Handling Polynomial and Transcendental Functions in SMT via Unconstrained Optimisation and Topological Degree Test

Alessandro Cimatti[1], Alberto Griggio[1], Enrico Lipparini[1,2], Roberto Sebastiani[3]

Fondazione Bruno Kessler, Trento, Italy

DIBRIS, University of Genoa, Italy

DISI, University of Trento, Italy

The 20th International Symposium on Automated Technology for Verification and Analysis

Problem definition

Context: Satisfiability Modulo Theories (SMT)

Theories of interest:

- $\mathcal{NRA}$: Quantifier-free theory of non linear arithmetic over the reals
- $\mathcal{NTA}$: $\mathcal{NRA}$ augmented with exponential and trigonometric functions

Example of formula in $\mathcal{NTA}$:

$$(b_1 \rightarrow (sin(y)^2 = x^3 - 2x^2y + 1 \wedge y > 0.5)) \wedge$$
$$(\neg b_1 \rightarrow (b_2 \rightarrow (y = exp(0.1) \vee x = 0.1)))$$

where b_1, b_2 are Boolean variables and x, y are real-valued individual variables

Problem definition

Context: Satisfiability Modulo Theories (SMT)

Theories of interest:

- $\mathcal{NRA}$: Quantifier-free theory of non linear arithmetic over the reals
- $\mathcal{NTA}$: $\mathcal{NRA}$ augmented with exponential and trigonometric functions

Example of formula in $\mathcal{NTA}$:

$$(b_1 \rightarrow (sin(y)^2 = x^3 - 2x^2y + 1 \wedge y > 0.5)) \wedge$$
$$(\neg b_1 \rightarrow (b_2 \rightarrow (y = exp(0.1) \vee x = 0.1)))$$

where b_1, b_2 are Boolean variables and x, y are real-valued individual variables

Focus on: proving satisfiability for $\mathcal{NTA}$ formulas

Strategy

1. Translate a satisfiability problem into a numerical minimization one (*Logic-To-Optimisation*)
2. Use unconstrained optimisation to generate a set of candidate approximate solutions
3. Given a candidate approximate solution, restrict the original formula to a sub-formula in a specific form (variables are bounded, the number of equations is equal to the number of variables, ...)
4. Apply a procedure based on the topological degree test and interval arithmetic to witness the existence of a solution for the sub-formula

Proving satisfiability in SMT(NTA)

$\mathcal{NTA}$ has been proved undecidable by Richardson (1969).

Existing methods:

- Interval Constrain Propagation (ICP): based on interval arithmetic, can prove "sat" if it finds a box in which every point is a solution
- Incremental Linearization: starts from an abstract model and tries to check whether the formula is satisfiable under all possible interpretations of the transcendental functions involved
- δ-satisfiability: based on ICP, tries to prove that there exists a δ-perturbation of the original formula that is satisfiable

Proving satisfiability in SMT(NTA)

$\mathcal{NTA}$ has been proved undecidable by Richardson (1969).

Existing methods: Interval Constrain Propagation, Incremental Linearization, δ-satisfiability

Proving satisfiability in SMT(NTA)

$\mathcal{NTA}$ has been proved undecidable by Richardson (1969).

Existing methods: Interval Constrain Propagation, Incremental Linearization, δ-satisfiability

