applicable to sum-of-square polynomials!

Solutions for conjunctions of inequations [Fontaine et al 2017]

$$\varphi := -7x^2y^2 - 5x^1y^2 + 2x^2y^1 > 0 \wedge -7x^1y^2 - 5y^1 + 2x^1y^1 > 0$$

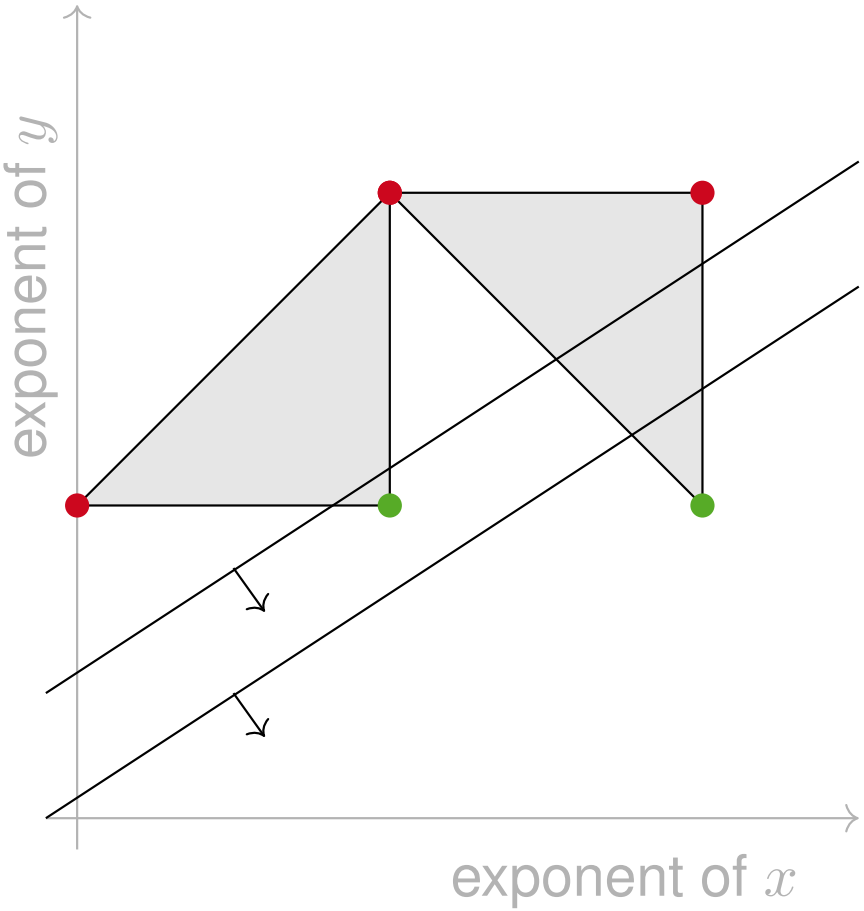
Solutions for conjunctions of inequations [Fontaine et al 2017]

$$\varphi := -7x^2y^2 - 5x^1y^2 + 2x^2y^1 > 0 \wedge -7x^1y^2 - 5y^1 + 2x^1y^1 > 0$$



