Cylindrical Algebraic Decomposition for Nonlinear Arithmetic Problems

Gereon Kremer

PhD defense talk

12.03.2020

Grateful for the cooperation with and advise from:
Erika Ábrahám, Jens Brandt, Christopher Brown, Florian Corzilius, James H. Davenport, Matthew England, Vijay Ganesh, Rebecca Haehn, Marcel Hark, Einar Broch Johnsen, Sebastian Junges, Viktor Levandovskyy, Jacopo Mauro, Jasper Nalbach, Stefan Schupp, Tarik Viehmann, and many more.

Formal verification (with satisfiability modulo theories)

Formal verification (with satisfiability modulo theories)


Abstract
&
Formalize

Formal verification (with satisfiability modulo theories)

Abstract
&
Formalize

CBMC APROVE CPA✓ Vcc Dafny

Formal verification (with satisfiability modulo theories)

Abstract & Formalize

CBMC

APROVE

CPA✓

Dafny

Encode to SMT

Formal verification (with satisfiability modulo theories)

Abstract & Formalize

CBMC
AProVE
CPA✓
Vcc
nuXmv
Dafny

Encode to SMT

MathSAT 5
veriT
Yices2
CVC4
Z3
SMT RAT

Formal verification (with satisfiability modulo theories)

Abstract & Formalize

CBMC

AProVE

CPA✓

VCC

nuXmv

Dafny

Encode to SMT

Interpret results

MathSAT 5

VeriT

Yices2

CVC4

Z3

SMT RAT

Formal verification (with satisfiability modulo theories)

Abstract & Formalize

Encode to SMT

Interpret results

SMT formulae:

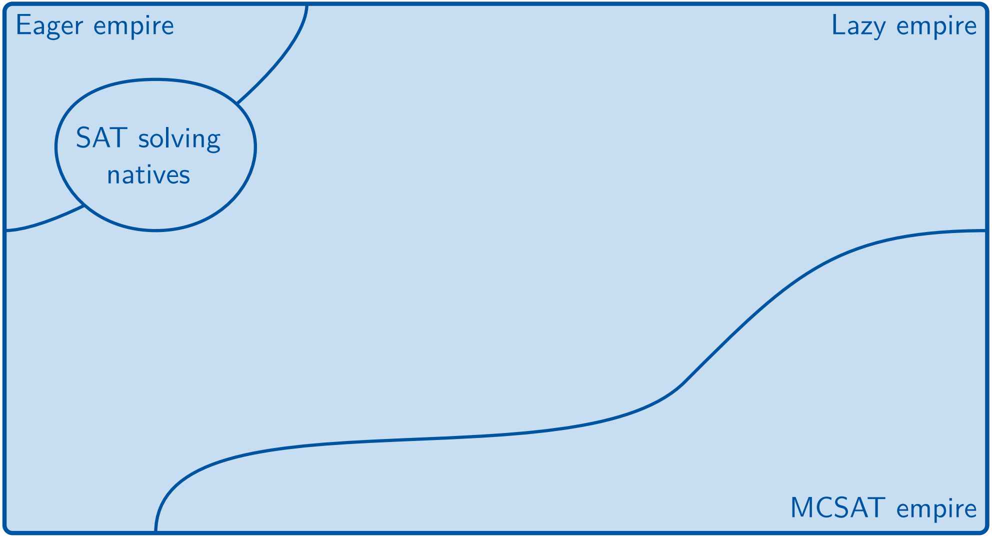
$$\exists x_1, \ldots, x_n \in \mathbb{R}.\ \varphi(x_1, \ldots, x_n)$$

Theory constraints:

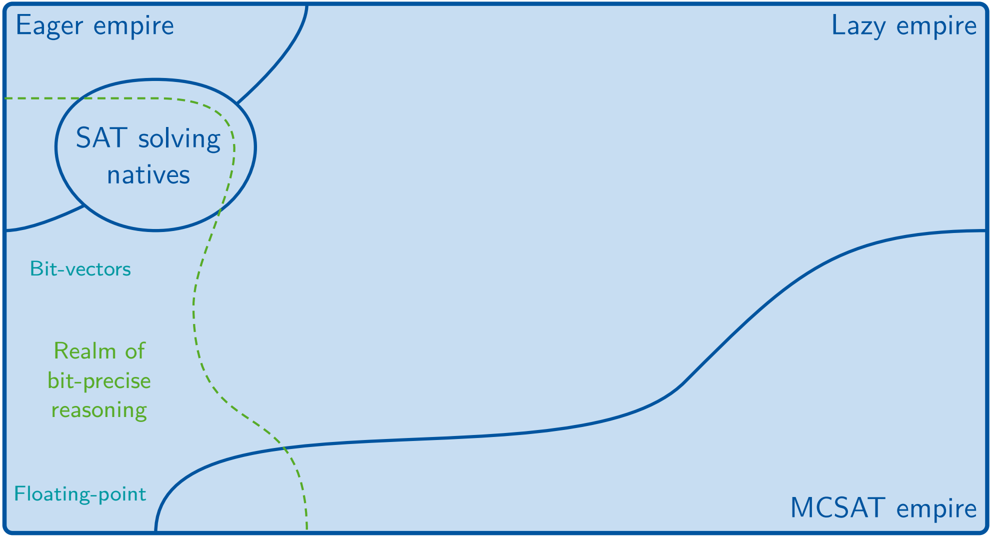
$$p \sim 0 \qquad p \in \mathbb{Z}[x_1, \ldots, x_n]$$

