raSAT: An SMT Solver for Polynomial Constraints

Vu Xuan Tung, Mizuhito Ogawa
(Japan Advanced Institute of Science and Technology, Japan)
To Van Khanh
(University of Engineering and Technology, Vietnam)

IJCAR 2016, 30th June 2016

Download: http://www.jaist.ac.jp/~s1310007/raSAT/, or google “raSAT SMT”

Solving Polynomial Constraints

- Satisfiability of : $\varphi ::= g(x_1, \ldots, x_n) \diamond 0 \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \neg\varphi$

  where $\diamond \in \{>, \geq, <, \leq, =, \neq\}$ and $g(x_1, \ldots, x_n)$ is a polynomial.

- Example: $(x^3y - y^4 + 1 > 0) \wedge (y^3 - xy > 0)$ is satisfiable with

  $x = 2.652, y = 2.346$

- Notion: UNSAT for Unsatisfiability, SAT for Satisfiability

Applications: Invariants Generation, Round-off and Over-flow Error Analysis, Automatic Termination Prover for Term Rewriting Systems.

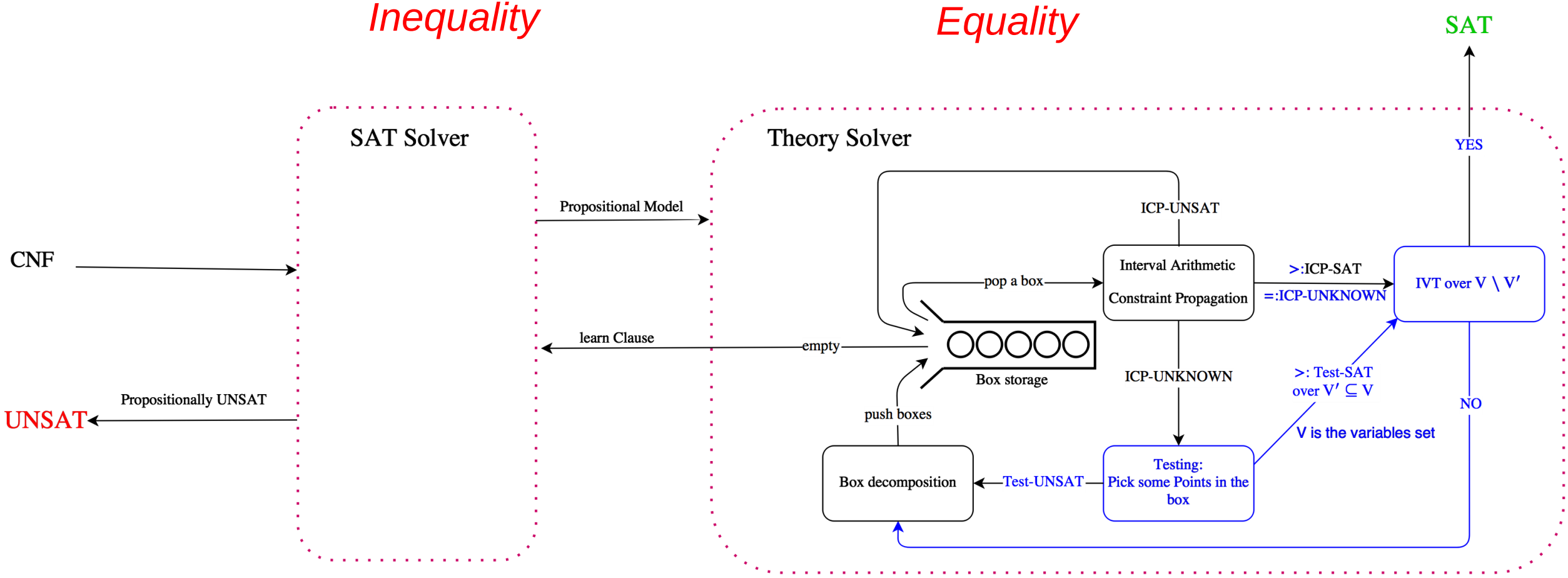
Methods for Solving Polynomial Constraints