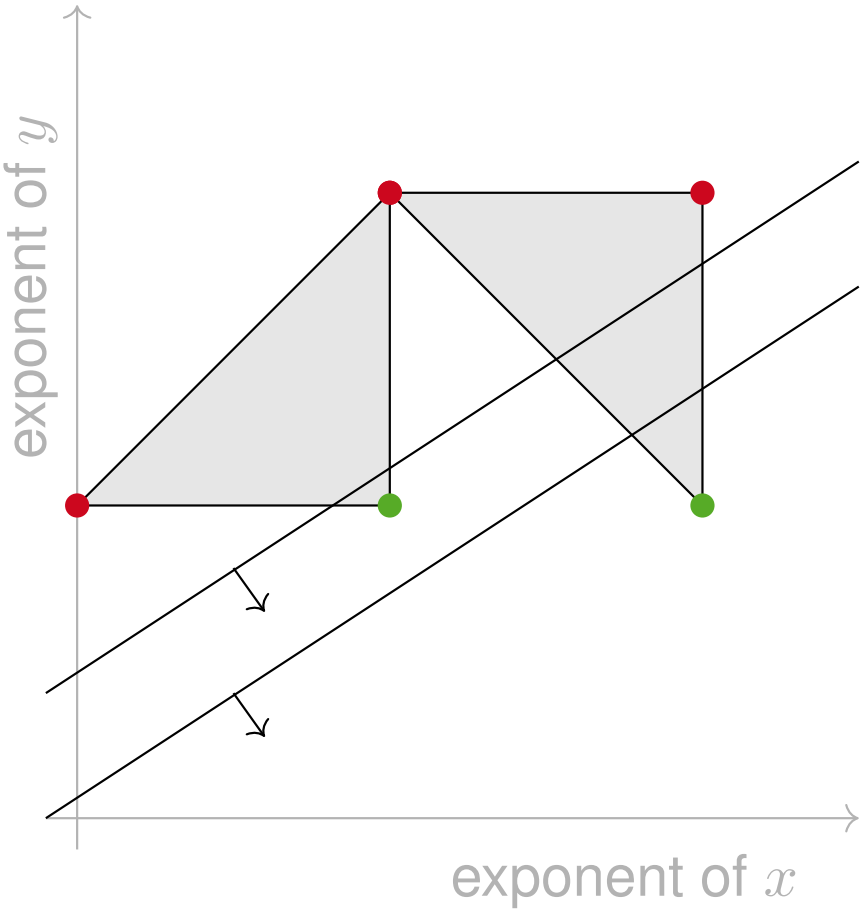
Solutions for conjunctions of inequations [Fontaine et al 2017]

$$\varphi := -7x^2y^2 - 5x^1y^2 + 2x^2y^1 > 0 \wedge -7x^1y^2 - 5y^1 + 2x^1y^1 > 0$$



Solutions for conjunctions of inequations [Fontaine et al 2017]

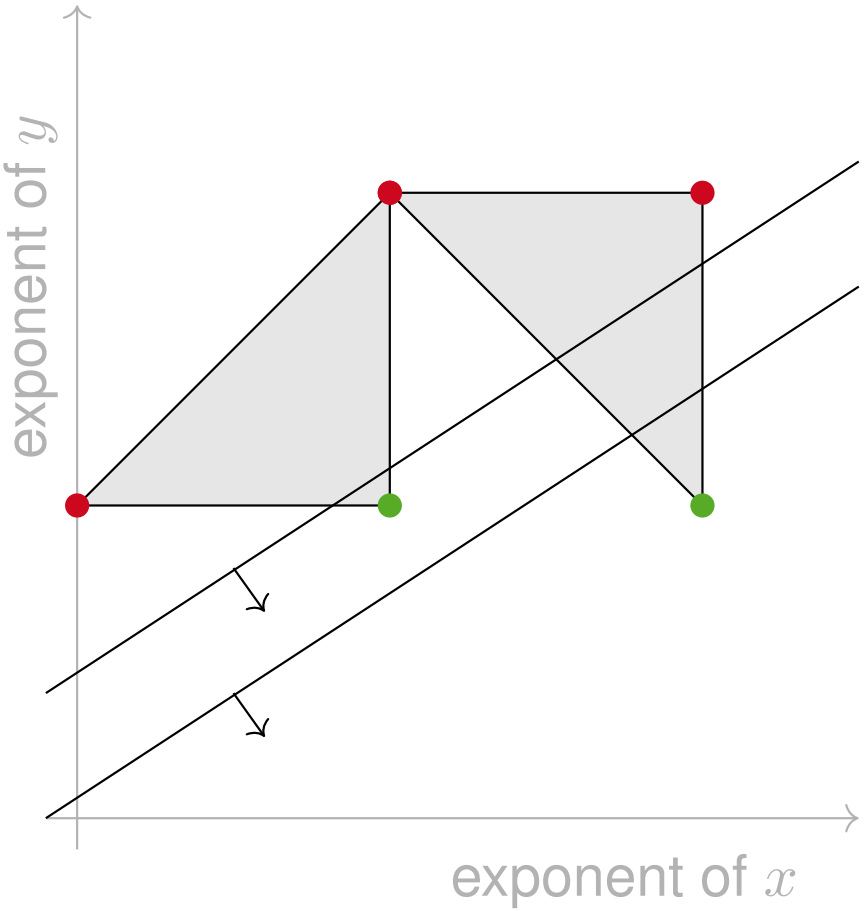
$$\varphi := -7x^2y^2 - 5x^1y^2 + 2x^2y^1 > 0 \wedge -7x^1y^2 - 5y^1 + 2x^1y^1 > 0$$