In SMT-LIB as `QF_NRA` since 2013

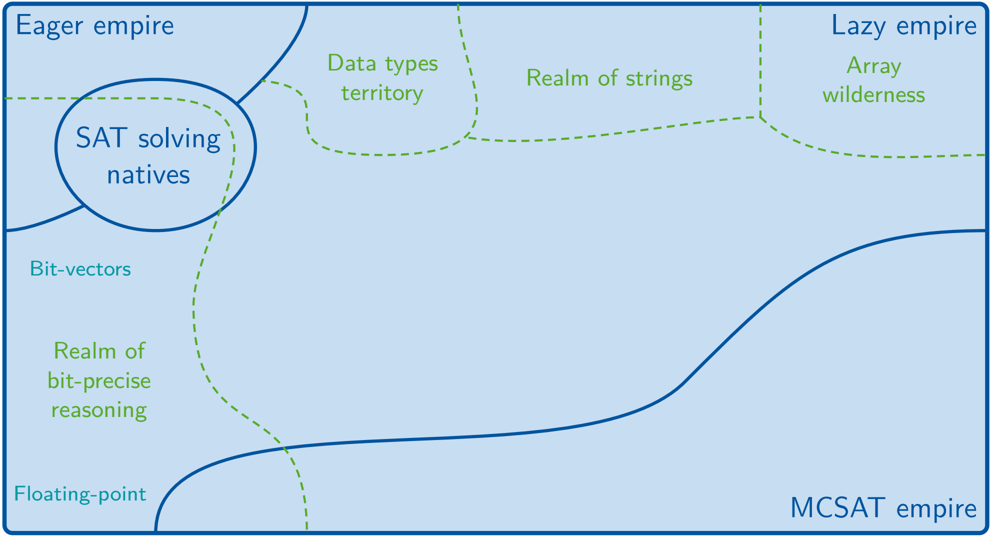
World Map of SMT



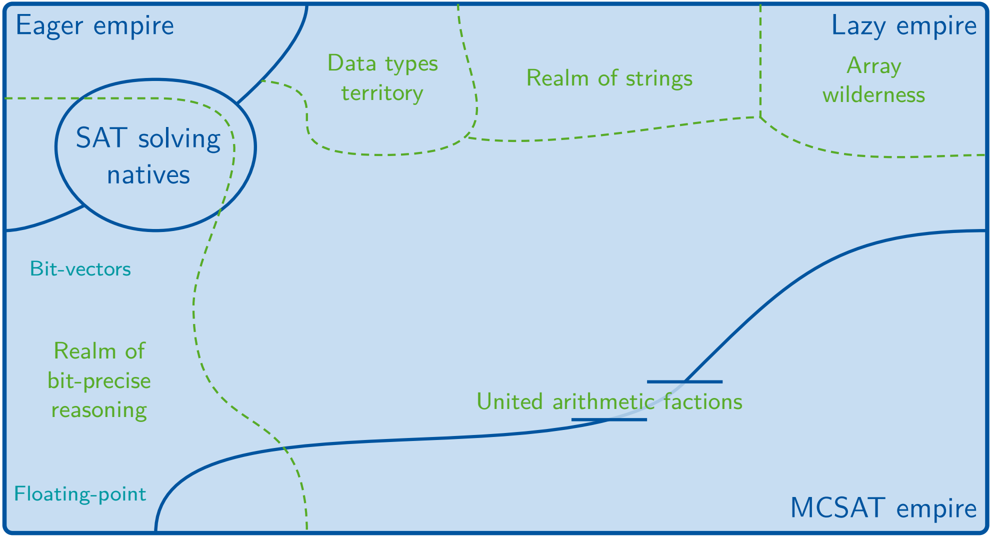
World Map of SMT



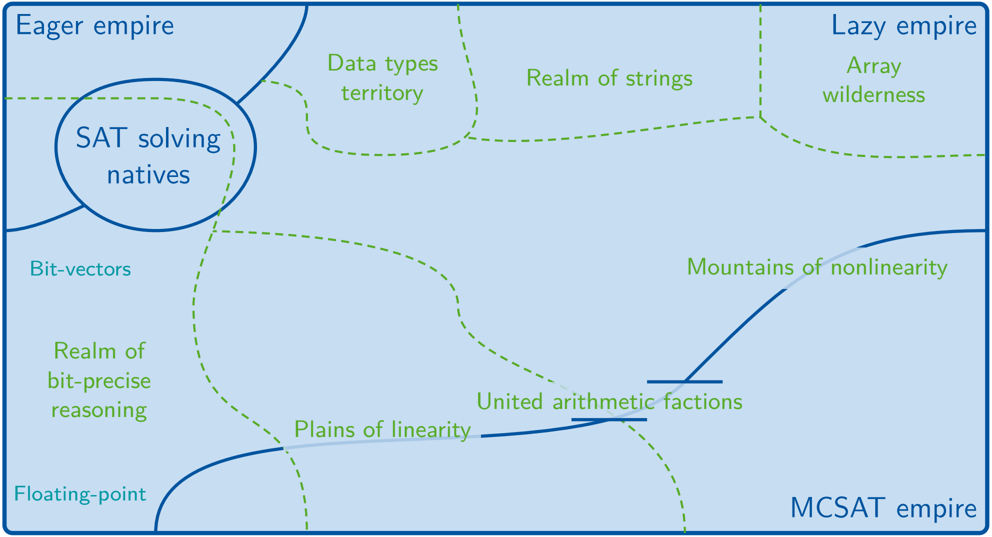
World Map of SMT



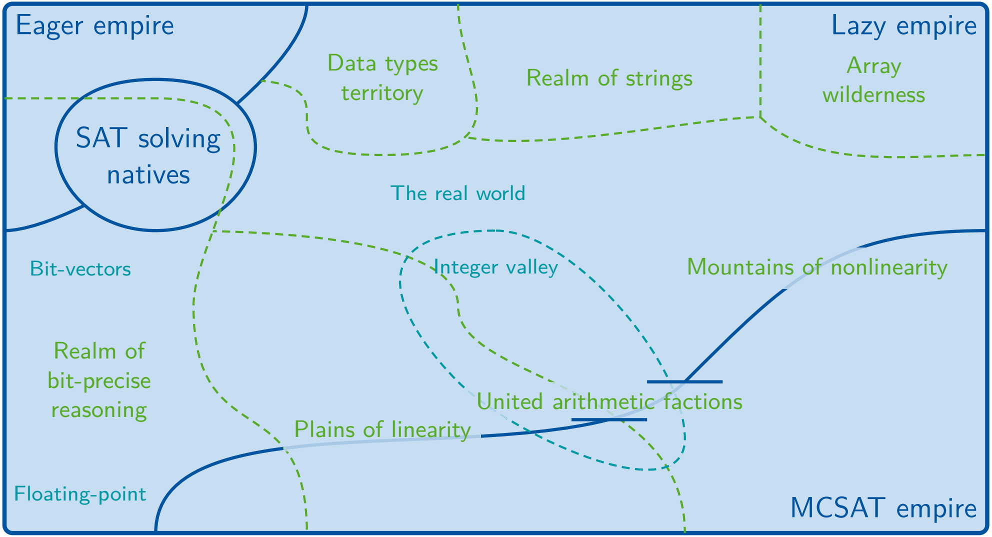
World Map of SMT



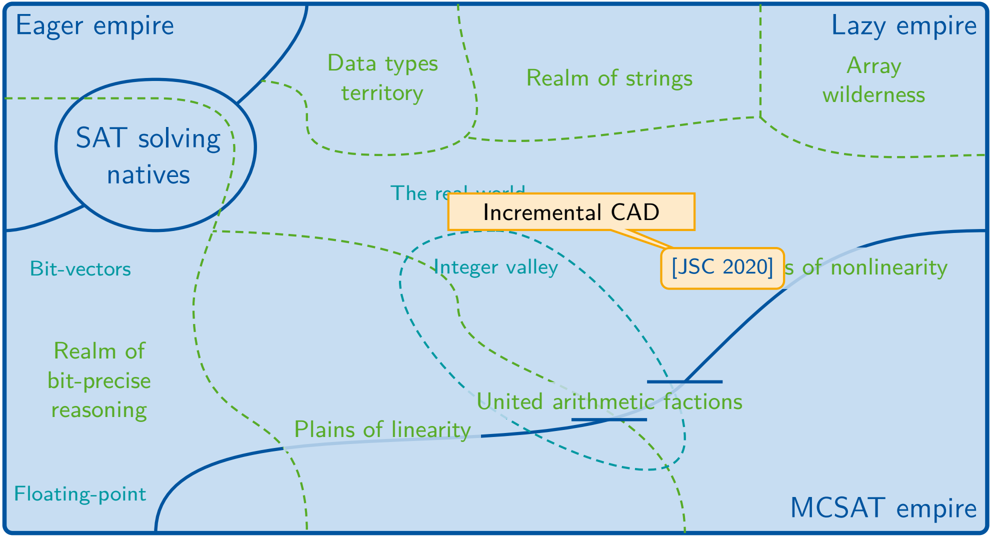
World Map of SMT



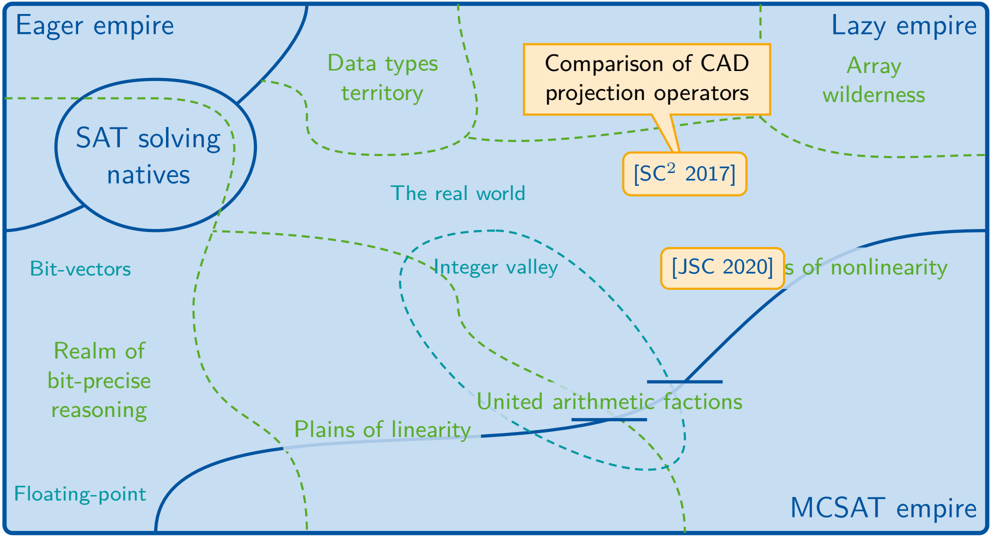
World Map of SMT



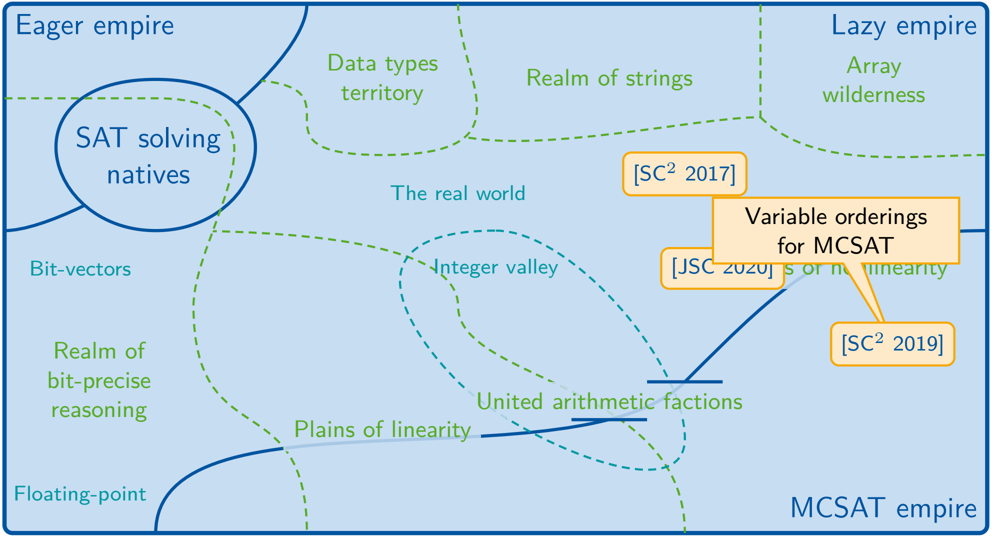
World Map of SMT



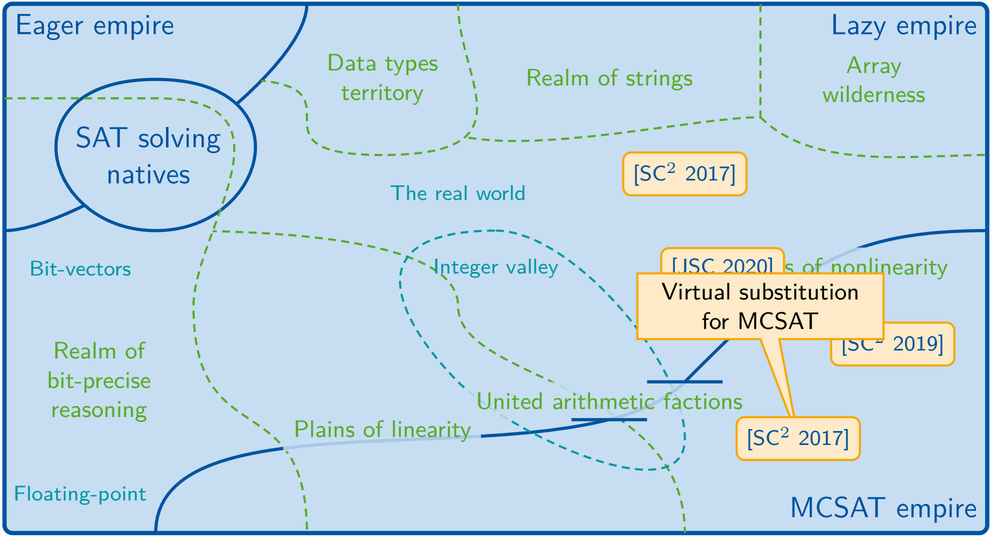
World Map of SMT



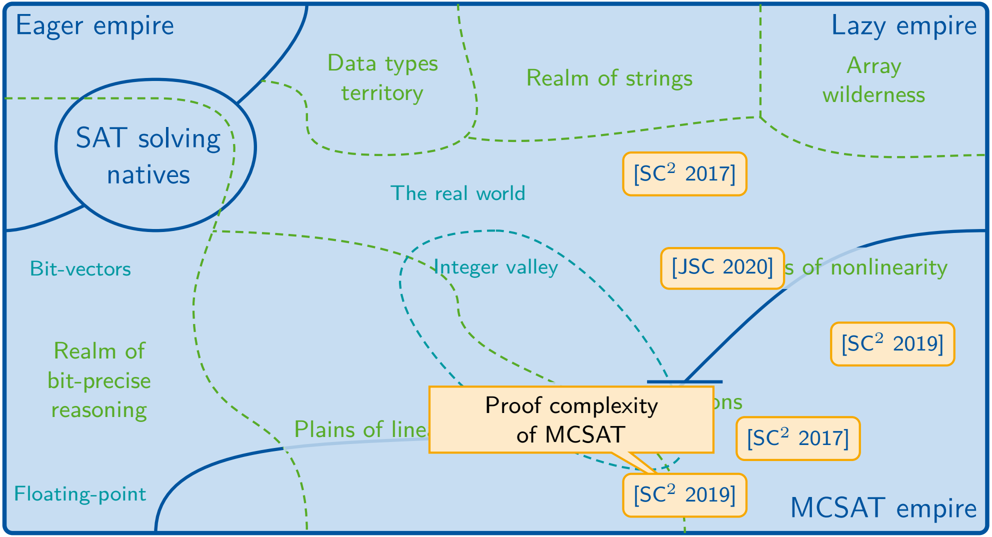
World Map of SMT



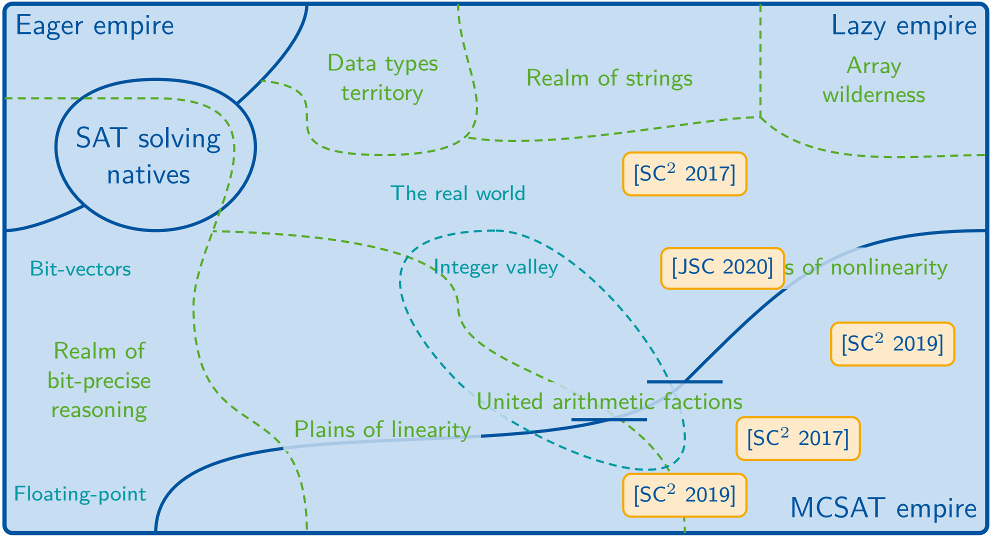
World Map of SMT



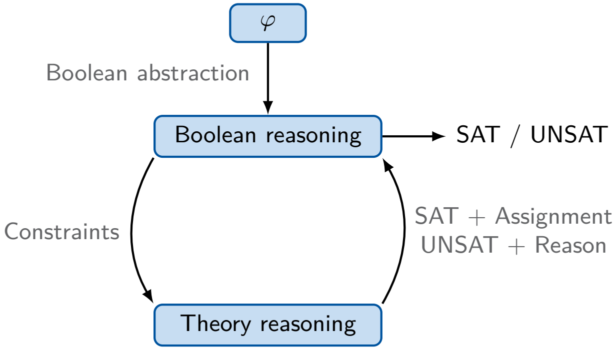
World Map of SMT



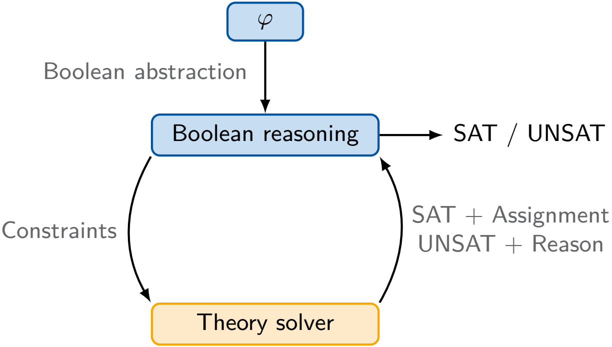
World Map of SMT



Lazy SMT solving

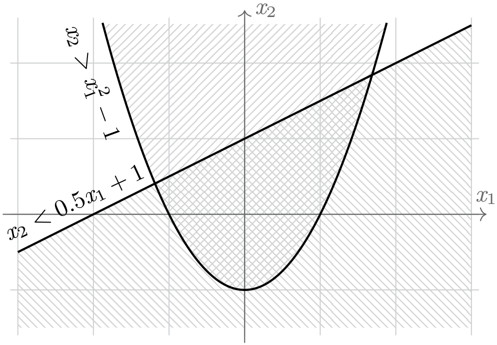


Lazy SMT solving



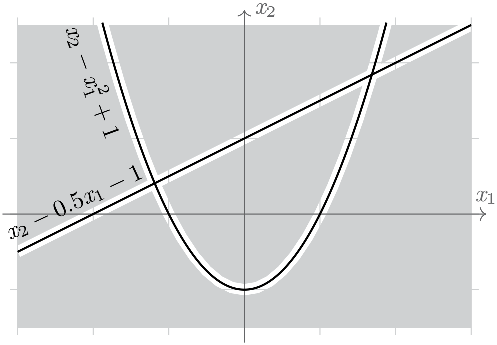
Core ideas of CAD

- Input: set of constraints



