Advances in Automated Theorem Proving

Leonardo de Moura, Nikolaj Bjørner
Ken McMillan, Margus Veanes

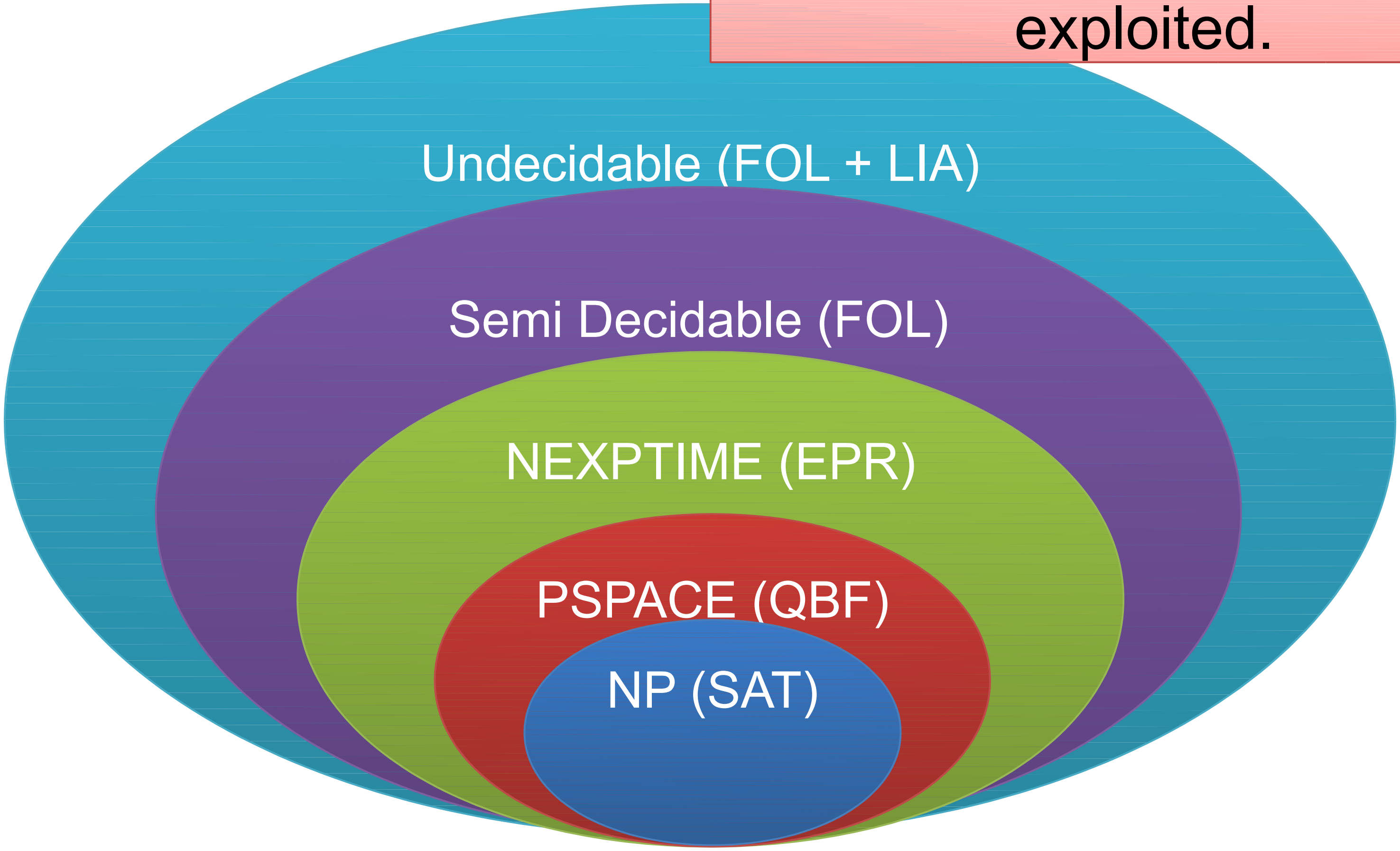
presented by
Thomas Ball

http://research.microsoft.com/rise/
http://rise4fun.com/z3py/

Symbolic Reasoning

Logic is “The Calculus of Computer Science” Zohar Manna

Practical problems often have **structure** that can be exploited.



Undecidable (FOL + LIA)

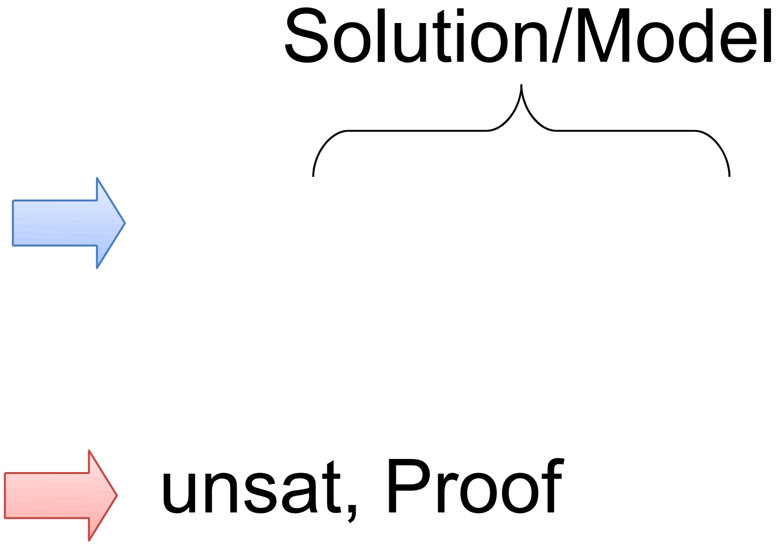
Semi Decidable (FOL)

NEXPTIME (EPR)

PSPACE (QBF)

NP (SAT)

Satisfiability



Solution/Model

unsat, Proof

Automated Theorem Provier

http://research.microsoft.com/projects/z3/

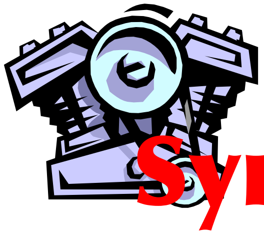
Leonardo de Moura and Nikolaj Bjørner

DPLL

Simplex

Rewriting

Superposition



Z3 is a collection of

Symbolic Reasoning Engines

Congruence Closure

Groebner Basis

elimination

Euclidean Solver

Learn about Z3 and get the source code!

Start here

http://rise4fun.com/Z3Py/tutorial/guide

Strategies

http://rise4fun.com/Z3Py/tutorial/strategies

Advanced topics

http://rise4fun.com/Z3Py/tutorial/advanced

Source code

http://z3.codeplex.com/

Some Applications

Functional verification

Defect detection

Test generation

Design-space exploration

New programming languages

Impact

Z3 used by many research groups (> 700 citations)

More than 17k downloads

Z3 placed 1st in 17/21 categories in 2011 SMT competition

Design & PL

FORMULA

Modeling Foundations.