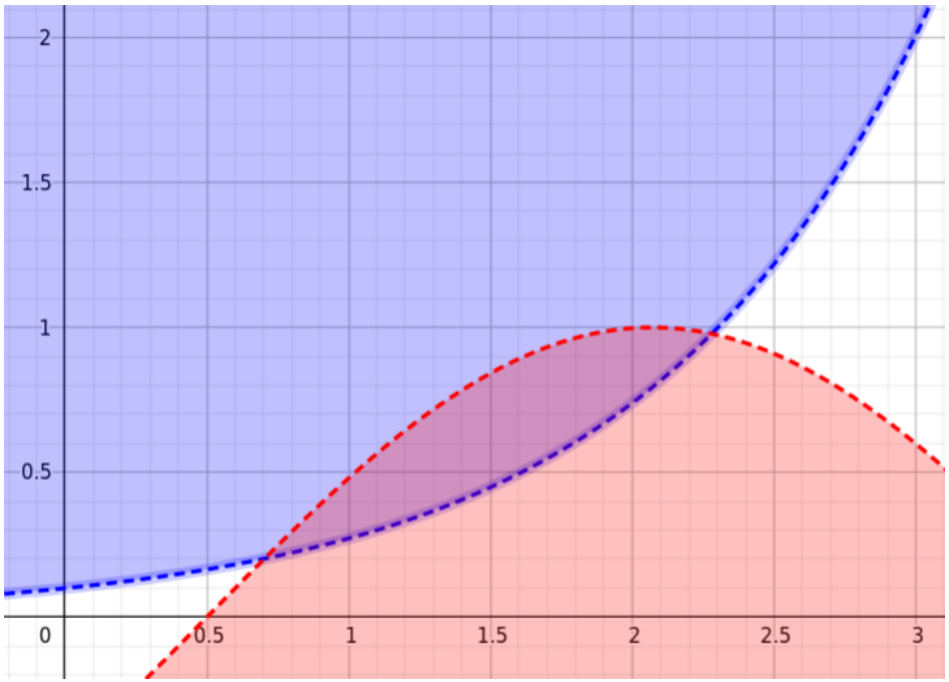

Figure 1: Graph of $(exp(x) < 10y) \wedge (y < sin(x - 1/2)$

Proving satisfiability in SMT(NTA)

$\mathcal{NTA}$ has been proved undecidable by Richardson (1969).

Existing methods: Interval Constrain Propagation, Incremental Linearization, δ-satisfiability

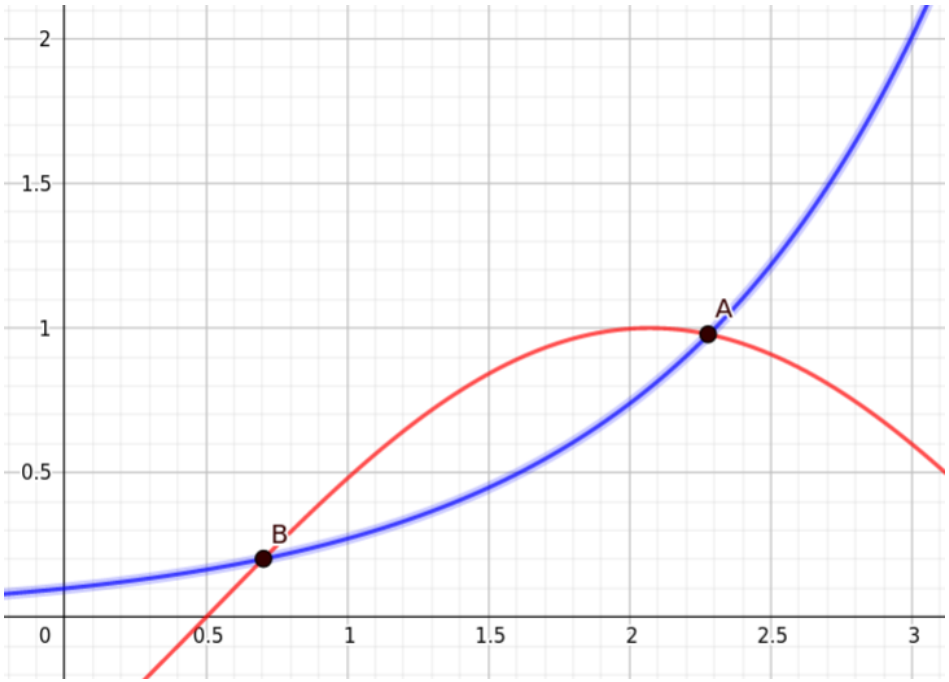

Figure 2: Graph of $(exp(x) = 10y) \wedge (y = sin(x - 1/2)$

Proving satisfiability in SMT(NTA)

$\mathcal{NTA}$ has been proved undecidable by Richardson (1969).