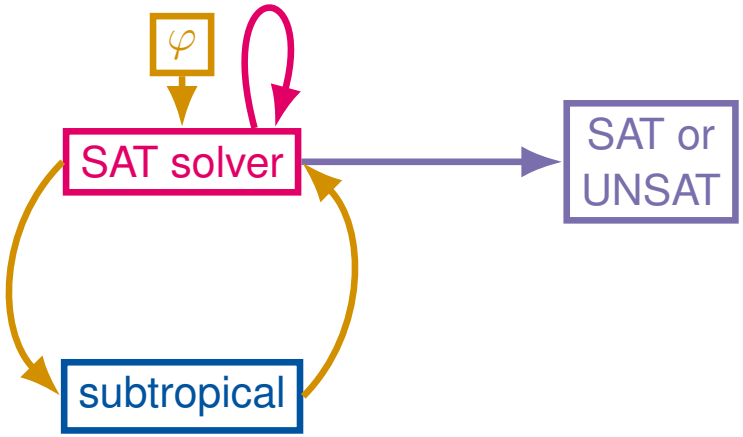
$$\mathcal{ST}_{\varphi}(n) := \mathcal{ST}_{p_1>0}(n) \wedge \mathcal{ST}_{p_2>0}(n)$$

Solutions for conjunctions of inequations [Fontaine et al 2017]

$$\varphi := -7x^2y^2 - 5x^1y^2 + 2x^2y^1 > 0 \wedge -7x^1y^2 - 5y^1 + 2x^1y^1 > 0$$



$$\mathcal{ST}_{\varphi}(n) := \mathcal{ST}_{p_1>0}(n) \wedge \mathcal{ST}_{p_2>0}(n)$$



Generalization of encoding to QFNRA

$$\varphi := -7x^2y^2 - 5x^1y^2 + 2x^2y^1 > 0 \vee -7x^1y^2 - 5y^1 + 2x^1y^1 > 0$$

Generalization of encoding to QFNRA

$$\varphi := -7x^2y^2 - 5x^1y^2 + 2x^2y^1 > 0 \vee -7x^1y^2 - 5y^1 + 2x^1y^1 > 0$$

Idea: Same as for conjunctions, but keep Boolean structure!

Generalization of encoding to QFNRA

$$\varphi := -7x^2y^2 - 5x^1y^2 + 2x^2y^1 > 0 \vee -7x^1y^2 - 5y^1 + 2x^1y^1 > 0$$

Idea: Same as for conjunctions, but keep Boolean structure!

$$\mathcal{ST}_{\varphi}(n) := \mathcal{ST}_{p_1>0}(n) \vee \mathcal{ST}_{p_2>0}(n)$$

Generalization of encoding to QFNRA

$$\varphi := -7x^2y^2 - 5x^1y^2 + 2x^2y^1 > 0 \vee -7x^1y^2 - 5y^1 + 2x^1y^1 > 0$$

Idea: Same as for conjunctions, but keep Boolean structure!

$$\mathcal{ST}_{\varphi}(n) := \mathcal{ST}_{p_1>0}(n) \vee \mathcal{ST}_{p_2>0}(n)$$

Alternative: Use auxiliary Boolean variables.

$$\mathcal{ST}^{aux}_{\varphi}(n) := (a_1 \vee a_2) \wedge (a_1 \to \mathcal{ST}_{p_1>0}(n)) \wedge (a_2 \to \mathcal{ST}_{p_2>0}(n))$$

Experiments

- implementation in SMT-RAT
- QF_NRA benchmark set from SMT-LIB, 12134 instances
- Intel® Xeon® Platinum 8160 2.1GHz
- 2 minutes time limit, 4GB memory limit per instance