F

Verification/Defect Detection

SLAM

Vcc

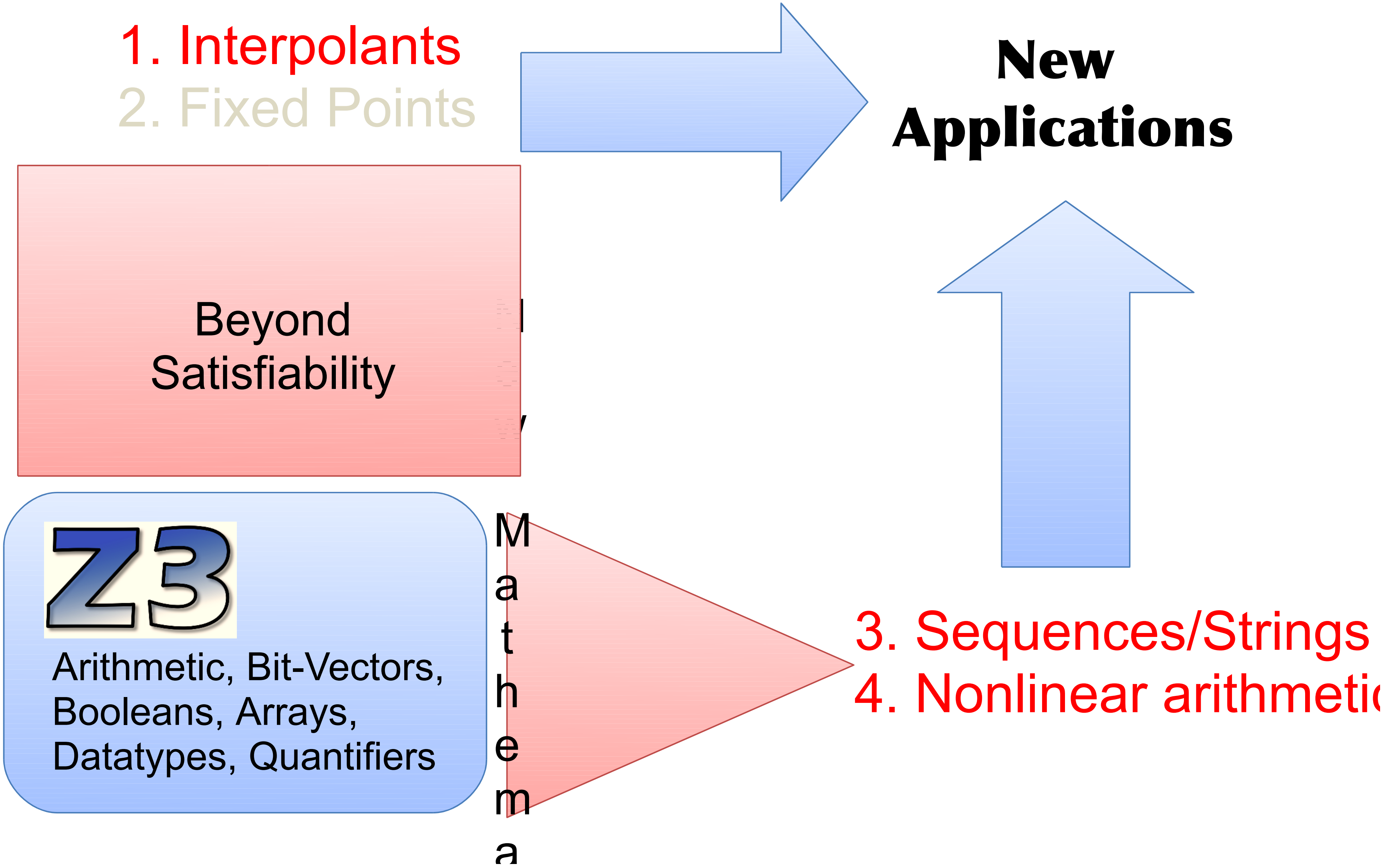
BOOGIE

Testing

Pex

SAGE

Recent Progress



1. Interpolants
2. Fixed Points

New Applications

Beyond Satisfiability

Arithmetic, Bit-Vectors, Booleans, Arrays, Datatypes, Quantifiers

Mathema

3. Sequences/Strings
4. Nonlinear arithmeti

Craig Interpolation and Interpolating Z3

Ken McMillan

(FMCAD 2011)

Introduction

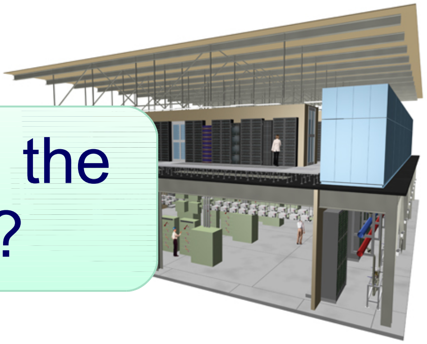
Imagine two companies that want to do business...

Click to edit Master text styles

Second level

- Third level
  - Fourth level
    - Fifth level

How do we explain the problem to Bob?



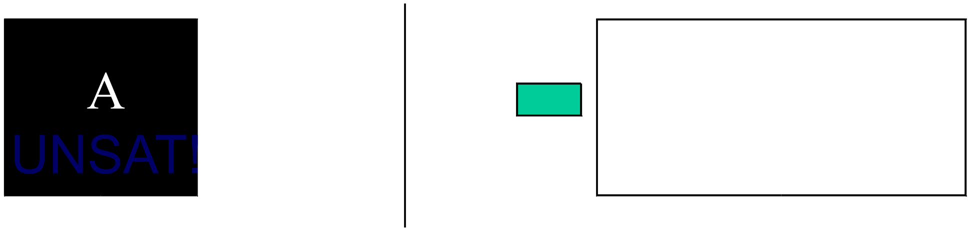
Alice's Business Machines

Bob's Good Hosting

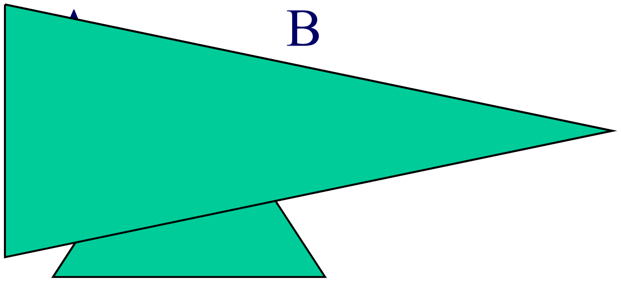
Constraints → UNSAT ← Constraints

Interpolants as Explanations

unknown,
complex



Proof



feasible interpolation

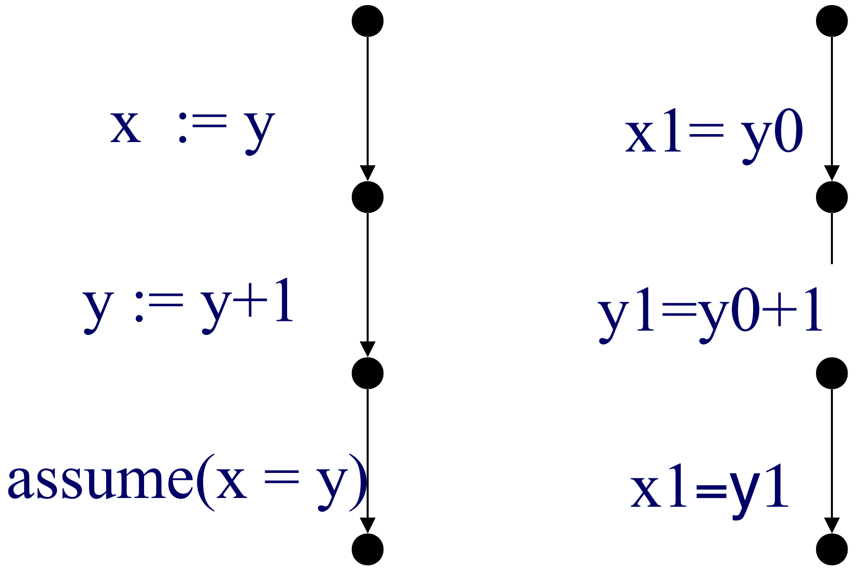
most general

I ie
f
known variables.