Existing methods: Interval Constrain Propagation, Incremental Linearization, δ-satisfiability

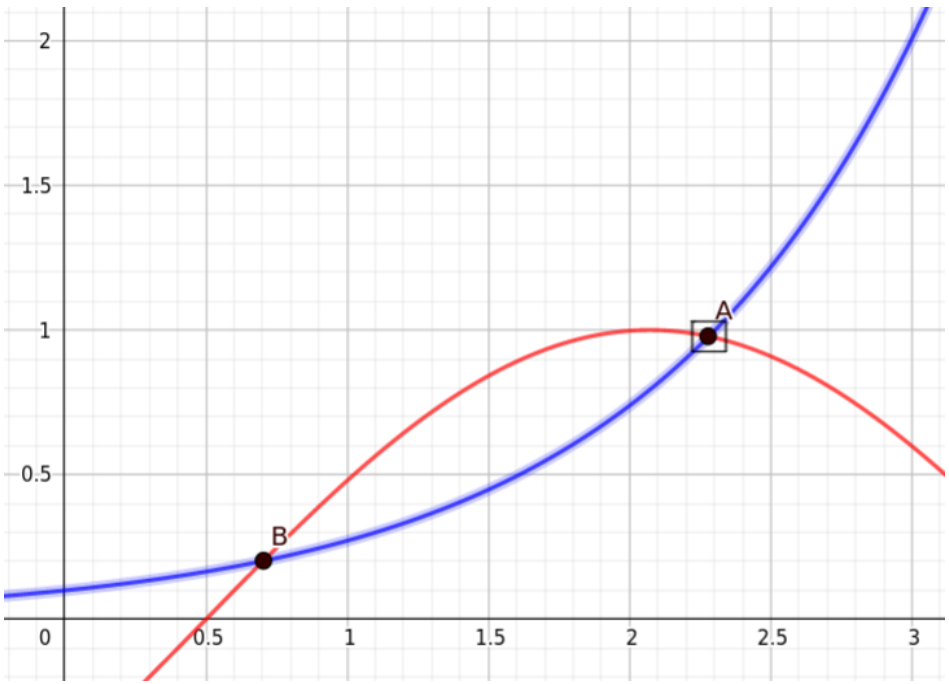

Figure 3: Graph of $(exp(x) = 10y) \wedge (y = sin(x - 1/2)$

Topological degree test

Theorem (Bolzano)

If a continuous function $f : [a, b] \subseteq \mathbb{R} \to \mathbb{R}$ *is s.t.* $f(a) < 0$ *and* $f(b) > 0$ *(or vice-versa), then* f *has a zero in* $[a, b]$.

Proof.

Corollary of the Intermediate Value Theorem □

Topological degree test

Theorem (Bolzano)

If a continuous function $f : [a, b] \subseteq \mathbb{R} \to \mathbb{R}$ *is s.t.* $f(a) < 0$ *and* $f(b) > 0$ *(or vice-versa), then* f *has a zero in* $[a, b]$.

Proof.

Corollary of the Intermediate Value Theorem □

Given a continuous function $f : B \subseteq \mathbb{R}^n \to \mathbb{R}^n$, the *topological degree* $\deg(f, B)$ is an integer that can be defined in several (very technical) ways.

Property (Topological degree test)

If $\deg(f, B) \neq 0$, *then the equation* $f = 0$ *has a solution in* B.

The topological degree is computable if $0 \notin f(\partial B)$ (tool: TopDeg)

Limitations of the topological degree test

The topological degree test is applicable only for conjunctions of constraints in a specific form:

i. there are only equations (no inequalities)
ii. the number of equations is equal to the number of variables
iii. all variables are bounded
iv. (empirical limitation) the bounds on the variables are "sufficiently small" for the practical effectiveness of the test.

Limitations of the topological degree test

The topological degree test is applicable only for conjunctions of constraints in a specific form:

i. there are only equations (no inequalities)
ii. the number of equations is equal to the number of variables
iii. all variables are bounded
iv. (empirical limitation) the bounds on the variables are "sufficiently small" for the practical effectiveness of the test.

Remember the example:

$$(b_1 \rightarrow (sin(y)^2 = x^3 - 2x^2y + 1 \wedge y > 0.5)) \wedge$$
$$(\neg b_1 \rightarrow (b_2 \rightarrow (y = exp(0.1) \vee x = 0.1)))$$

Limitations of the topological degree test

The topological degree test is applicable only for conjunctions of constraints in a specific form:

i. there are only equations (no inequalities)
ii. the number of equations is equal to the number of variables
iii. all variables are bounded
iv. (empirical limitation) the bounds on the variables are "sufficiently small" for the practical effectiveness of the test.