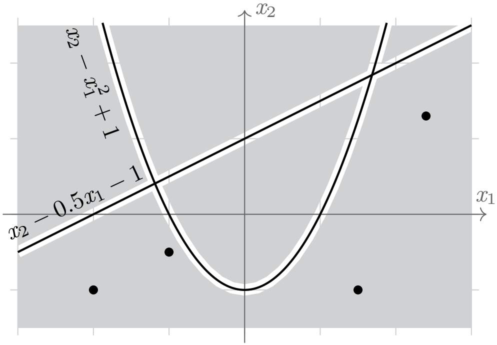
Core ideas of CAD

- Input: set of constraints
- Consider polynomials of these constraints
- Identify sign-invariant regions



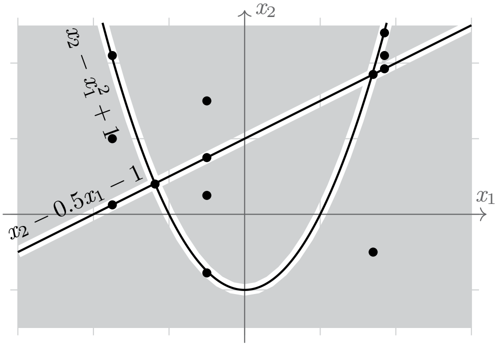
Core ideas of CAD

- Input: set of constraints
- Consider polynomials of these constraints
- Identify sign-invariant regions
- Sign-invariance (of polynomials) implies truth-invariance (of the formula)
- All samples in one region are equivalent



Core ideas of CAD

- Input: set of constraints
- Consider polynomials of these constraints
- Identify sign-invariant regions
- Sign-invariance (of polynomials) implies truth-invariance (of the formula)
- All samples in one region are equivalent
- Construct one sample per region
- Evaluate samples on constraints



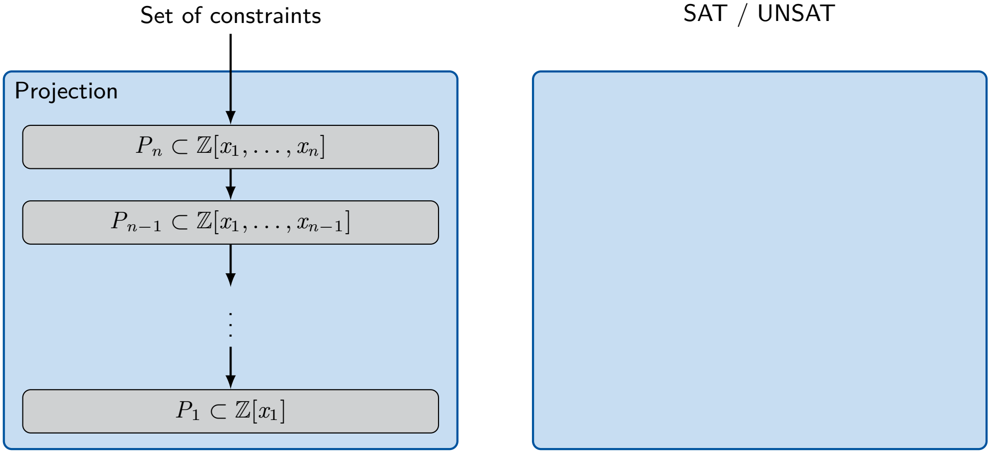
CAD in a nutshell

Set of constraints

SAT / UNSAT

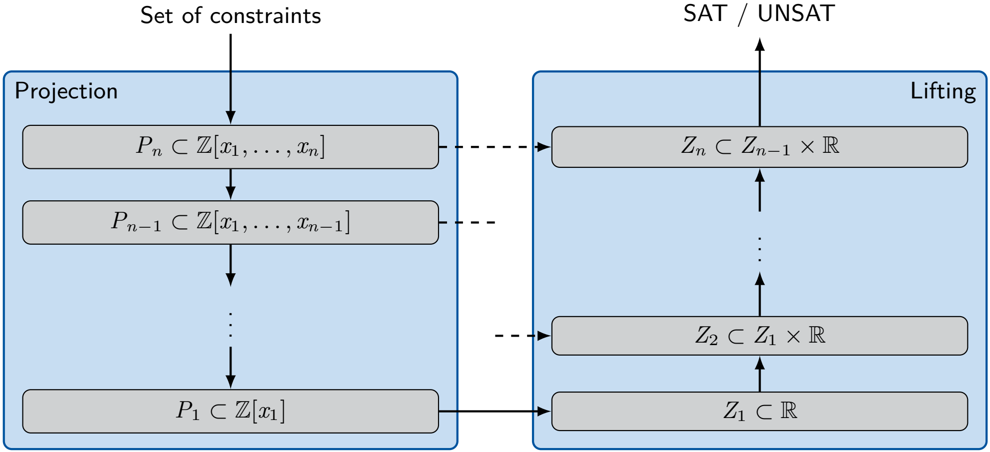
[Collins 1975]