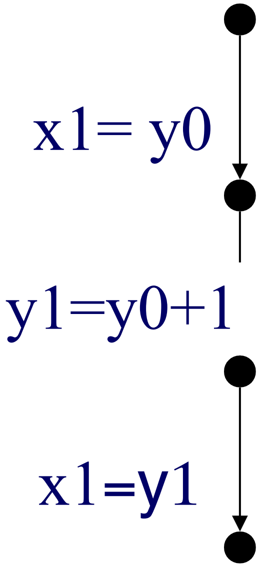
RELEVANT
GENERALIZATION

most specific

Interpolants as Floyd-Hoare proofs



Interpolants as Floyd-Hoare proofs

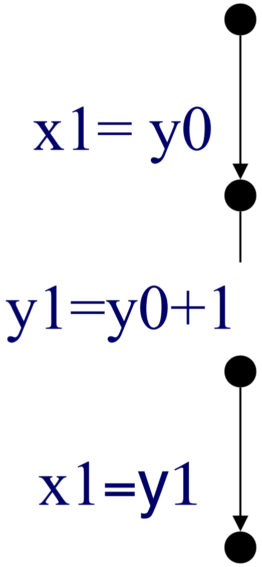


$x1= y0$

$y1=y0+1$

$x1=y1$

Interpolants as Floyd-Hoare proofs

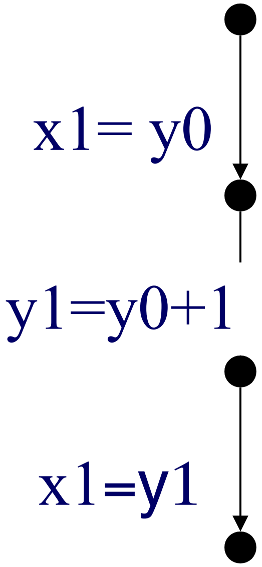


x1= y0

y1=y0+1

x1=y1

Interpolants as Floyd-Hoare proofs

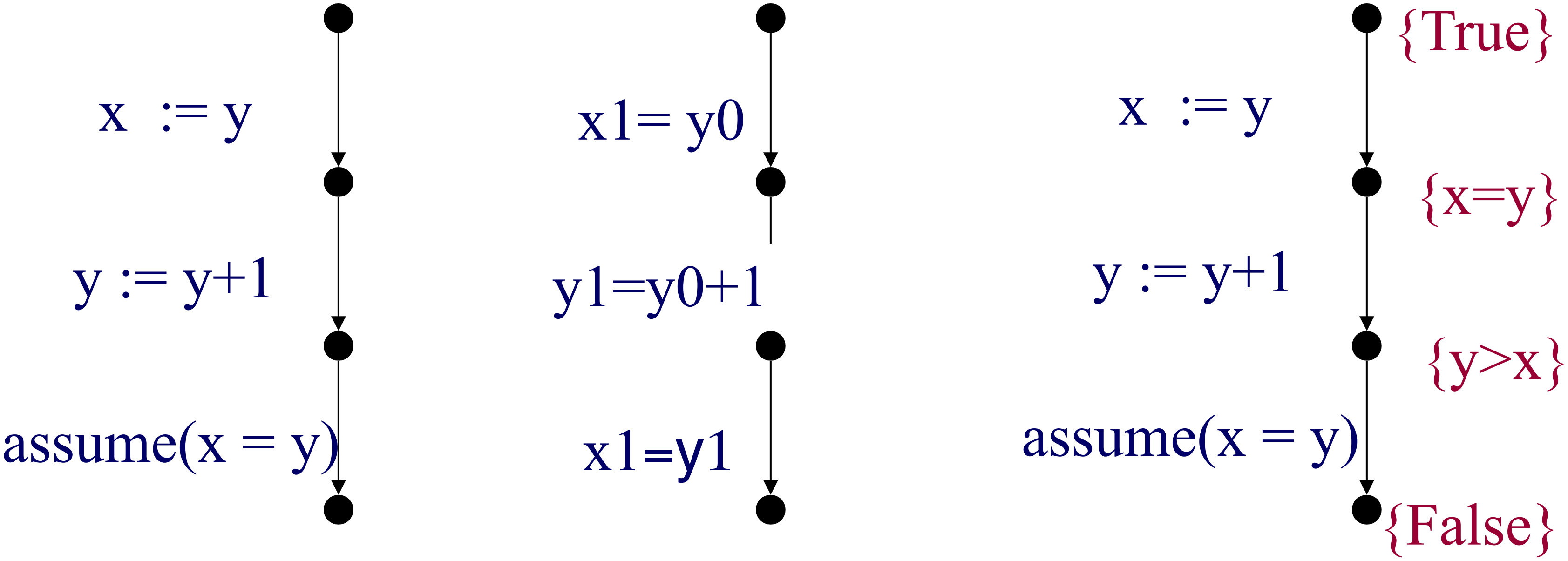


$x1= y0$

$y1=y0+1$

$x1=y1$

Interpolants as Floyd-Hoare proofs



x := y

y := y+1

assume(x = y)

x1= y0

y1=y0+1

x1=y1

x := y

y := y+1

assume(x = y)