- CAD: **complete** for **general** quantified formulas, DEXP
  - ✓ Z3, SMT-RAT, QEPCAD, Redlog
- Virtual substitution: degree < 4, EXP
  - ✓ SMT-RAT, Redlog
- Gröbner basis: Equalities, DEXP
  - ✓ SMT-RAT, Mathematica, Maple, Reduce
- Interval Constraint Propagation (ICP): Inequalities, incomplete
  - ✓ iSAT3, dReal, raSAT
- Bit-blasting: Bounded variables and precision
  - ✓ miniSmt
- Linearization using CORDIC: Bounded variables and precision
  - ✓ CORD

raSAT – an SMT Solver for Polynomial Constraints

- raSAT: ICP + Testing + Intermediate Value Theorem (IVT).

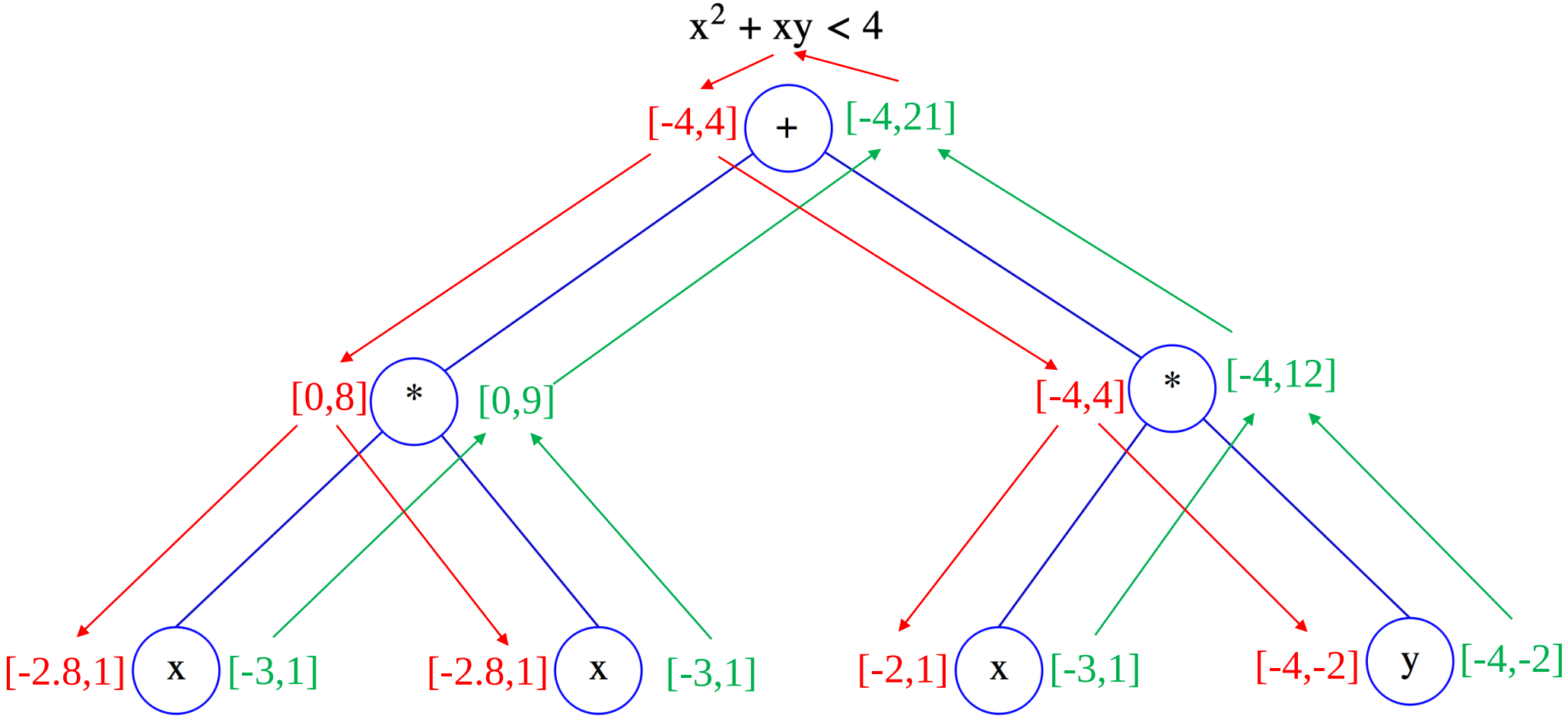
Inequality *Equality*



ICP = Interval arithmetic + Constraint propagation + Box decomposition

Interval Arithmetic and Constraint Propagation

- E.g., $x^2 + xy < 4$ $x \in [-3,1]$, $y \in [-4,-2]$

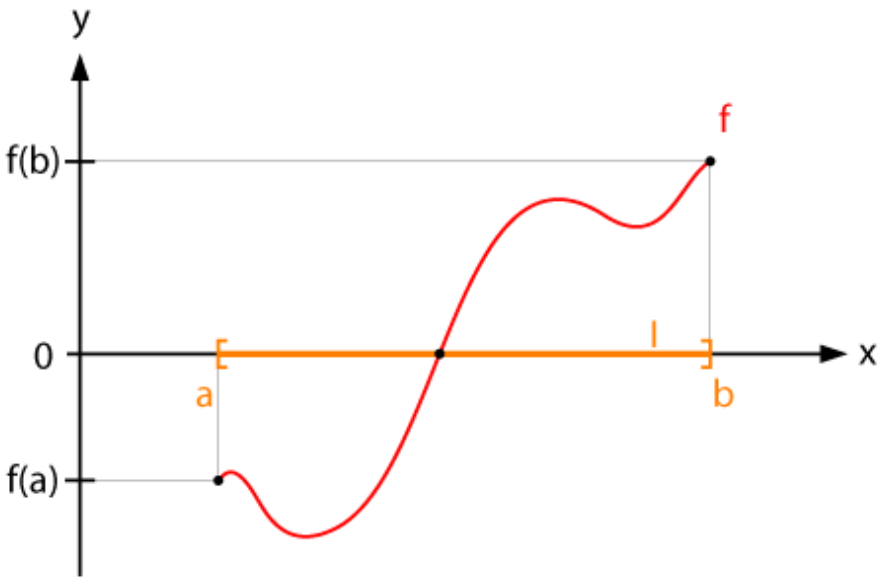


Input: x ∈[-3,1], y ∈[-4,-2]

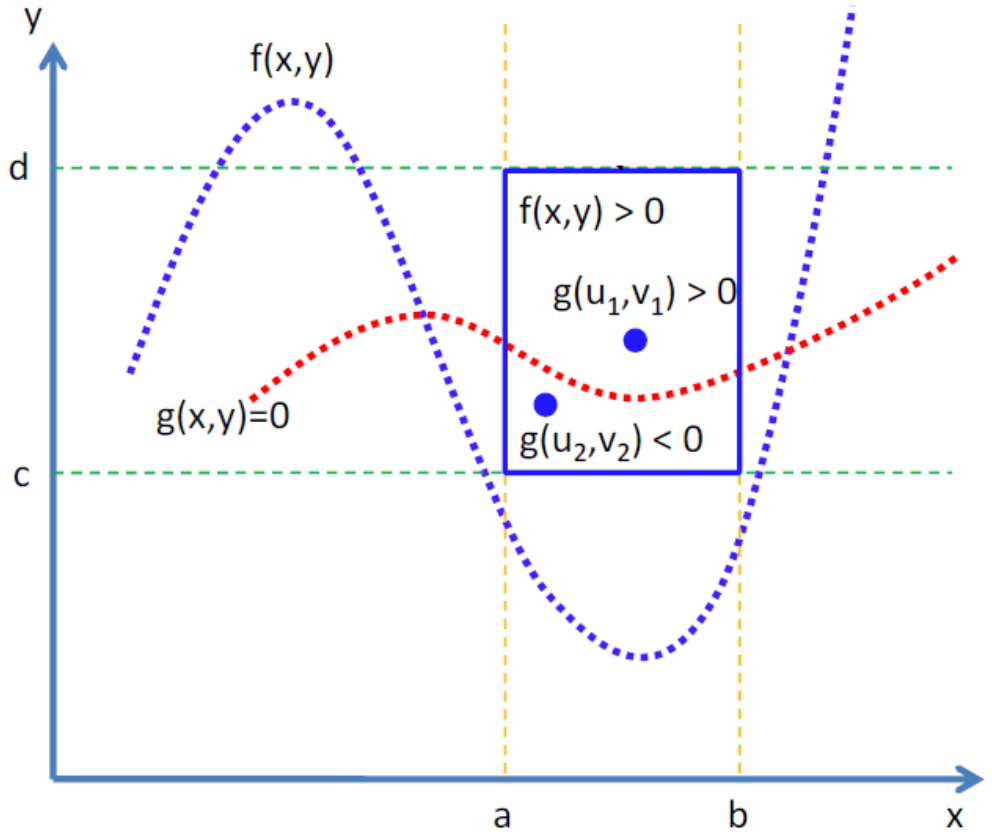
Output: ICP-UNKNOWN, x ∈[-2,1], y ∈[-4,-2]