CAD in a nutshell



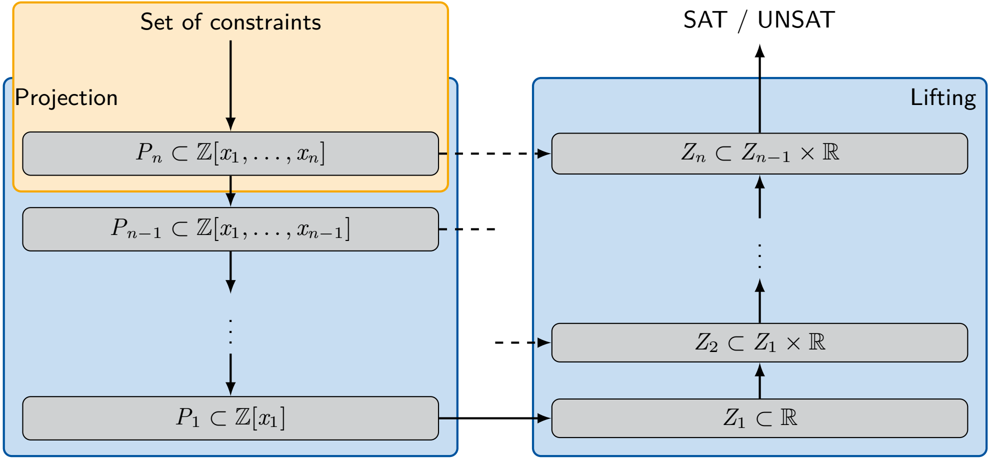
[Collins 1975]

CAD in a nutshell



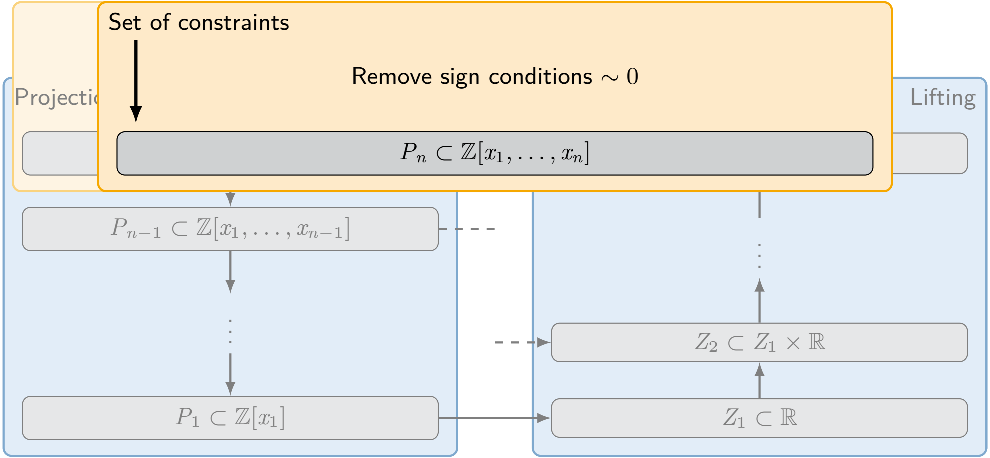
[Collins 1975]

CAD in a nutshell



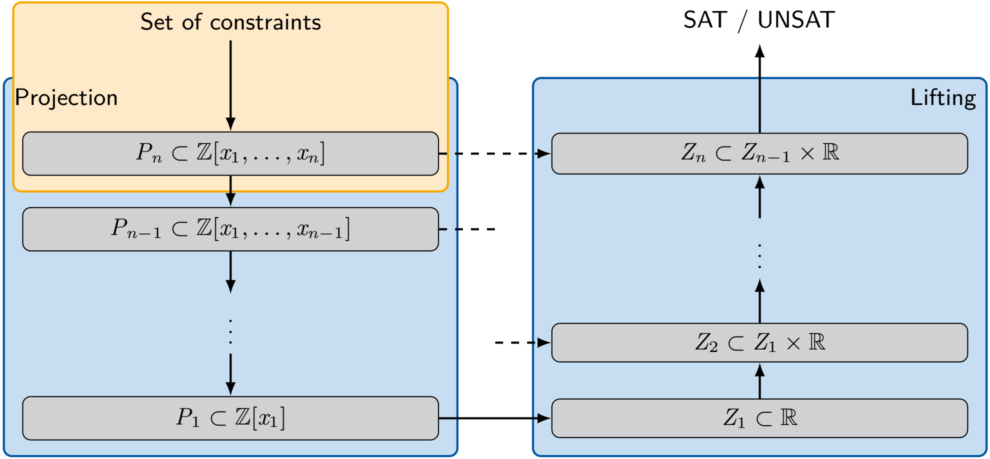
[Collins 1975]

CAD in a nutshell



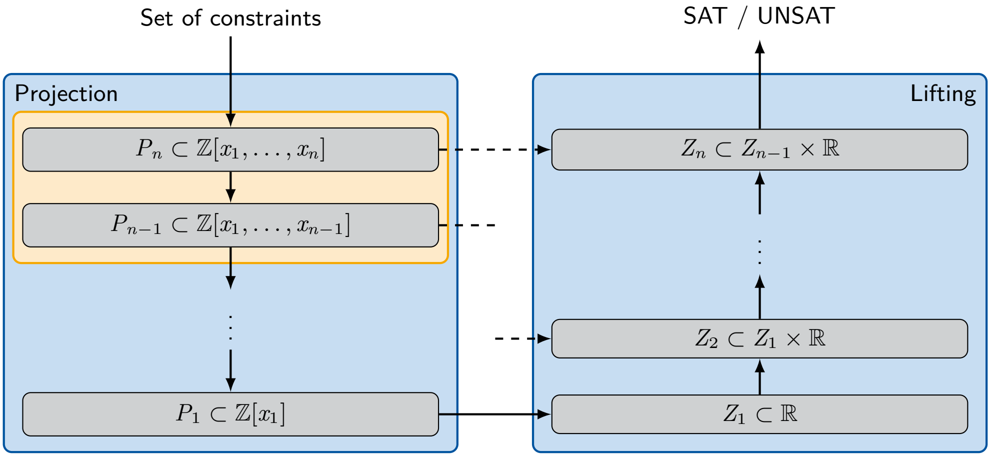
[Collins 1975]

CAD in a nutshell



[Collins 1975]

CAD in a nutshell



[Collins 1975]

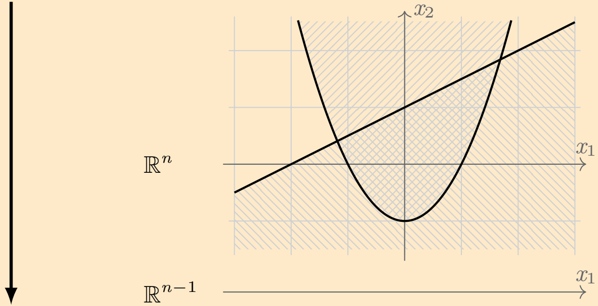
CAD in a nutshell

Projecti

Lifting

Projection

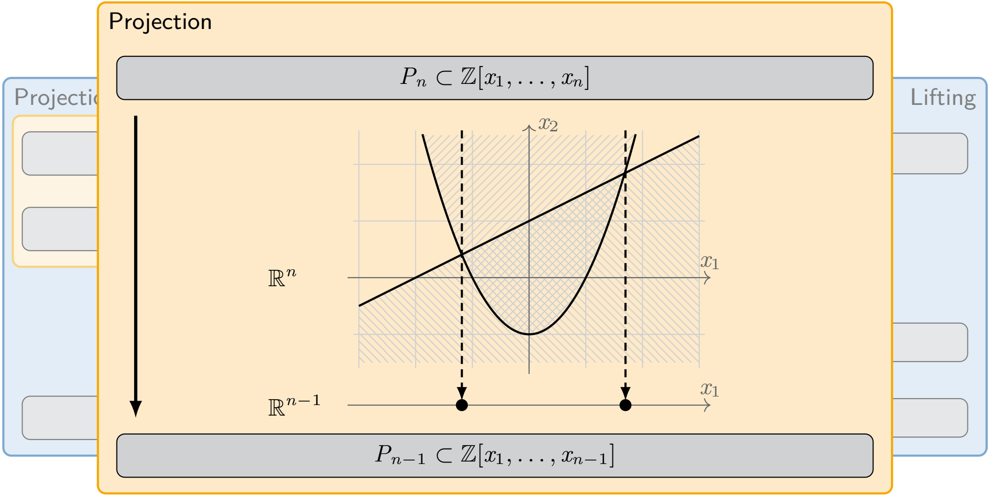
$$P_n \subset \mathbb{Z}[x_1, \ldots, x_n]$$



$$P_{n-1} \subset \mathbb{Z}[x_1, \ldots, x_{n-1}]$$

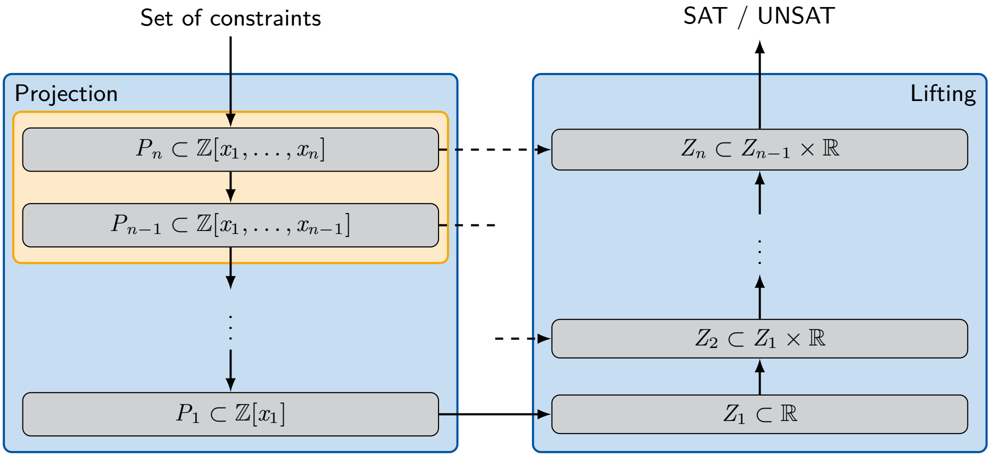
[Collins 1975]

CAD in a nutshell



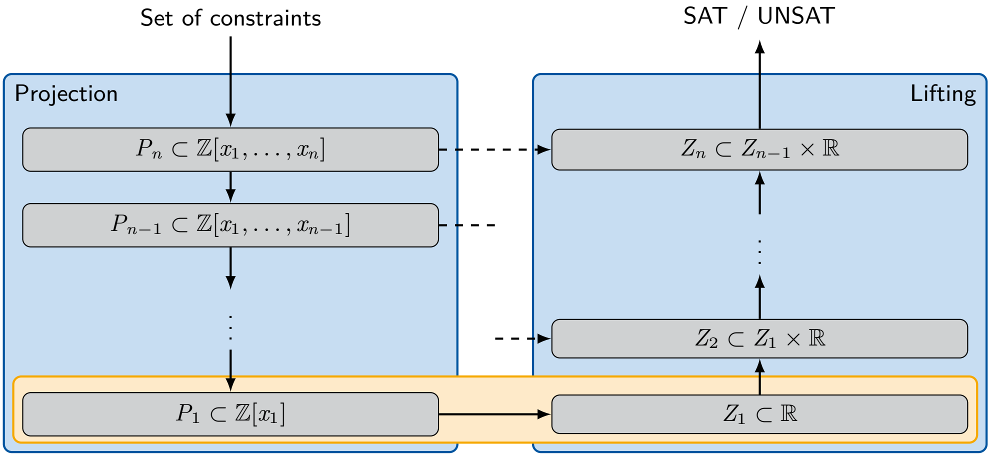
[Collins 1975]