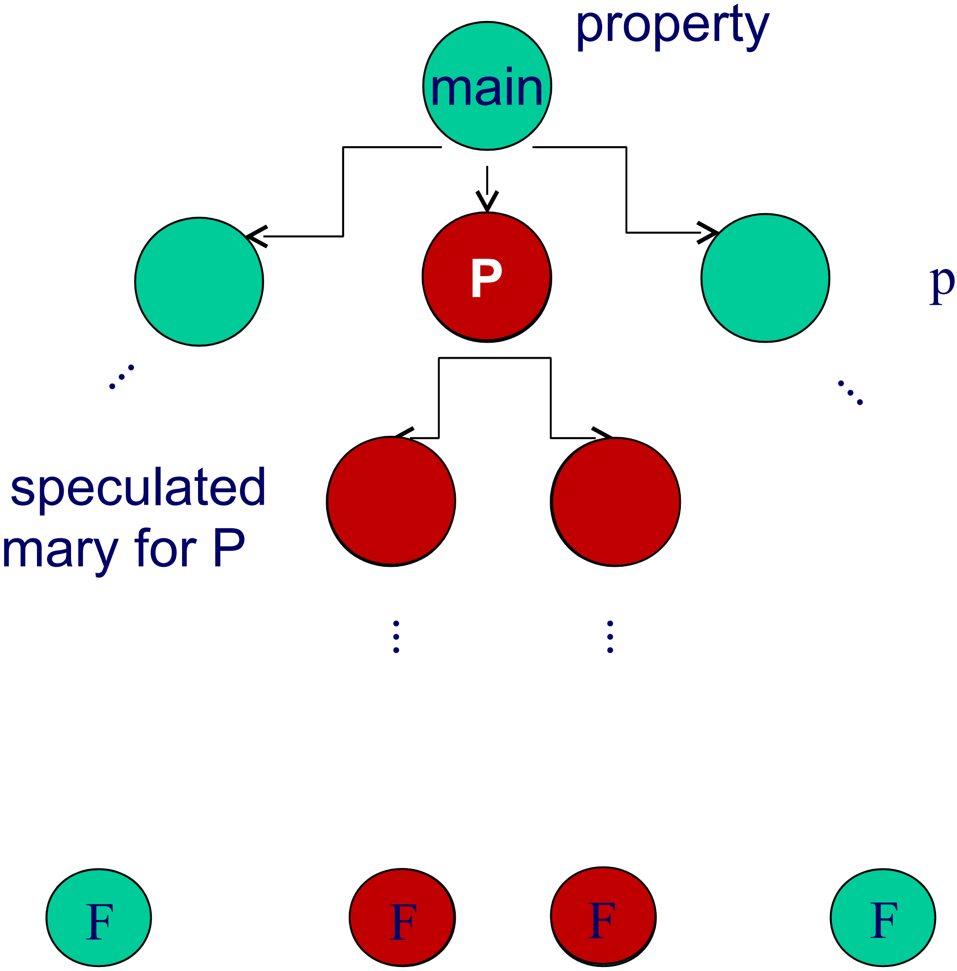
{True}

{x=y}

{y>x}

{False}

Duality: Summaries from Interpolants

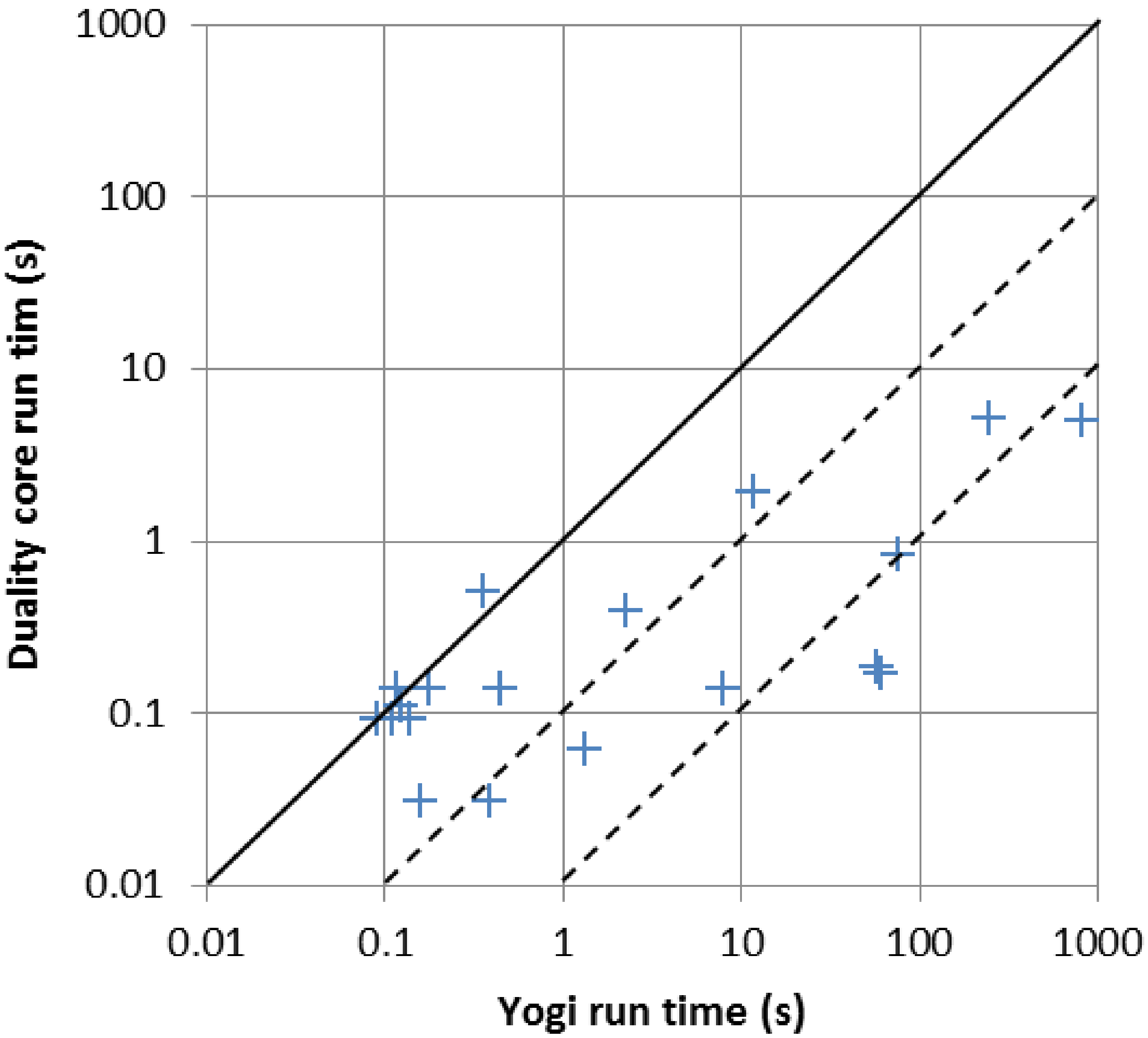


property

procedure instances

Interpolant is a speculated procedure summary for P

Duality performance vs. Yogi



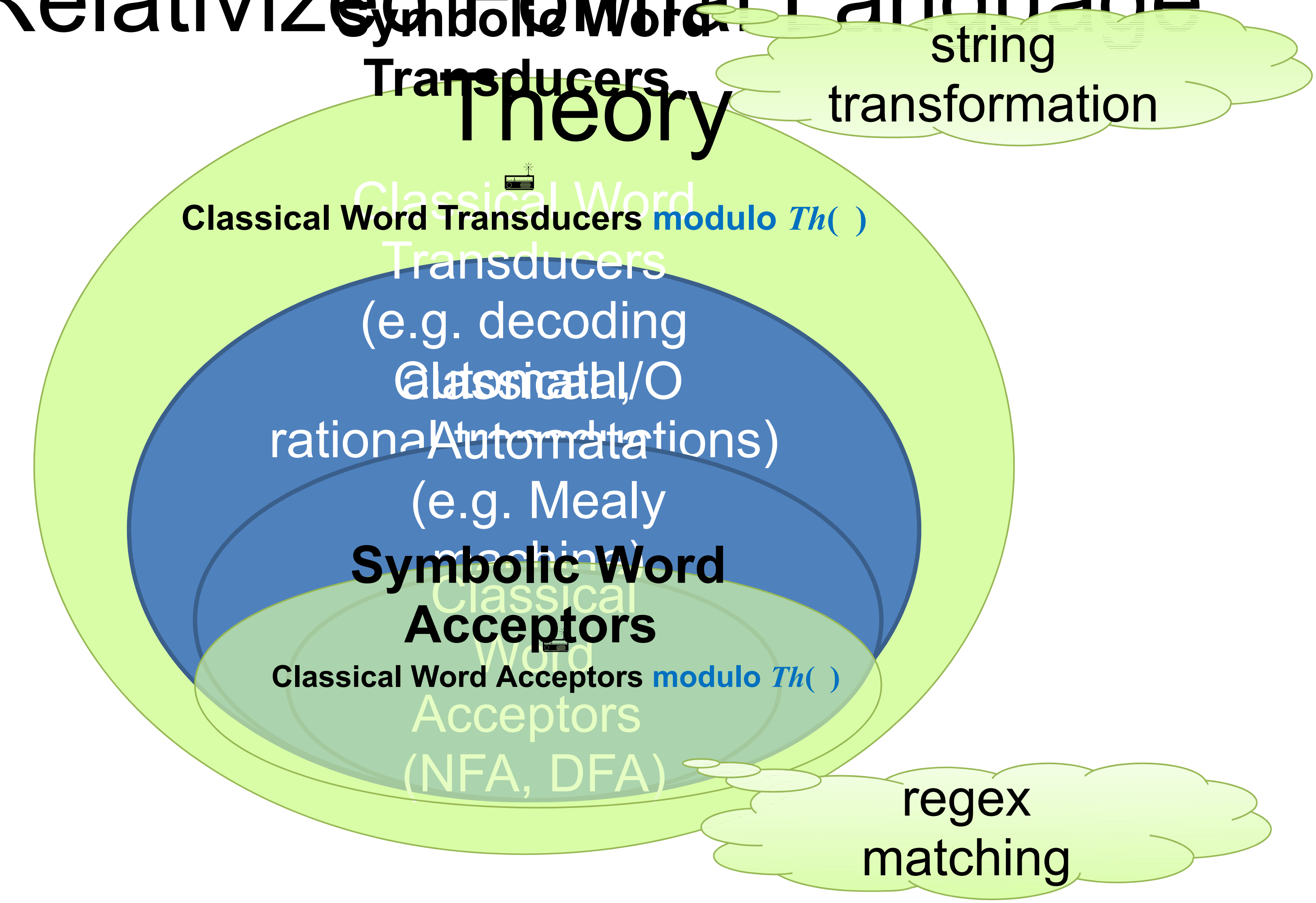
Symbolic Automata and Transducers

Margus Veanes, Nikolaj Bjørner
(POPL 2011)

Core Question

Can classical automata theory and algorithms be extended to work *modulo* large (infinite) alphabets 🍸 ?