IVT for a single equation

IVT
when $f(x)$ is continuous



SAT detection by IVT
$f(x,y) > 0 \wedge g(x,y) = 0$



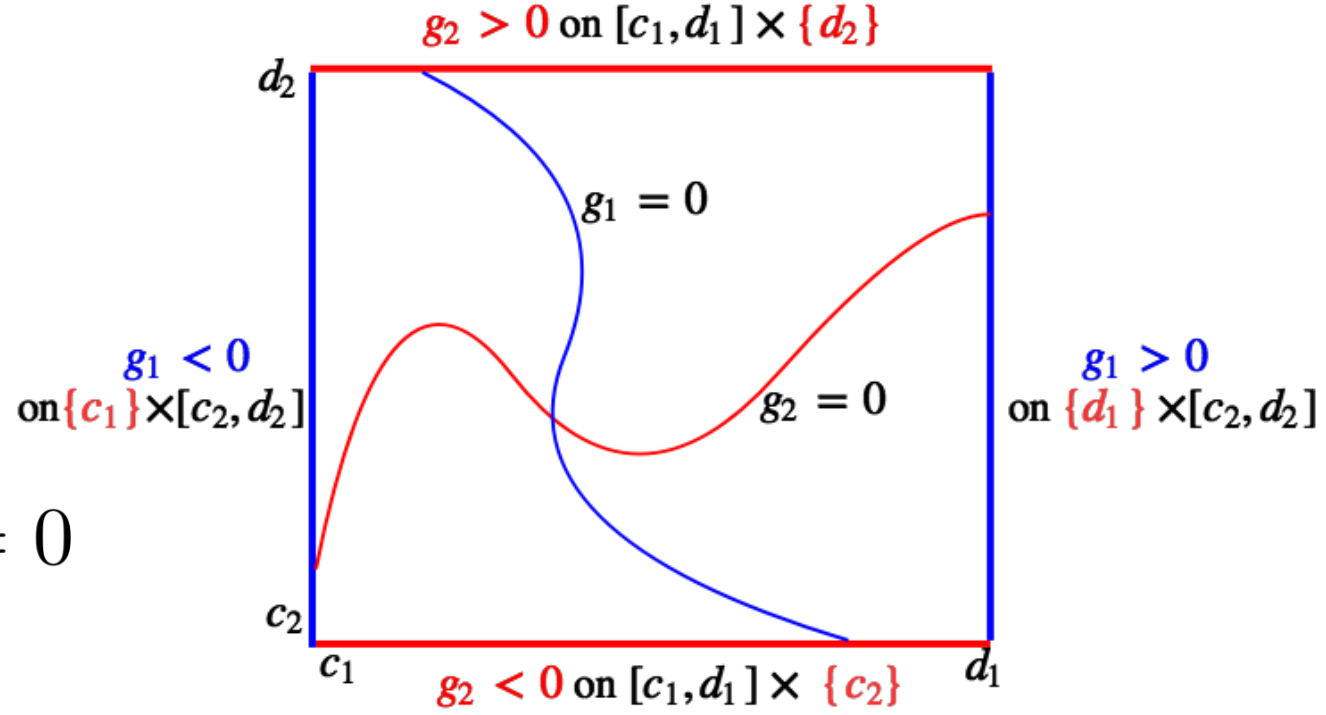
Generalized IVT for Multiple Equations

- The Generalized IVT†
  - ✓ Multiple equations
  - ✓ Requires
    |Variables| ≧ |Equations|

- Example:

$$g_1(x, y) = 0 \wedge g_2(x, y) = 0$$

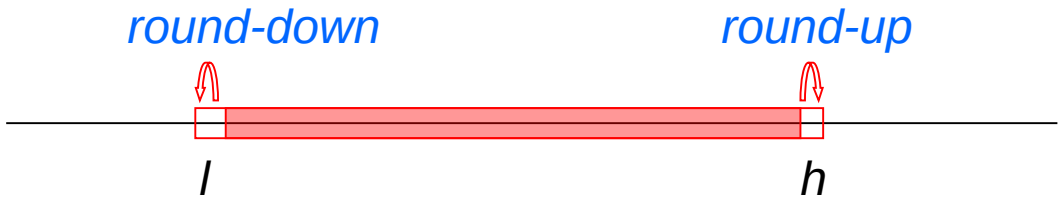
$$x \in [c_1, d_1], y \in [c_2, d_2]$$



†Neumaier, A.: Interval Methods for Systems of Equations. Cambridge Middle East Library, Cambridge University Press (1990)

raSAT is Sound