CAD in a nutshell



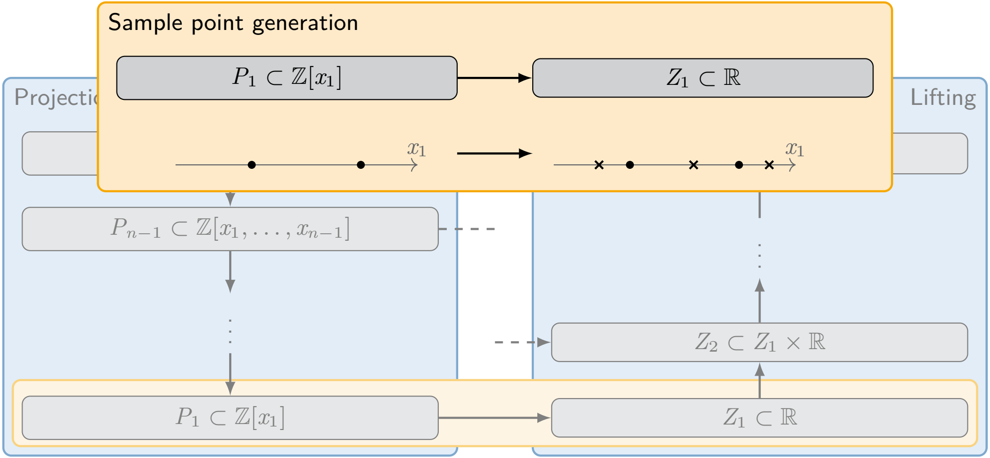
[Collins 1975]

CAD in a nutshell



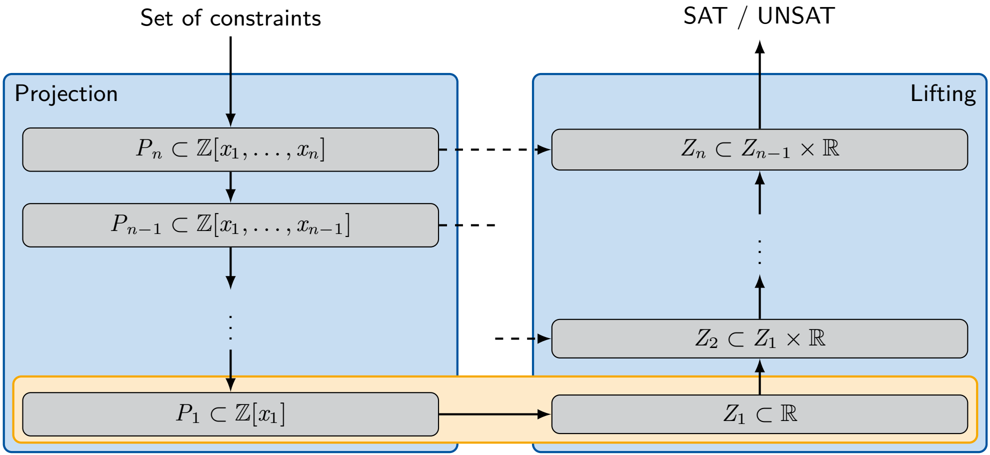
[Collins 1975]

CAD in a nutshell



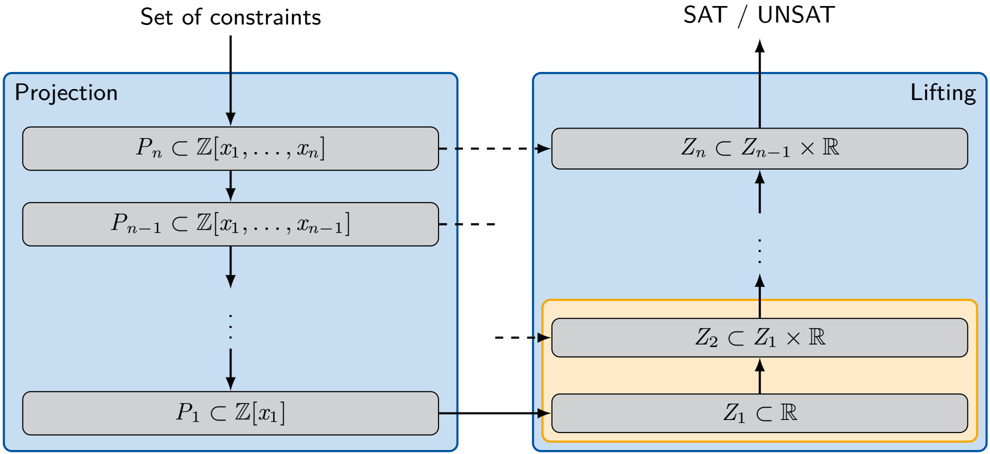
[Collins 1975]

CAD in a nutshell



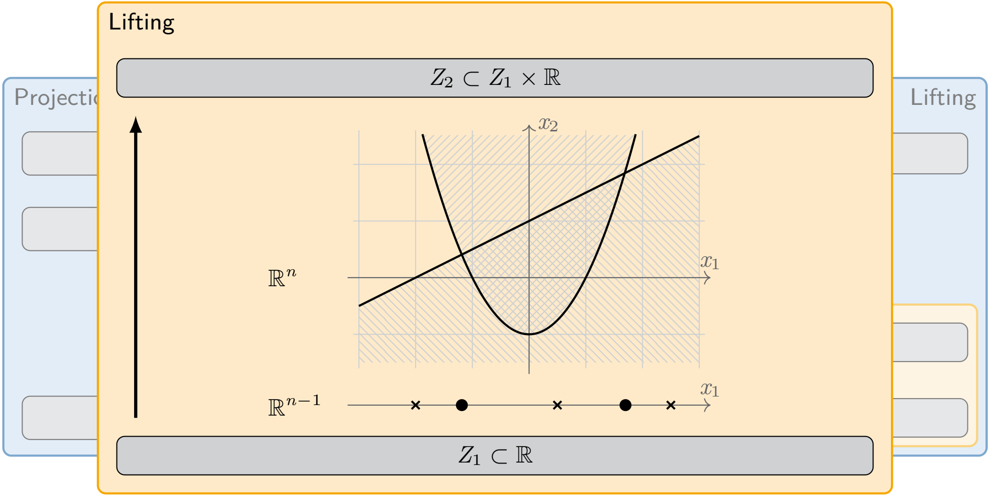
[Collins 1975]

CAD in a nutshell



[Collins 1975]

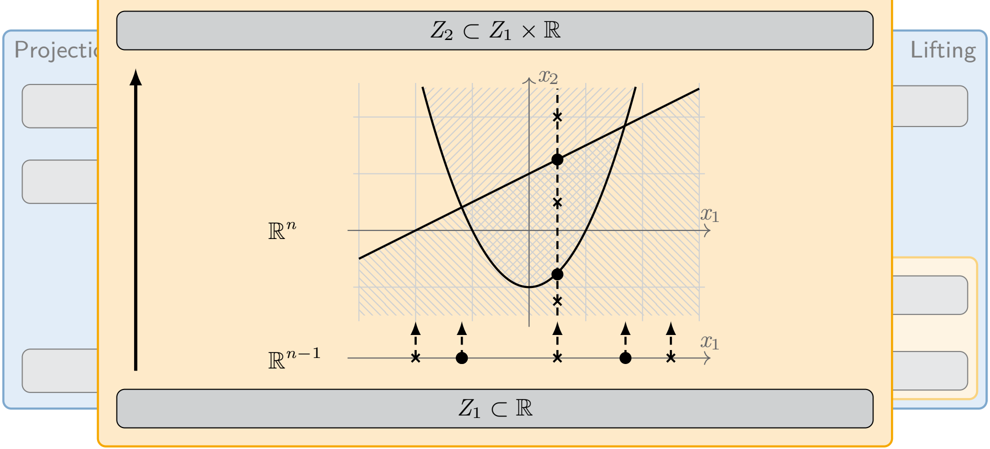
CAD in a nutshell



[Collins 1975]

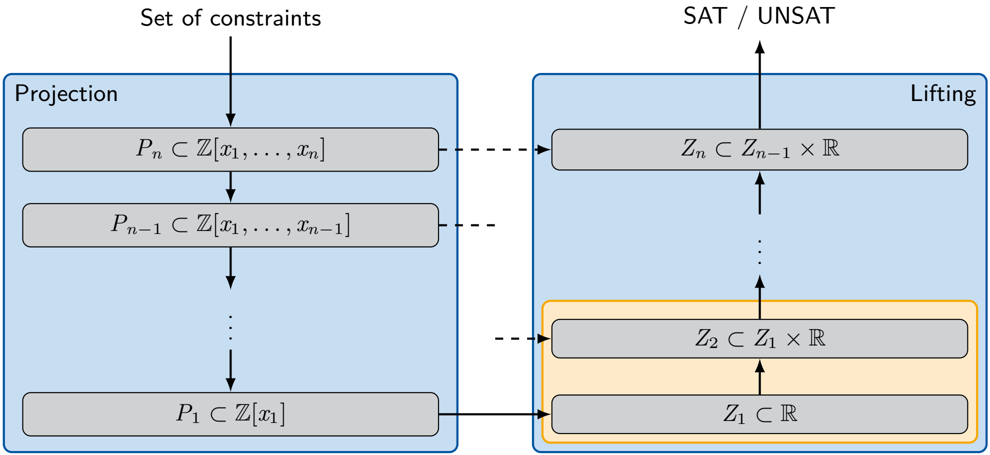
CAD in a nutshell

Lifting



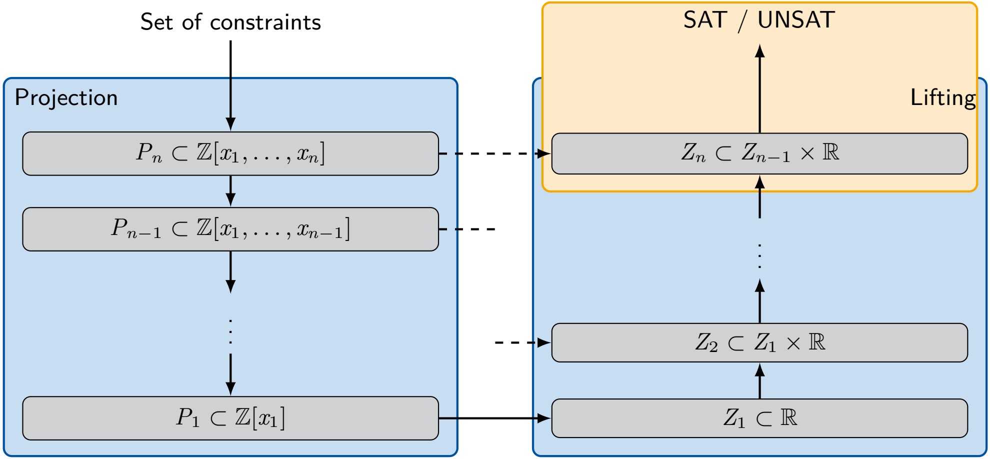
[Collins 1975]