Symbolic Automata:
Relativized Formal Language Theory



Symbolic Word Transducers

Classical Word Transducers modulo *Th*( )

Classical Word Transducers (e.g. decoding automata, rational transductions)

Classical I/O Automata (e.g. Mealy machine)

Symbolic Word Acceptors

Classical Word Acceptors modulo *Th*( )

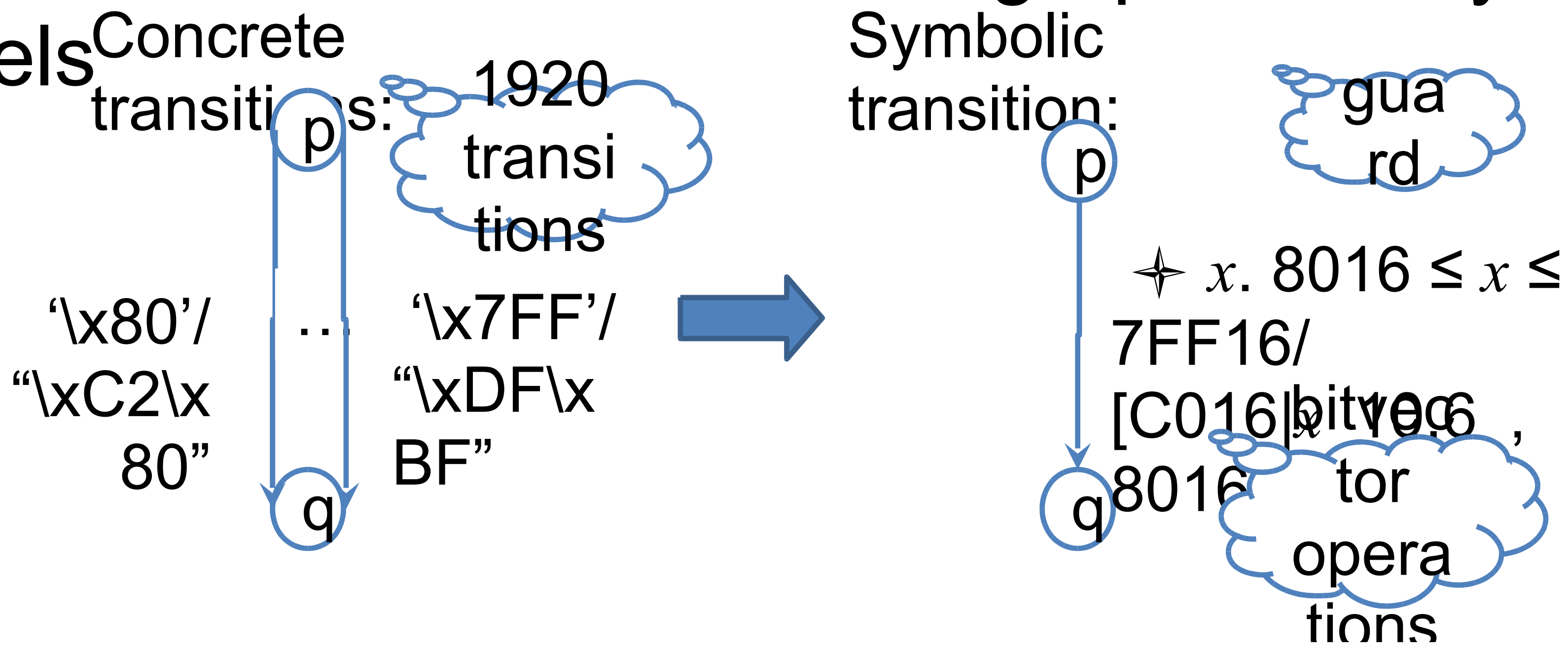
Classical Word Acceptors (NFA, DFA)

Symbolic Finite Transducer (SFT)

Classical transducer *modulo* a rich *label theory*

Core Idea: represent labels with guarded transformers

Separation of concerns: finite graph / theory of labels



Concrete transitions:

Symbolic transition:

Algorithms

New algorithms for SFAs and SFTs

Usin
g Z3

Extensions of classical algorithms *modulo* Th( )

Big-O complexity matches that of classical algorithms, with factor for decision procedure

Analysis

- Example 1: 🕶 *x*(utf8encode(*x*) 🔒 *Rutf8*) ?

1. $E = SFT(\text{utf8encode})$
2. $A = Complement(SFA(Rutf8))$
3. $B =$ ✦ $x.\ A(E(x))$
4. B 📹 ✆ ?

Does there exist an input x that causes a bad output ?

- Example 2: ✦ *x*.utf8decode(utf8encode(*x*)) 📻 *Id* ?

Links

Symbolic Automata Tool Kit
http://research.microsoft.com/automata/

Rex (acceptors) online
http://rise4fun.com/rex/

Bek (transducers) online
Samples: http://rise4fun.com/Bek/
Tutorials: http://rise4fun.com/Bek/tutorial

Solving Nonlinear Arithmetic

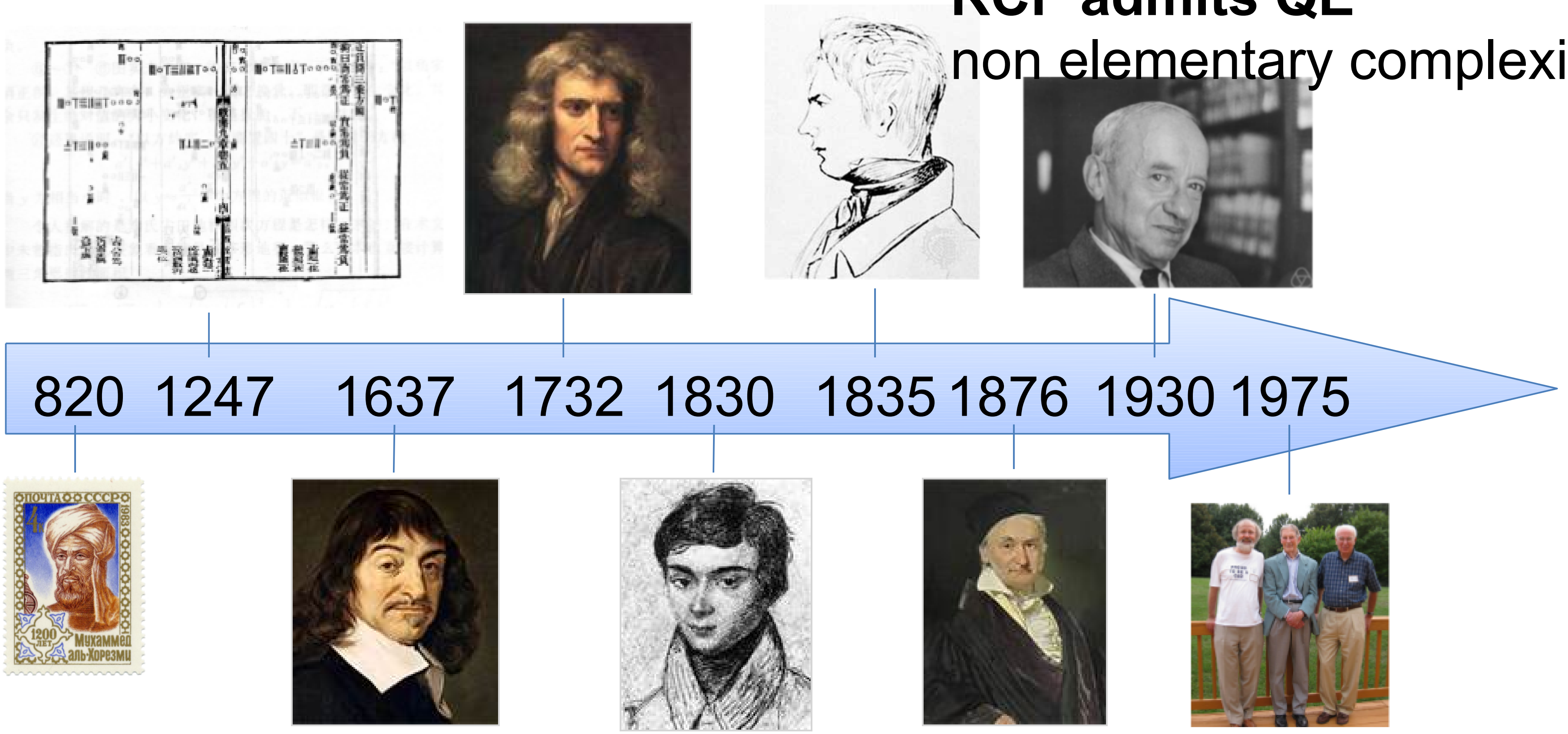
Dejan Jovanović (NYU) and Leonardo de Moura