- **Soundness** under floating point arithmetic
  - ✓ Outward rounding in Interval Arithmetic (library of Alliot et al.†)



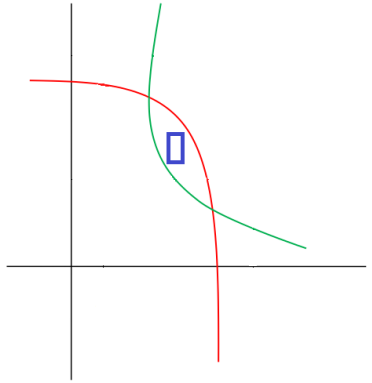
  - ✓ Confirming SAT instance by iRRAM, an guaranteed round-off error bound package: http://irram.uni-trier.de/

- Easily extended to constraints over Integers (NIA) by picking only integers as test data.

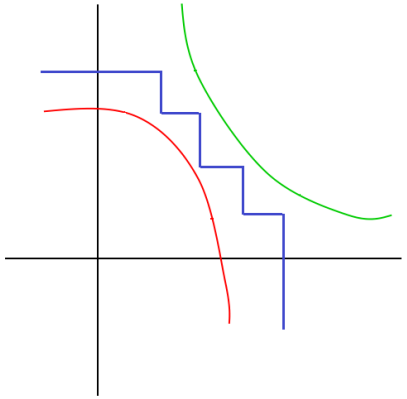
† Alliot, J.M., Gotteland, J.B., Vanaret, C., Durand, N., Gianazza, D.: Implementing an interval computation library for OCaml on x86/amd64 architectures. In: ICFP. ACM (2012)