Remember the example:

$$(b_1 \rightarrow (sin(y)^2 = x^3 - 2x^2y + 1 \wedge y > 0.5)) \wedge$$
$$(\neg b_1 \rightarrow (b_2 \rightarrow (y = exp(0.1) \vee x = 0.1)))$$

Limitation (i) has been tackled by Franek, Ratschan, and Zgliczynski (2016), who proposed a quasi-decidability (i.e. that always terminates on robust instances) procedure that pairs the topological degree test with interval arithmetic to deal with inequalities.

Method

Can we use topological degree test, given a general formula with m variables, n equations, k inequalities?

Plan:

- Find candidate solutions through unconstrained optimisation
- Bound the variables in the formula to a box
- If $n < m$, try to instantiate $m - n$ variables
- Apply a procedure that uses interval arithmetic and the topological degree test

First, we work only with conjunctions of constraints. Then, we extend to full Boolean combinations.

Logic-to-Optimisation

Idea: try to find candidate solutions by translating a satisfiability problem into a numerical minimization problem (*Logic-To-Optimisation*)

We define an operator $\mathcal{L}2\mathcal{O}$ that maps a formula in m variables to a non-negative real function from $\mathbb{R}^m$ to $\mathbb{R}^{\geqslant 0}$ as follows:

$$\mathcal{L}2\mathcal{O}(f \bowtie 0) : \mathbf{x} \mapsto \begin{cases} 0 & \text{if } \bowtie \in \{\leqslant, =\} \text{ and } [f](\mathbf{x}) \bowtie 0 \\ \mathcal{L}2\mathcal{O}(f \leqslant 0) & \text{if } \bowtie \text{ is } < \\ [f]^2(\mathbf{x}) & \text{otherwise} \end{cases}$$

$$\mathcal{L}2\mathcal{O}(b) \equiv \mathcal{L}2\mathcal{O}(-x_b \leqslant 0)$$

$$\mathcal{L}2\mathcal{O}(\neg b) \equiv \mathcal{L}2\mathcal{O}(x_b + \epsilon \leqslant 0)$$

$$\mathcal{L}2\mathcal{O}(\phi_1 \wedge \phi_2) \equiv \mathcal{L}2\mathcal{O}(\phi_1) + \mathcal{L}2\mathcal{O}(\phi_2)$$

$$\mathcal{L}2\mathcal{O}(\phi_1 \vee \phi_2) \equiv \mathcal{L}2\mathcal{O}(\phi_1) * \mathcal{L}2\mathcal{O}(\phi_2),$$

Property: every model of ϕ is a zero of $\mathcal{L}2\mathcal{O}(\phi)$ (but not the vice-versa)