(IJCAR 2012)

Polynomial Constraints

AKA
Existential Theory of the Reals
🕶 R

Milestones

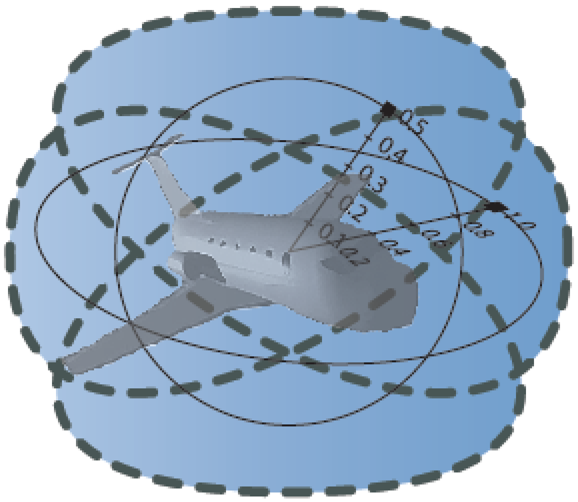


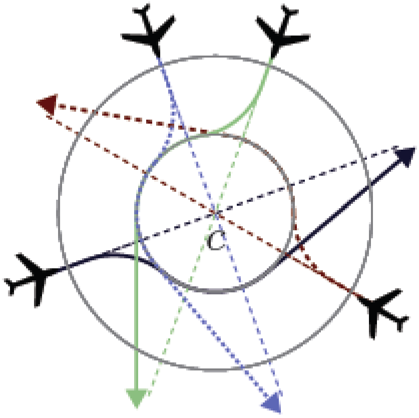
820 1247 1637 1732 1830 1835 1876 1930 1975