raSAT is Incomplete

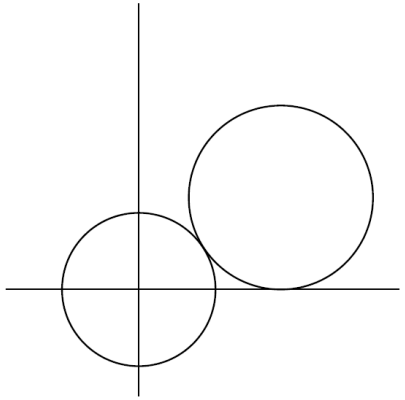
- Inequality



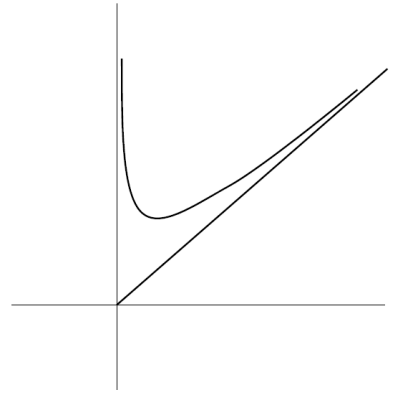
✓ SAT Detection



✓ UNSAT Detection



✗ Kissing case



✗ Convergence

- Equality
  - ✓Generalized IVT requires |Variables| ≧ |Equations|
  - ✓Potentially 0-dim ideal case when |Variables| < |Equations|

SMT-COMP results in QF_NRA and QF_NIA

- 2014: raSAT 0.1 (QF_NRA) solved **88** problems, 3rd among 3.
- 2015: raSAT 0.2 (QF_NRA) solved **7952** problems, 3rd among 6.
  (QF_NIA) solved **7917** problems, 2nd among 7.
- 2016: raSAT 0.3/0.4 preliminary results on 29th June, 2016

QF_NRA	Z3-4.4.1	Yices-2.4.2	SMT-RAT	raSAT 0.4	raSAT 0.3	CVC4
Solved No.	10056	10019	9026 (4 errors)	9024	8431	2694
Time (sec)	24785.38	61989.88	51053.15	11176.39	13576.52	150.24

QF_NIA	Z3-4.4.1	Yices-2.4.2	SMT-RAT	AProVE	CVC4	raSAT 0.4	ProB	raSAT 0.3
Solved No.	8566	8451	8443	8273	8231	8017	7557	7544
Time (sec)	27718.2	8523.4	6234.5	8527.66	161418.04	159247.55	13586.05	70228.9

https://www.starexec.org/starexec/secure/details/job.jsp?id=16079,16070

Observations from experiments

- SAT is detected on several large constraints solely by raSAT:
  (in comparison with dReal, iSAT3, SMT-RAT, Z3)
  - ✓ zankl/matrix-2-all-3.smt2: 57 variables
  - ✓ zankl/matrix-2-all-8.smt2: 17variables
  - ✓ zankl/matrix-3-all-5.smt2: 81variables
  - ✓ zankl/matrix-4-all-3.smt2: 139 variables
  - ✓ zankl/matrix-4-all-9.smt2: 193variables

  where Z3 4.4 solely solves many, among them some large ones are
  - ✓ zankl/matrix-3-all-7.smt2: 75 variables
  - ✓ zankl/matrix-4-all-12.smt2: 200variables
  - ✓ zankl/matrix-5-all-6.smt2: 258variables
- **Not** much good in UNSAT detection → needs to improve UNSAT core.
- Adding IVT to raSAT increased **1330** SAT detection in benchmarks.

Future Work: Make ICP complete by Algebraic methods

- Make the procedure complete: ICP + CAD
  - ✓ Idea from Loup et al.†
  - ✓ Call CAD when **a box** in ICP becomes **small**
  - ✓ The small box of ICP guides CAD to reduce the numbers of polynomials / cells in Projection / Lifting phases.
- Improve UNSAT detection: When ICP-UNSAT or Test-UNSAT, reduce the search space by clause learning.
  - ✓ Idea from Jovanovic et al.‡
  - ✓ Project only polynomials in ICP(Test)-UNSAT constraints
  - ✓ Compute only the cell that contains the UNSAT box or UNSAT test case
  - ✓ Learn the negation of the constraint representing such a cell

† Loup, U., et.al.: A symbiosis of interval constraint propagation and cylindrical algebraic decomposition. In: CADE. LNAI 7898, pp.193–207 (2013)

‡ Jovanovic, D., de Moura, L.M.: Solving non-linear arithmetic. In: IJCAR, pp.339–354 (2012)