Logic-to-Optimisation

Example: $\phi \equiv x^3 - 2x - exp(x) + 5 = 0$

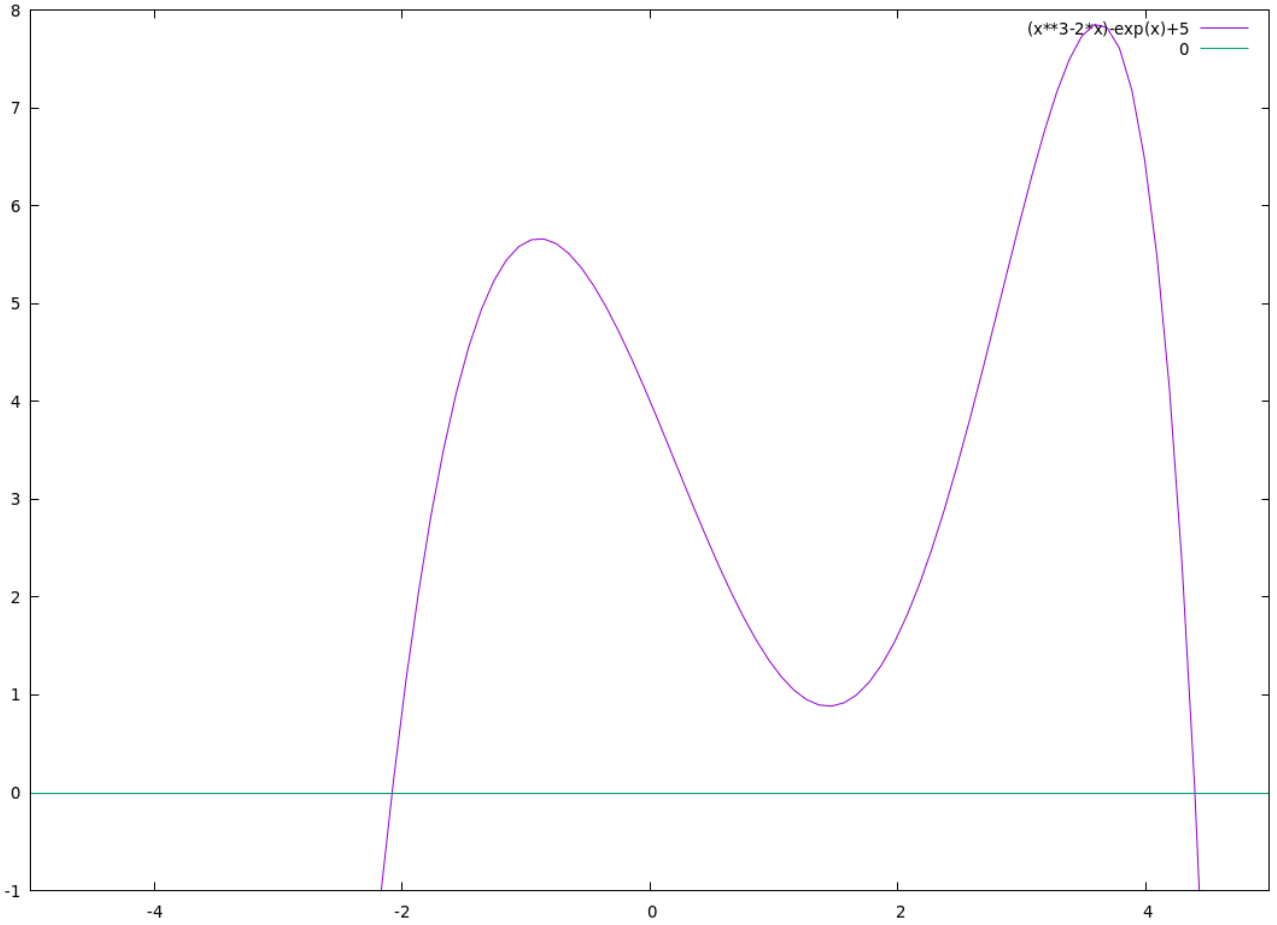

Figure 4: Graph of $x^3 - 2x - exp(x) + 5$

Logic-to-Optimisation

Example: $\phi \equiv x^3 - 2x - exp(x) + 5 = 0$

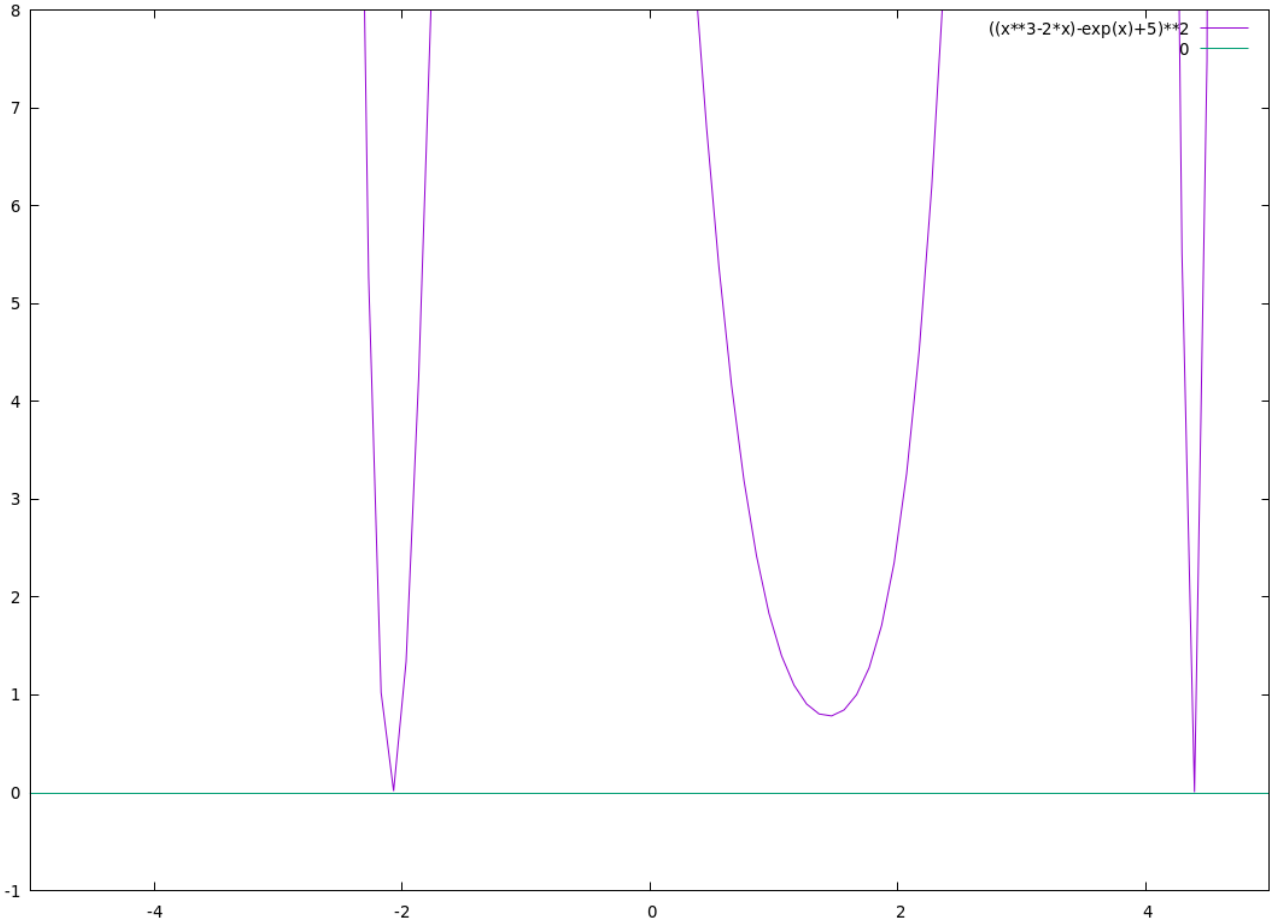

Figure 5: Graph of $\mathcal{L}2\mathcal{O}(\phi)$

Local search using L2O

$\phi \rightarrow \mathcal{L}2\mathcal{O}$ + Unconstr. Opt. (basin-hopping) $\rightarrow$ several local minima $\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_k$

Given a local minimum $\tilde{\mathbf{x}}$, we have 3 tactics:

1. Check if it is a solution (just for $\mathcal{NRA}$)
2. Reduce to a linear under-approximation in UFLRA by forcing all the multiplications to be linear (just for $\mathcal{NRA}$)
3. - Restrict to a sub-formula $\phi_{|B}$ by imposing that all the variables range over a box B containing $\tilde{\mathbf{x}}$.
   - Instantiate exceeding variables.
   - Apply the quasi-decidability procedure

Quasi-decidability procedure (Idea)

Input: A formula $\phi_{|B}$ in m variables, n equations, and k inequalities$_{\leqslant}$ s.t. $n = m$ or $n = 0$.

Loop: We start with a grid that initially contains only B. At each iteration, we split each box of the grid into sub-boxes and:

1. Remove sub-boxes that cannot contain a solution (using interval arithmetic)
2. Try to find a sub-box in which the inequalities are satisfied everywhere and the topological degree test witnesses the existence of a solution for the equations

Output:

- **False** if at some point the grid is emptied
- **True** if at some point test 2 of the Loop succeeds