CAD in a nutshell



[Collins 1975]

CAD in a nutshell



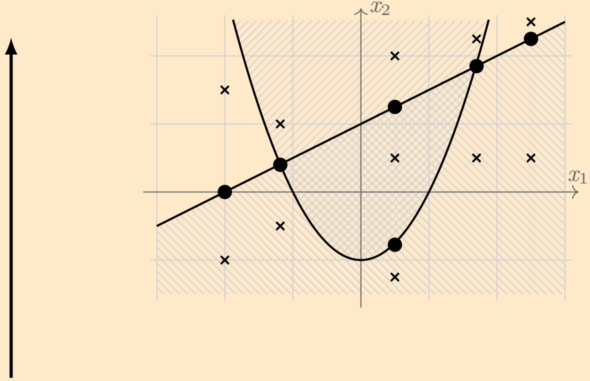
[Collins 1975]

CAD in a nutshell

Projecti

Lifting

SAT / UNSAT



$Z_n \subset Z_{n-1} \times \mathbb{R}$

[Collins 1975]

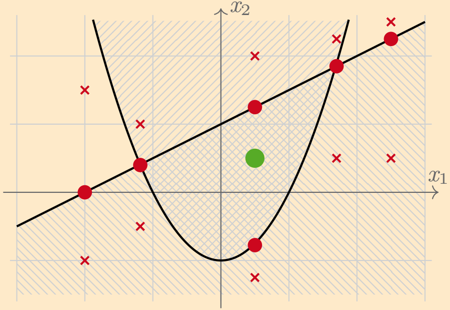
CAD in a nutshell

Projecti

Lifting

SAT / UNSAT

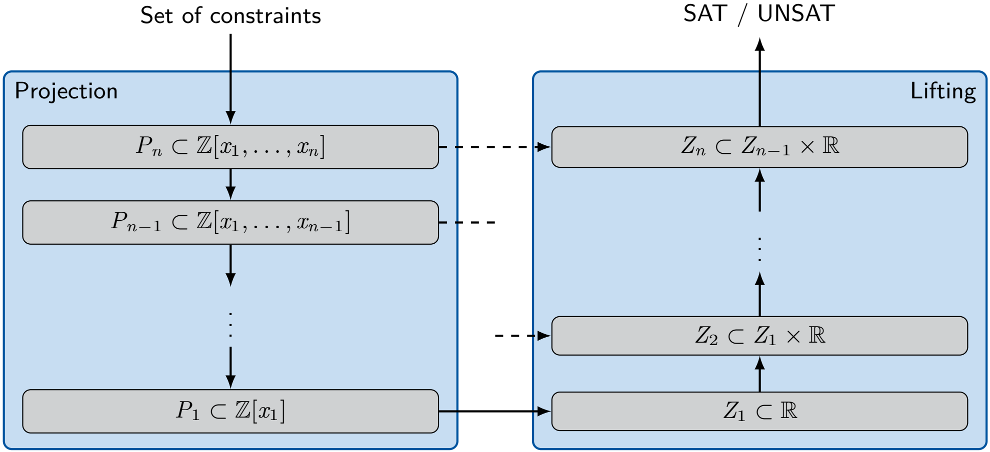
SAT



$Z_n \subset Z_{n-1} \times \mathbb{R}$

[Collins 1975]

CAD in a nutshell



[Collins 1975]

Towards incrementality

~ 2010: How can we solve nonlinear real arithmetic in SMT?

Towards incrementality

~ 2010: How can we solve nonlinear real arithmetic in SMT?

- CAD established for quantifier elimination
- First attempts with CAD seem promising [Loup + 2011] [Corzilius + 2012] [Loup + 2013]
- Since 2016: SC^2

Towards incrementality

~ 2010: How can we solve nonlinear real arithmetic in SMT?

- CAD established for quantifier elimination
- First attempts with CAD seem promising [Loup + 2011] [Corzilius + 2012] [Loup + 2013]
- Since 2016: SC^2

Our goal: make CAD as efficient as possible for SMT

Towards incrementality

~ 2010: How can we solve nonlinear real arithmetic in SMT?

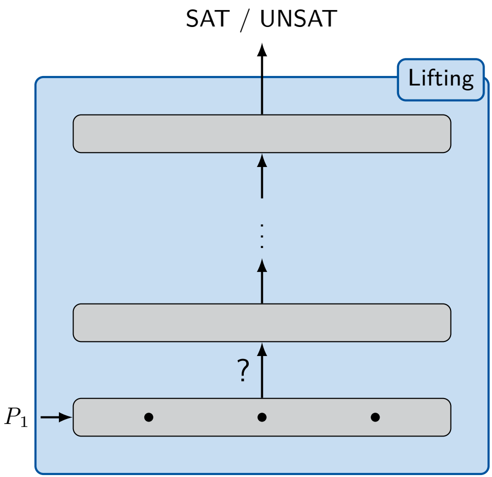
- CAD established for quantifier elimination
- First attempts with CAD seem promising [Loup + 2011] [Corzilius + 2012] [Loup + 2013]
- Since 2016: SC^2

Our goal: make CAD as efficient as possible for SMT

- Incrementality: reuse previously computed results
- Backtracking: remove part of the input efficiently
- Reasons for unsatisfiability: small infeasible subsets

Incremental lifting

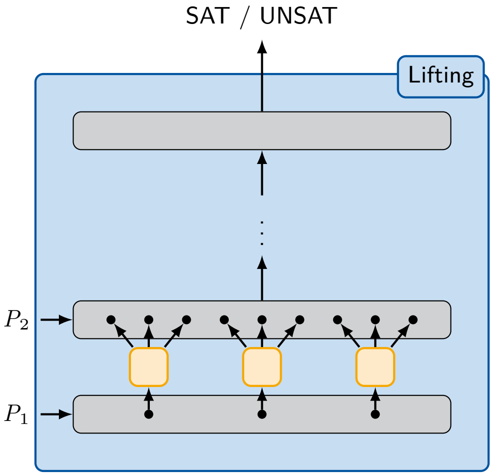
[JSC 2020]



[Collins + 1991]

Incremental lifting

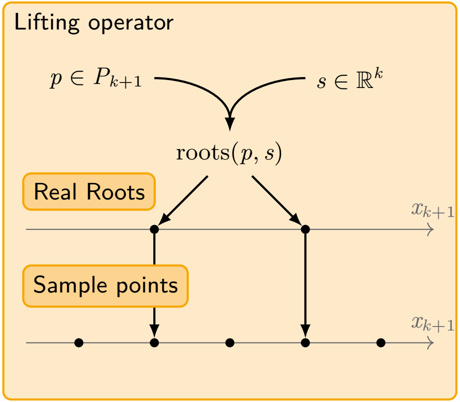
[JSC 2020]

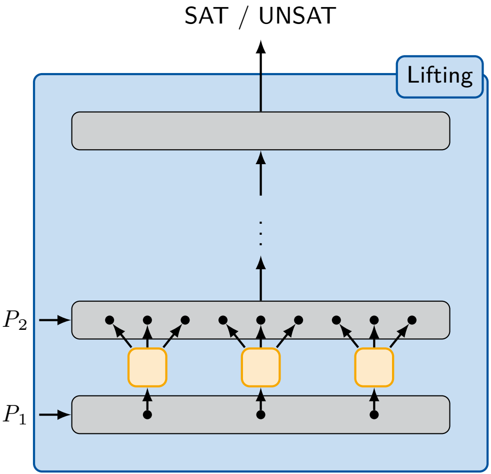


[Collins + 1991]

Incremental lifting

[JSC 2020]

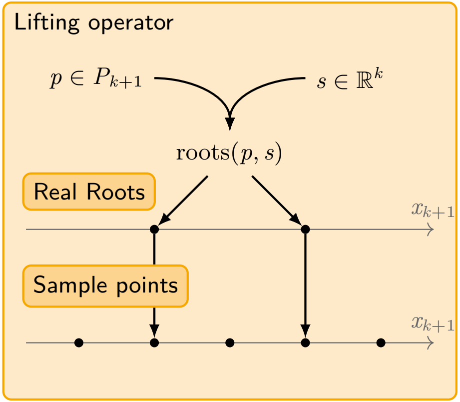


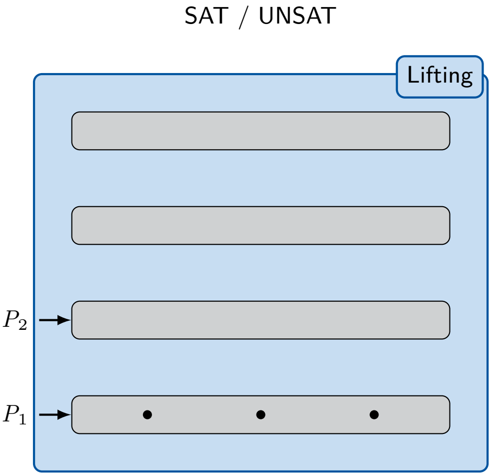


[Collins + 1991]

Incremental lifting

[JSC 2020]