Results for pure subtropical solvers

- focus on 3580 non-trivial instances (1735 sat, 499 unsat, 1346 unknown)

	Tr (equation)	$\mathcal{ST}$ enc.	$\mathcal{ST}^{aux}$ enc.	DPLL(T)
solved/sat	16	1399	1398	1399
unknown	2232	782	793	272
timeout	990	1057	1046	1566
memout	342	342	343	343

Results for pure subtropical solvers

- focus on 3580 non-trivial instances (1735 sat, 499 unsat, 1346 unknown)

	Tr (equation)	$\mathcal{ST}$ enc.	$\mathcal{ST}^{aux}$ enc.	DPLL(T)
solved/sat	16	1399	1398	1399
unknown	2232	782	793	272
timeout	990	1057	1046	1566
memout	342	342	343	343

Results for pure subtropical solvers

- focus on 3580 non-trivial instances (1735 sat, 499 unsat, 1346 unknown)

	Tr (equation)	$\mathcal{ST}$ enc.	$\mathcal{ST}^{aux}$ enc.	DPLL(T)
solved/sat	16	1399	1398	1399
unknown	2232	782	793	272
timeout	990	1057	1046	1566
memout	342	342	343	343

Results for pure subtropical solvers

- focus on 3580 non-trivial instances (1735 sat, 499 unsat, 1346 unknown)

	Tr (equation)	$\mathcal{ST}$ enc.	$\mathcal{ST}^{aux}$ enc.	DPLL(T)
solved/sat	16	1399	1398	1399
unknown	2232	782	793	272
timeout	990	1057	1046	1566
memout	342	342	343	343

Results for pure subtropical solvers

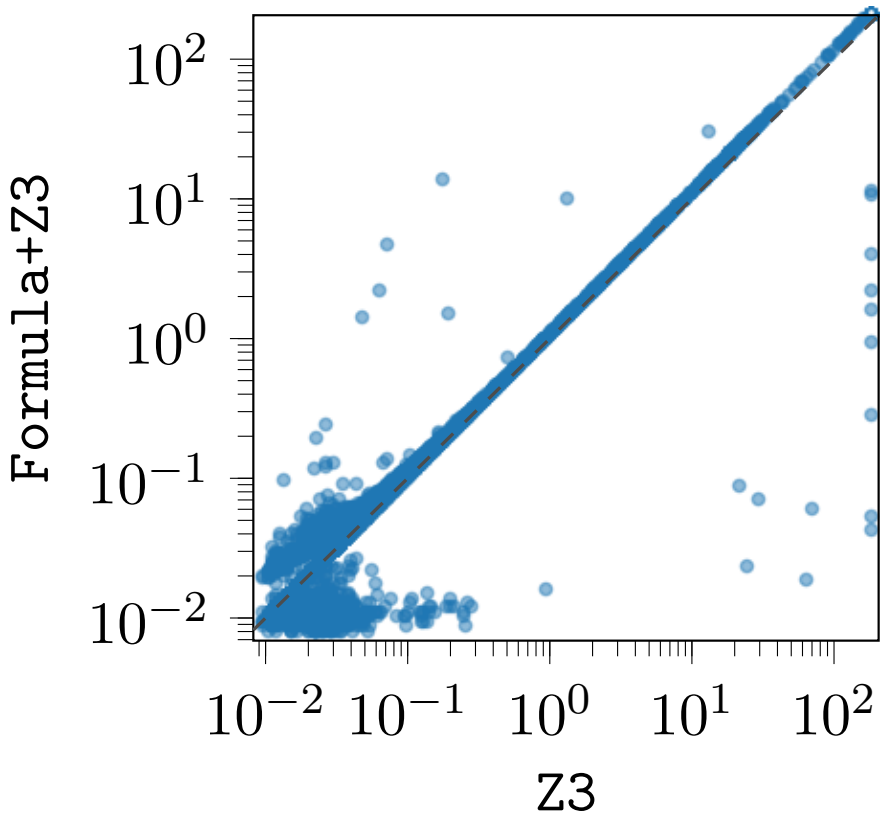
- focus on 3580 non-trivial instances (1735 sat, 499 unsat, 1346 unknown)