A general procedure

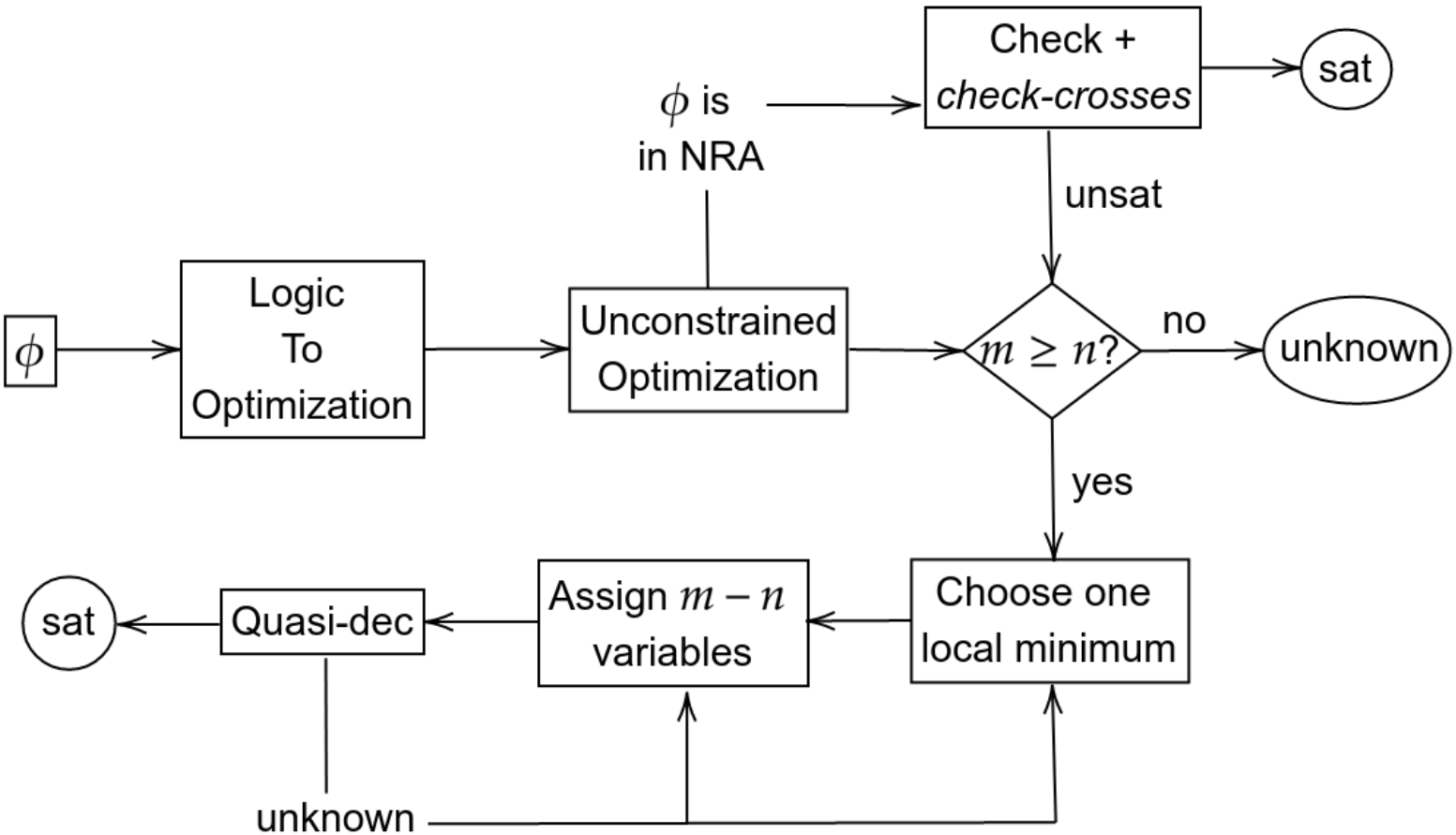

Figure 6: General procedure of UGOT (as in *Unconstrained Global Optimisation and Topological degree*)

From constraints set to formulas

Let ϕ in CNF. Two approaches:

- *Eager* approach:
  - $\mathcal{L}2\mathcal{O}(\phi) \rightarrow$ Unconstr. Opt. $\rightarrow$ several local minima $\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_k$
  - Given $\tilde{\mathbf{x}}$, we try to *decide* the disjunctions through the insight given by $\tilde{\mathbf{x}}$
    Example: $\tilde{\mathbf{x}} = 1$, we replace in ϕ the clause "$x < 3 \vee x > 5$" with "$x < 3$"
  - Apply UGOT

From constraints set to formulas

Let ϕ in CNF. Two approaches:

- *Eager* approach:
  - $\mathcal{L}2\mathcal{O}(\phi) \rightarrow$ Unconstr. Opt. $\rightarrow$ several local minima $\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_k$
  - Given $\tilde{\mathbf{x}}$, we try to *decide* the disjunctions through the insight given by $\tilde{\mathbf{x}}$
    Example: $\tilde{\mathbf{x}} = 1$, we replace in ϕ the clause "$x < 3 \vee x > 5$" with "$x < 3$"
  - Apply UGOT
- *Lazy* approach:
  - Integration of UGOT as a theory solver inside the DPLL(T)-based SMT-solver MATHSAT
  - Pair UGOT (good for proving sat) with Incremental Linearization (good for proving unsat)
  - Naive implementation. For each full Boolean assignment:
    - We call Incremental Linearization at most k times.
    - If it does not find a model or produce a conflict, we call UGOT