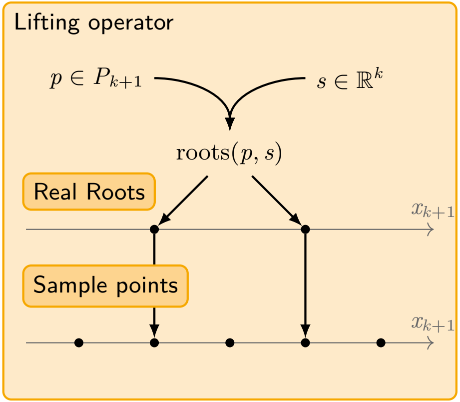




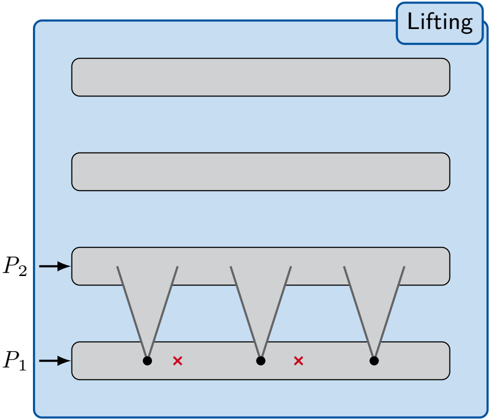
[Collins + 1991]

Incremental lifting

[JSC 2020]



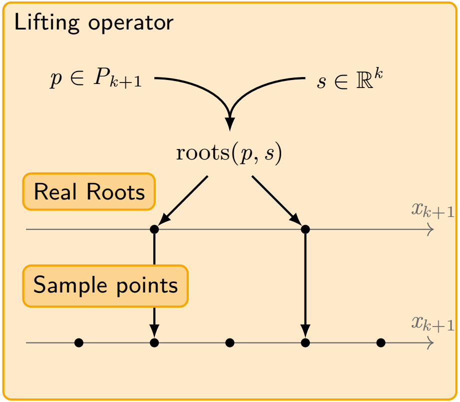
SAT / UNSAT



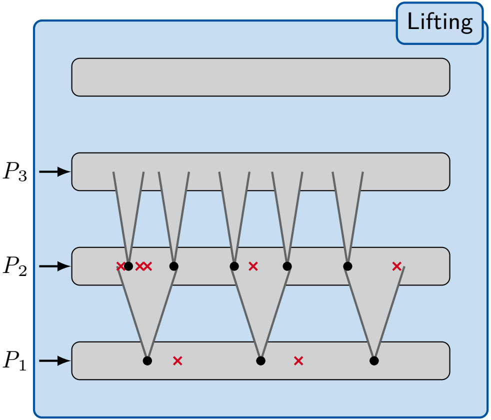
[Collins + 1991]

Incremental lifting

[JSC 2020]



SAT / UNSAT

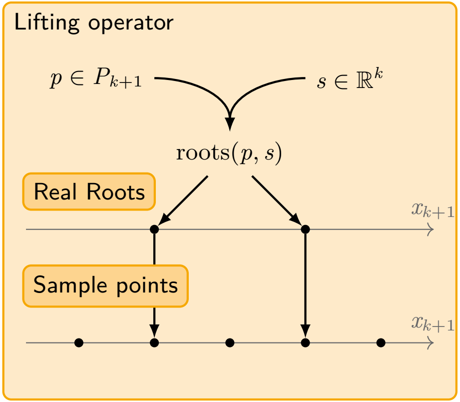


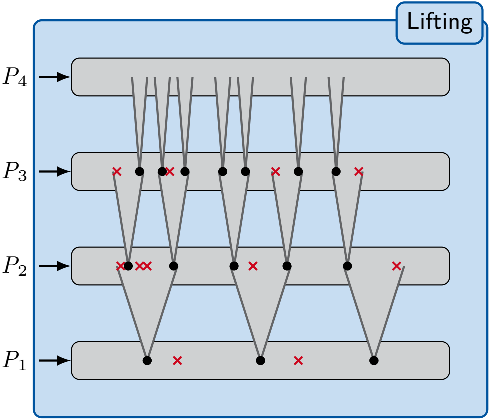
[Collins + 1991]

Incremental lifting

[JSC 2020]

SAT / UNSAT

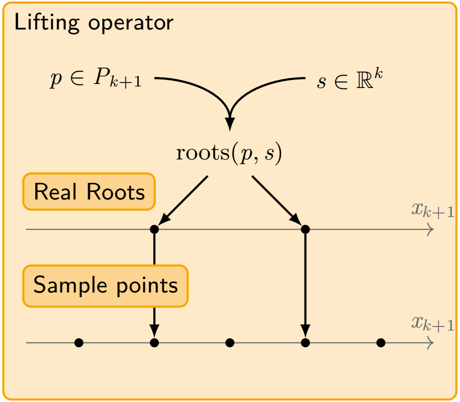


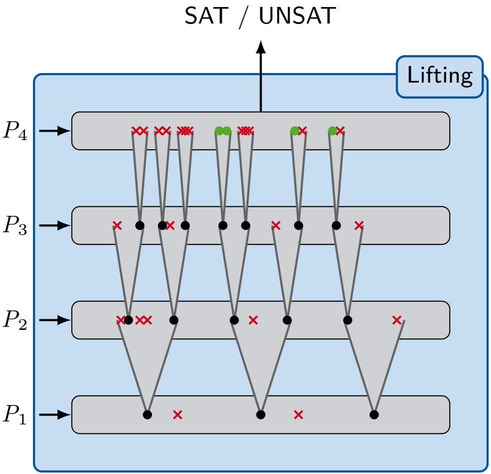


[Collins + 1991]

Incremental lifting

[JSC 2020]

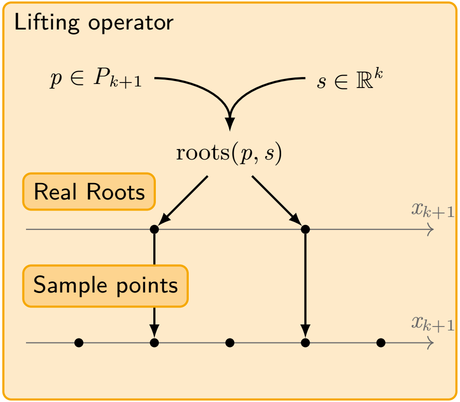




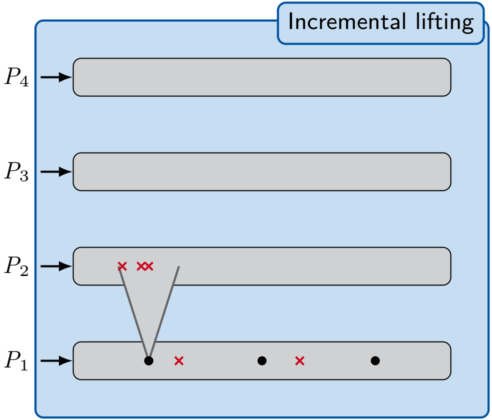
[Collins + 1991]

Incremental lifting

[JSC 2020]



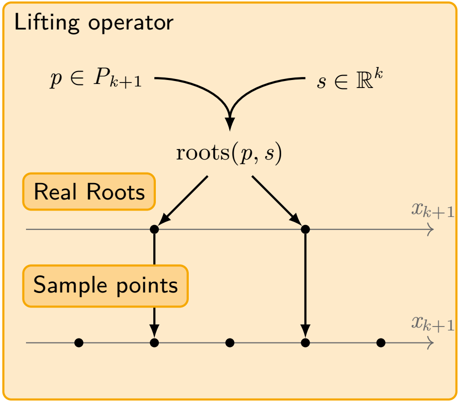
SAT / UNSAT



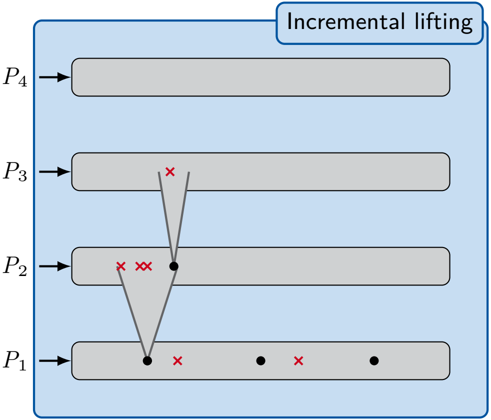
[Collins+ 1991]

Incremental lifting

[JSC 2020]



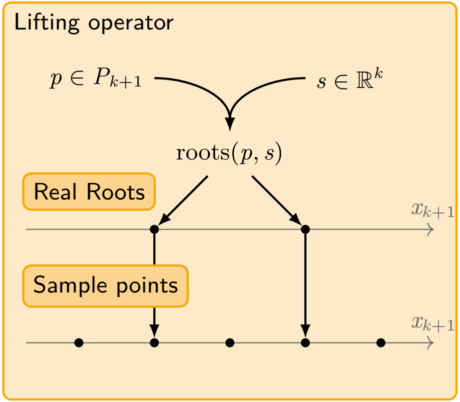
SAT / UNSAT



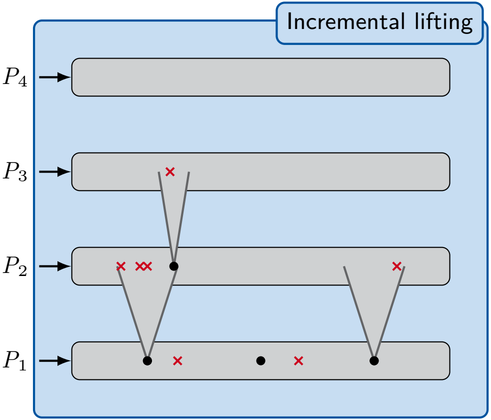
[Collins + 1991]

Incremental lifting

[JSC 2020]



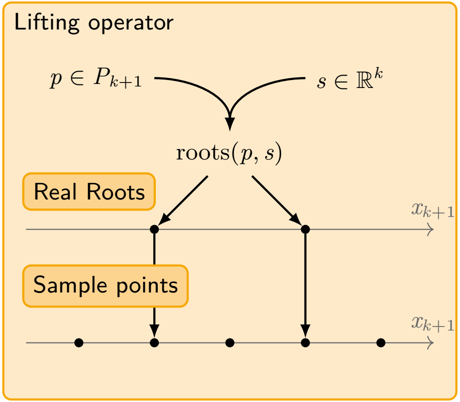
SAT / UNSAT



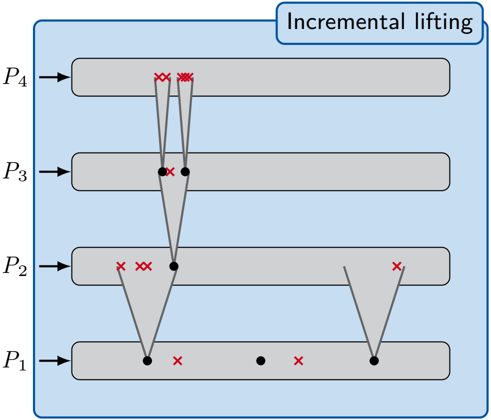
[Collins + 1991]