	Tr (equation)	$\mathcal{ST}$ enc.	$\mathcal{ST}^{aux}$ enc.	DPLL(T)
solved/sat	16	1399	1398	1399
unknown	2232	782	793	272
timeout	990	1057	1046	1566
memout	342	342	343	343

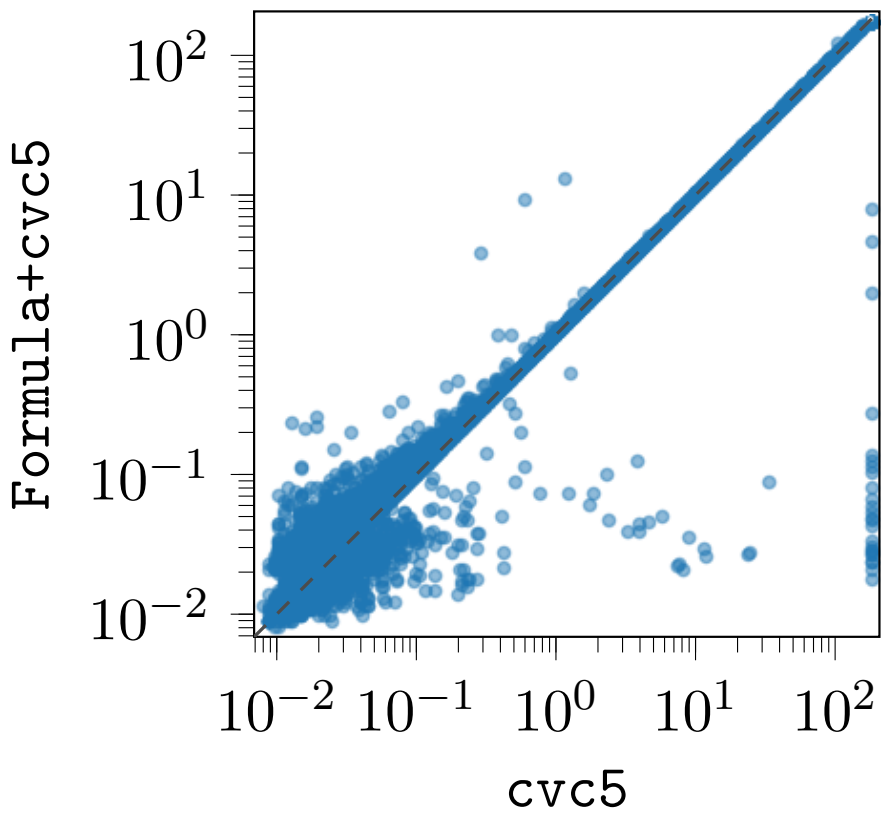
	CAlC	DPLL(T)+CAlC
sat	5055	5071

Combination with z3 and cvc5

- $10s$ for z3/cvc5 on subtropical encoding + $110s$ for z3/cvc5 on original input



9 new benchmarks solved



25 new benchmarks solved

Conclusion

- Our adaption is easy to implement and integrate

Conclusion

- Our adaption is easy to implement and integrate
- and it is fast.

Conclusion

- Our adaption is easy to implement and integrate
- and it is fast.
- In total numbers, our methods solves only few additional instances of SMT-LIB (not solvable before) ...

Conclusion

- Our adaption is easy to implement and integrate
- and it is fast.
- In total numbers, our methods solves only few additional instances of SMT-LIB (not solvable before) ...
- ... however, these instances are hard for state-of-the-art solvers.

Conclusion

- Our adaption is easy to implement and integrate
- and it is fast.
- In total numbers, our methods solves only few additional instances of SMT-LIB (not solvable before) ...
- ... however, these instances are hard for state-of-the-art solvers.
- The SMT-LIB benchmark set is not representative ...

Conclusion

- Our adaption is easy to implement and integrate
- and it is fast.
- In total numbers, our methods solves only few additional instances of SMT-LIB (not solvable before) ...
- ... however, these instances are hard for state-of-the-art solvers.
- The SMT-LIB benchmark set is not representative ...
- ... in particular contains few high-degree polynomials.