Experiments

	Total	Sturm-MGC	ezsmt	Sturm-MBO	zankl	UltimateAut	Economics-M	meti-tarski	Heizmann	hycomp	kissing	LassoRanker
MATHSAT	3193	0	**32**	0	32	31	85	2718	3	11	18	263
UGOT$_{\text{EAGER}}$	4388	0	0	**1**	52	32	68	4042	0	0	**36**	157
MATHSAT+UGOT	4441	0	**32**	0	63	27	84	3948	3	13	35	236
RASAT	4285	0	0	0	45	0	0	4225	0	0	15	0
YICES	4946	0	**32**	0	58	39	91	4369	0	227	10	120
CVC5	5108	0	**32**	0	63	36	89	4342	2	226	18	**300**
Z3	**5153**	**2**	30	0	**72**	**47**	**93**	**4391**	**6**	**280**	35	198
DREAL	/(5021)	/(9)	/(0)	/(274)	/(153)	/(55)	/(126)	/(4079)	/(51)	/(45)	/(19)	/(210)

Fig. 2. Summary of results for SMT(NRA) sat cases. The results in parenthesis indicate "MAYBE SAT" answers.

	Total	dreal	bmc
MATHSAT	94	39	57
UGOT$_{\text{EAGER}}$	**304**	**255**	49
MATHSAT+UGOT	170	70	**100**
CVC5	94	40	54
ISAT3	/	/	/
DREAL	/ (578)	/ (423)	/ (155)

	Total	dreal	bmc
MATHSAT	70	21	49
UGOT$_{\text{EAGER}}$	**253**	**203**	50
MATHSAT+UGOT	140	35	**105**
CVC5	63	17	46
ISAT3	38 (828)	7 (599)	31 (229)
DREAL	/ (137)	/ (36)	/ (101)

Fig. 3. Summary of results for SMT(NTA) sat cases. On the left the original instances; on the right the bounded instances.The results in parenthesis indicate "MAYBE SAT" answers.

Experiments (and what about unsat cases?)

	Total	Sturm-MGC	ezsmt	Sturm-MBO	zankl	UltimateAut	Economics-M	meti-tarski	Heizmann	hycomp	kissing	LassoRanker	hong
MathSAT	5280	0	**2**	**285**	**34**	7	11	2251	1	**2259**	0	412	**20**
MathSAT+ugot	5043	0	2	163	32	5	11	2239	0	2253	0	323	**20**
raSAT	4094	0	0	2	14	0	0	2018	0	1950	0	0	**20**
Yices	5449	0	**2**	**285**	32	**12**	**39**	**2587**	**12**	2201	0	259	**20**
cvc5	**5645**	0	**2**	**285**	31	10	35	2581	7	2206	0	**468**	**20**
z3	5281	**5**	**2**	153	27	**12**	19	2578	3	2225	0	248	9
dReal	3889	0	0	131	4	0	3	1784	0	1946	**1**	0	**20**

Fig. 4. Summary of results for SMT(NRA) unsat cases.

	Total	dreal	bmc
MathSAT	**533**	85	**448**
MathSAT+ugot	522	85	437
cvc5	453	75	378
iSAT3	/	/	/
dReal	468	**184**	284

	Total	dreal	bmc
MathSAT	**524**	88	**436**
MathSAT+ugot	521	88	433
cvc5	465	85	380
iSAT3	449	63	386
dReal	446	**156**	290

Fig. 5. Summary of results for SMT(NTA) unsat cases. On the left the original instances; on the right the bounded instances.

Conclusion and Discussion

Recap:

- Unconstrained optimisation to find candidate solutions
- Quasi-dec procedure with topological degree test to check if a region contains a solution
- Eager approach: candidate solution insight used to decide disjunctions
- Lazy approach: pair the new method with incremental linearization

Open questions:

- How to choose which candidate solution to test? Which variables to instantiate? How to determine the box?
- Extend to unsat (generate lemmas / conflict clauses)?
- Extend to solve problems involving differential equations?

Thank you

Thank you for your attention!

Questions?