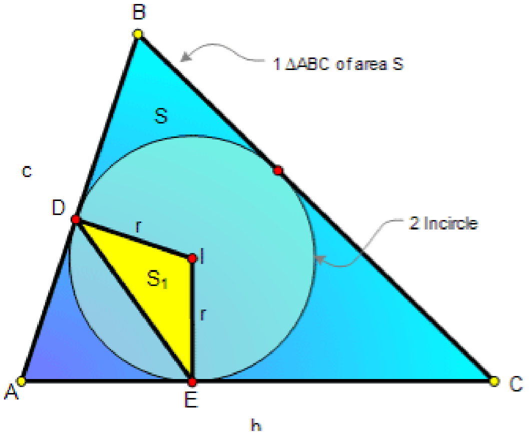
RCF admits QE
non elementary complexity

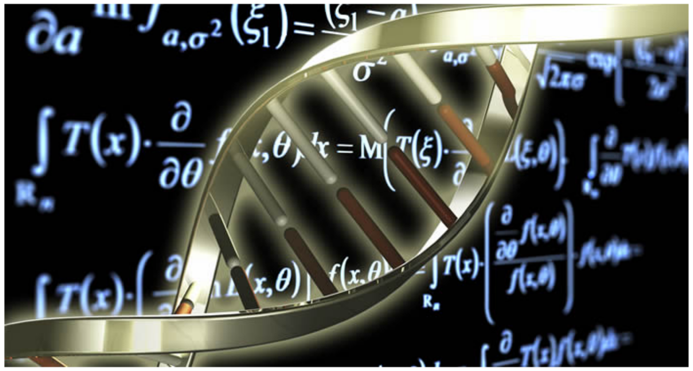
QE by CAD
Doubly exponential

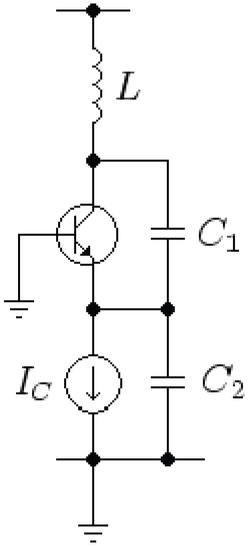
Applications





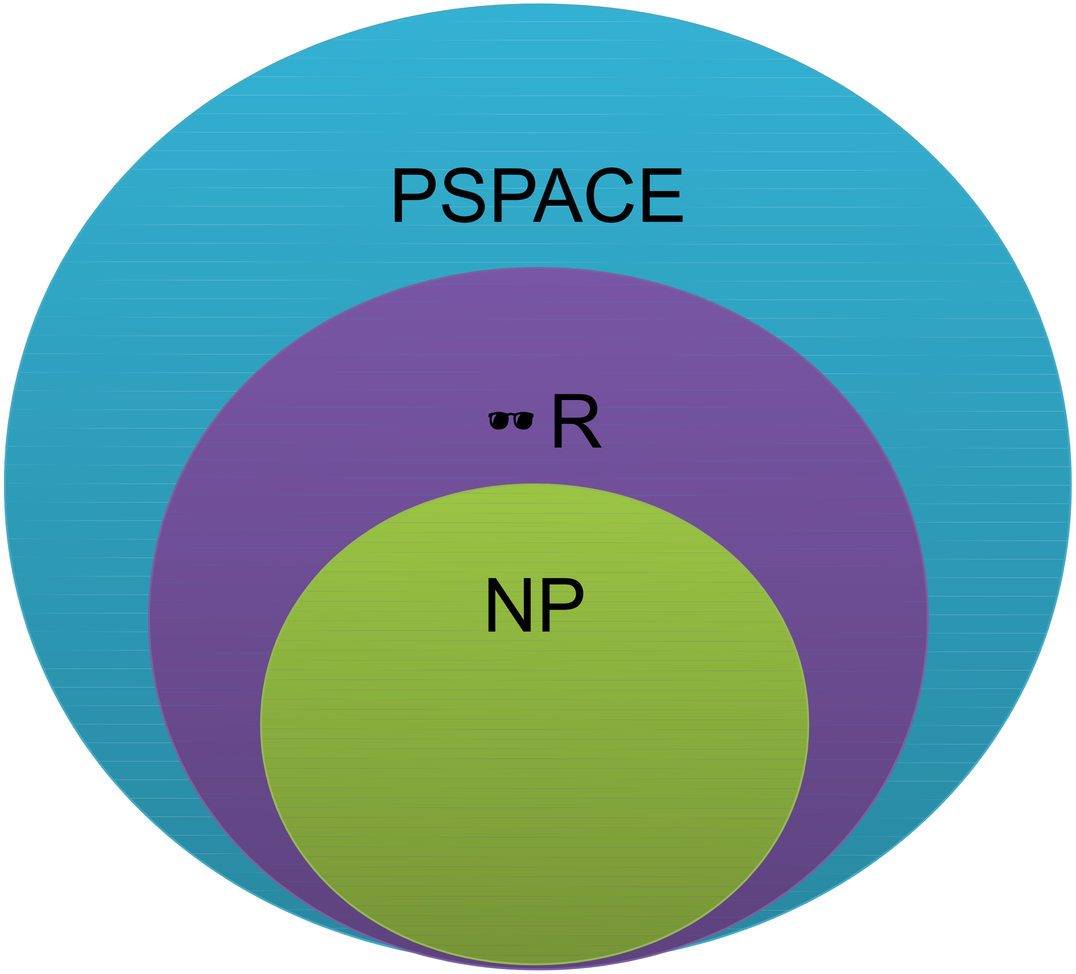






CiME

How hard is 🕶R?



PSPACE membership

Canny – 1988,

Grigor’ev – 1988

NP-hardness

x is “Boolean” x (x-1) = 0

x or y or z 🎭 x + y + z
0

CAD “Big Picture”

1. Saturate

2. Search

Our Procedure

Start search before saturate/project

Saturate on demand

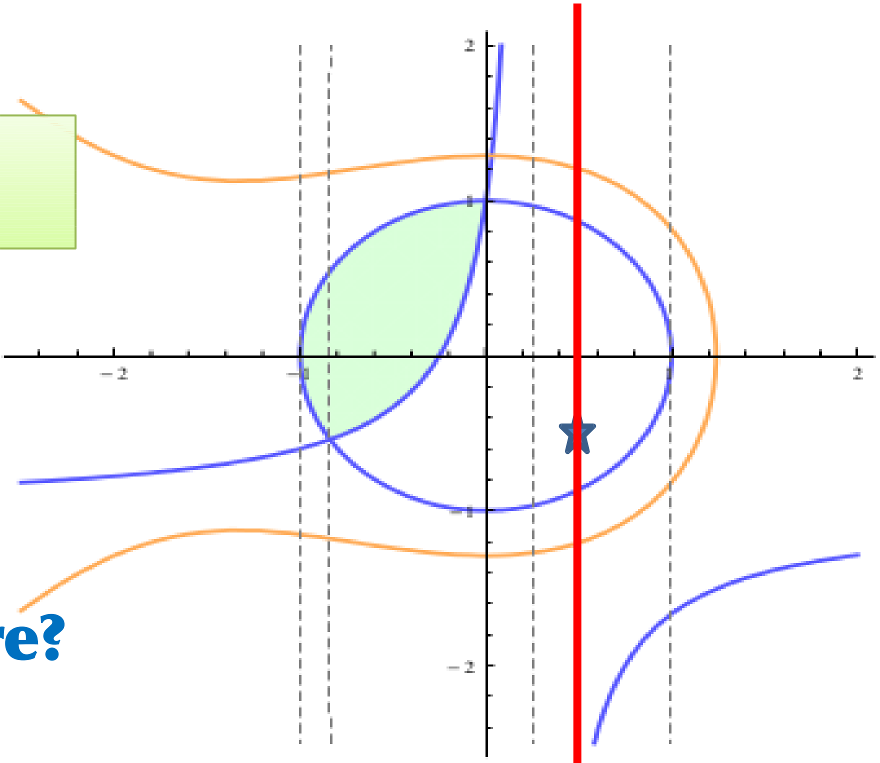
Apply SAT solver heuristics

Learn lemmas from conflicts

Non-chronological backtracking

Our Procedure (1)

Key ideas: Use partial solution to guide the search



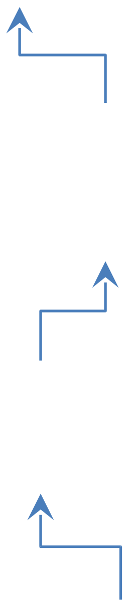
Feasible Region

What is the core?

Fig. 1. Solutions of $f_2 = x^2 + y^2 - 1 = 0$ and $f_3 = -4xy - 4x + y - 1 = 0$ in blue, solutions of $f_4 = x^3 + 2x^2 + 3y^2 - 5 = 0$ in orange. Solution set of $\{f_2 < 0, f_3 > 0, f_4 < 0$ in green. The dashed lines represent the zeroes of the projection set ($\boxed{2}$).

Our Procedure (2)

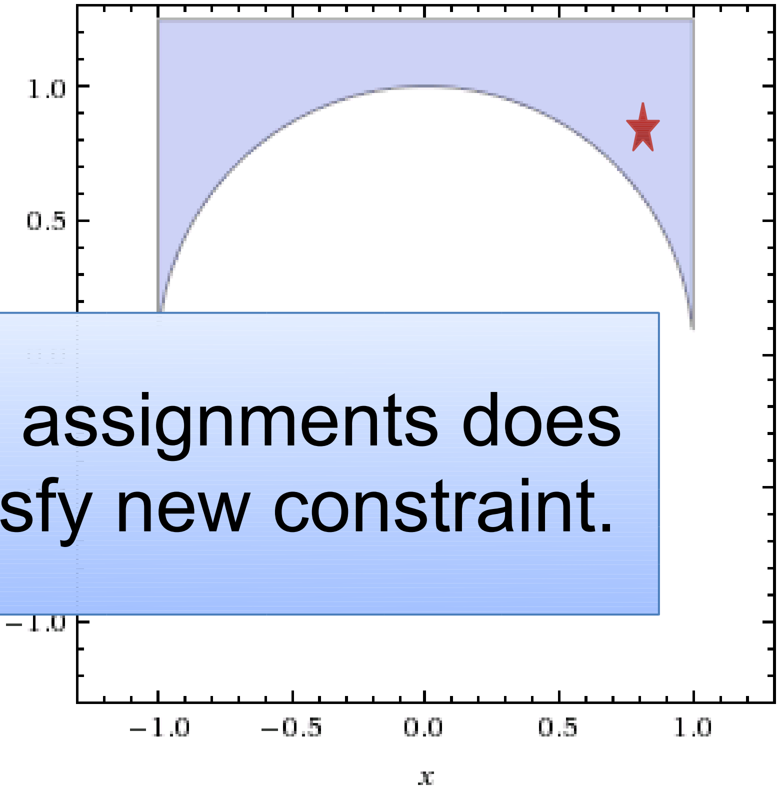
Key ideas: Nonchronological Backtracking



Our Procedure (3)

Key ideas: Lemma Learning

Prevent a **Conflict** from happening again.



Current assignments does not satisfy new constraint.

Complexity Trap: P Efficient

“Real algebraic numbers are efficient”

“CAD is polynomial for a fixed number of variables”

Every detail matters

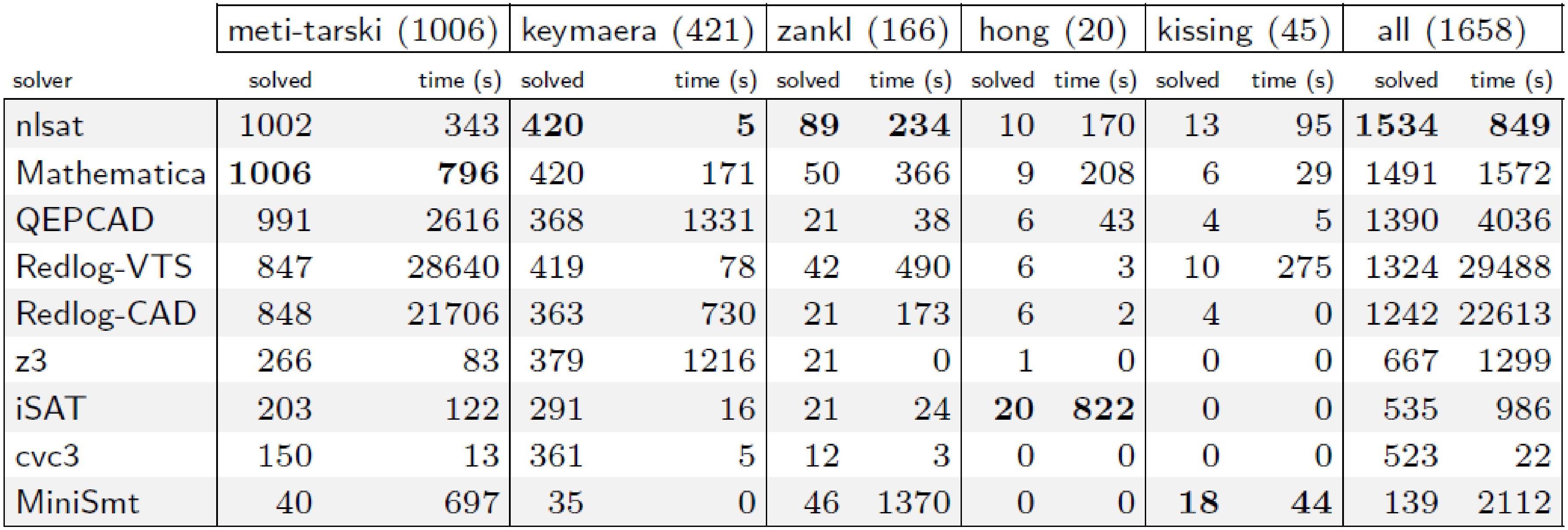
- GCD of two polynomials

Our procedure “dies” in polynomial time steps

- Real algebraic number computations
- Computing PSCs
- Root isolation of polynomials with irrational coefficients

Experimental Results

NEW ENGINE



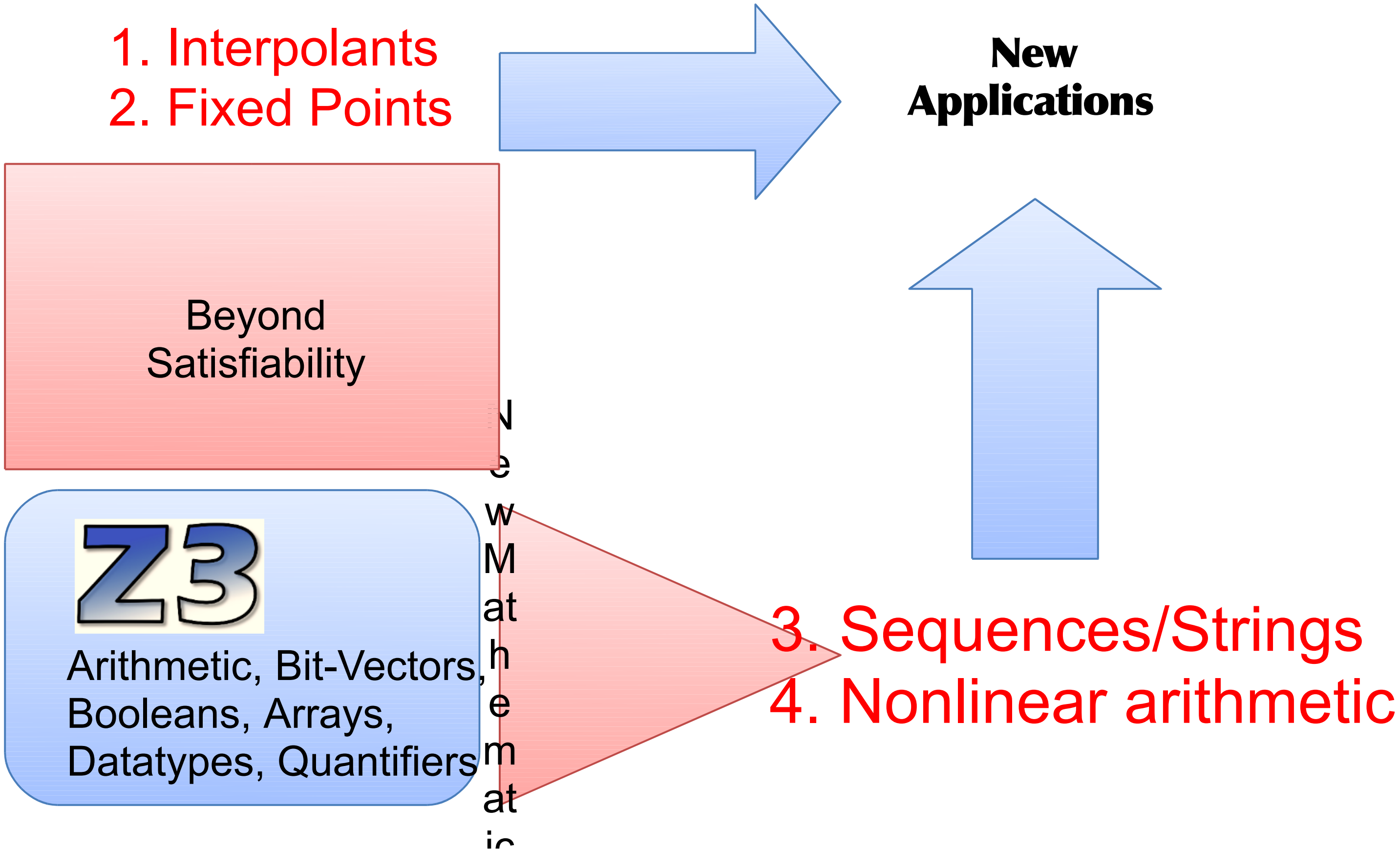
	meti-tarski (1006)		keymaera (421)		zankl (166)		hong (20)		kissing (45)		all (1658)	
solver	solved	time (s)	solved	time (s)	solved	time (s)	solved	time (s)	solved	time (s)	solved	time (s)
nlsat	1002	343	**420**	**5**	**89**	**234**	10	170	13	95	**1534**	**849**
Mathematica	**1006**	**796**	420	171	50	366	9	208	6	29	1491	1572
QEPCAD	991	2616	368	1331	21	38	6	43	4	5	1390	4036
Redlog-VTS	847	28640	419	78	42	490	6	3	10	275	1324	29488
Redlog-CAD	848	21706	363	730	21	173	6	2	4	0	1242	22613
z3	266	83	379	1216	21	0	1	0	0	0	667	1299
iSAT	203	122	291	16	21	24	**20**	**822**	0	0	535	986
cvc3	150	13	361	5	12	3	0	0	0	0	523	22
MiniSmt	40	697	35	0	46	1370	0	0	**18**	**44**	139	2112

Conclusions

"Logic is the Calculus of Computer Science"

Automating mathematical logic

Logic engines as a service



1. Interpolants
2. Fixed Points

New Applications

Beyond Satisfiability

Z3

Arithmetic, Bit-Vectors, Booleans, Arrays, Datatypes, Quantifiers

New Mathematic

3. Sequences/Strings
4. Nonlinear arithmetic