Incremental lifting

[JSC 2020]



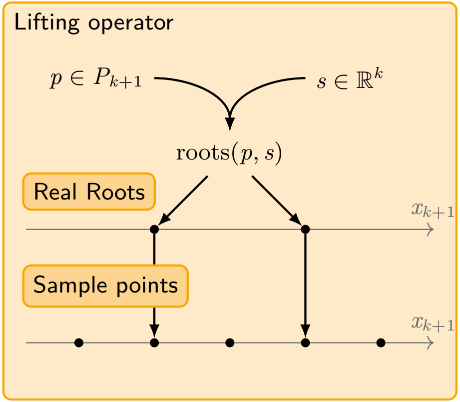
SAT / UNSAT



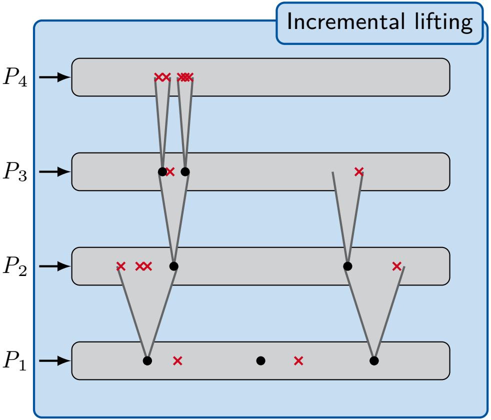
[Collins+ 1991]

Incremental lifting

[JSC 2020]



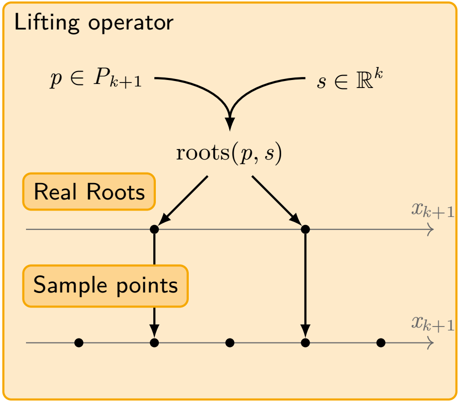
SAT / UNSAT



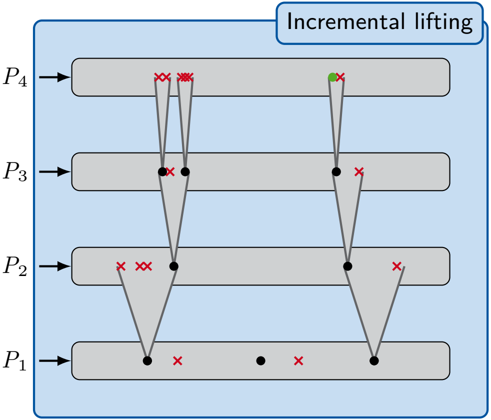
[Collins + 1991]

Incremental lifting

[JSC 2020]



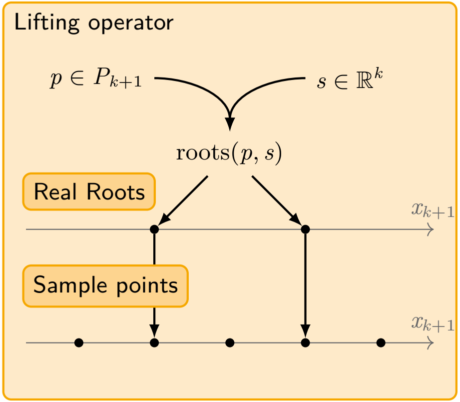
SAT / UNSAT

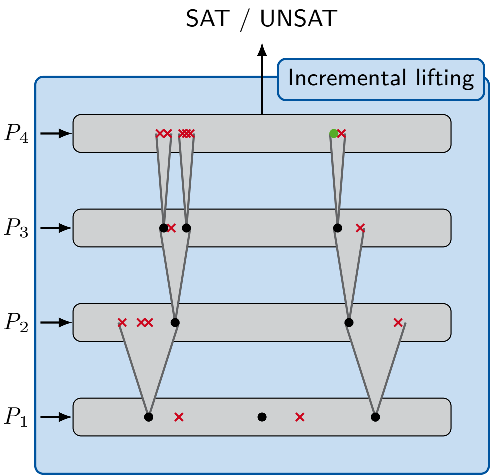


[Collins + 1991]

Incremental lifting

[JSC 2020]

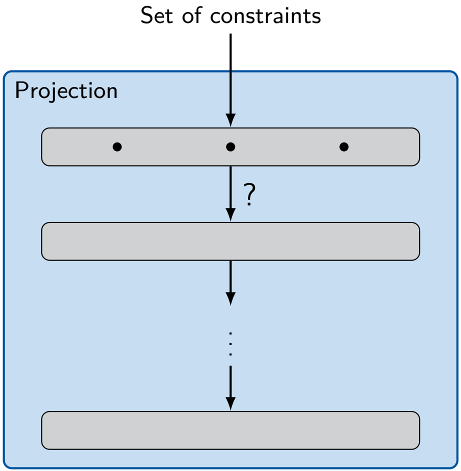




[Collins + 1991]

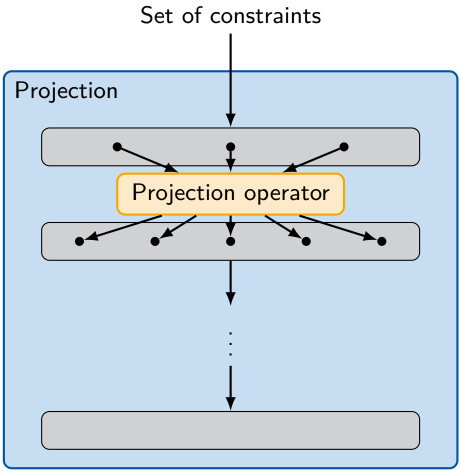
Incremental projection

[JSC 2020]



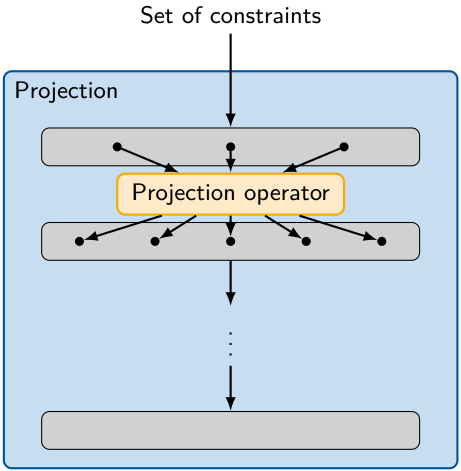
Incremental projection

[JSC 2020]



Incremental projection

[JSC 2020]

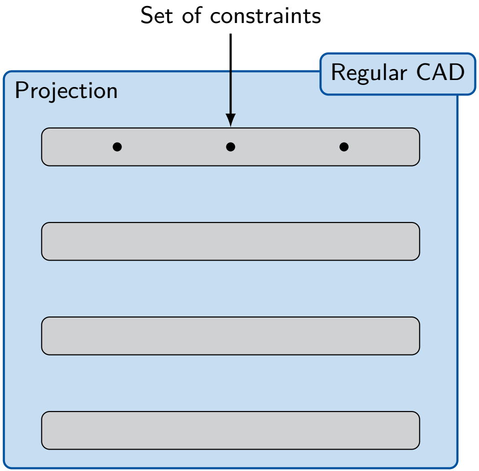


Projection operator

$$\begin{aligned}Proj(P) = &\{Proj(p) \mid p \in P\} \\ &\cup \{Proj(p, q) \mid p, q \in P\}\end{aligned}$$

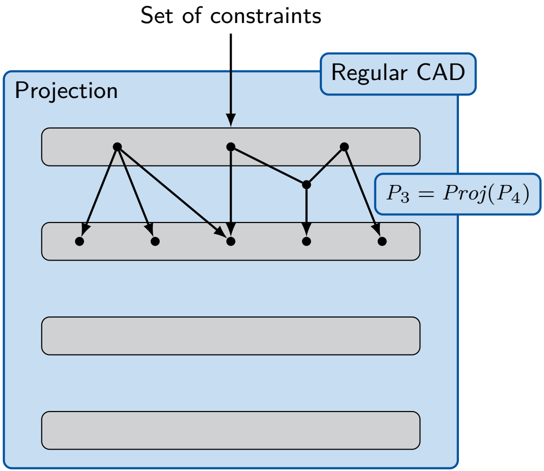
Incremental projection

[JSC 2020]



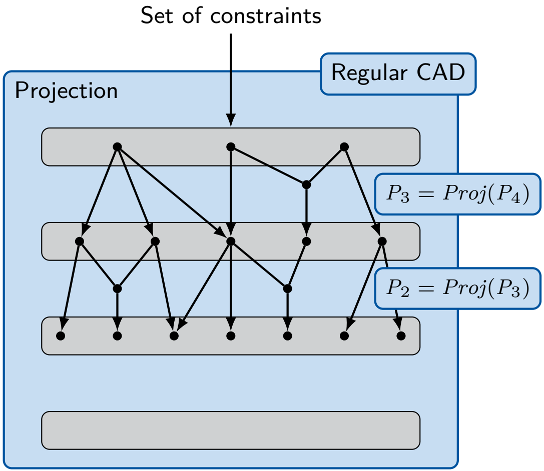
Incremental projection

[JSC 2020]



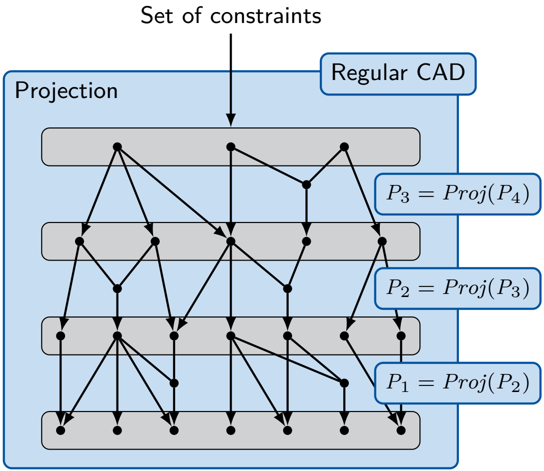
Incremental projection

[JSC 2020]



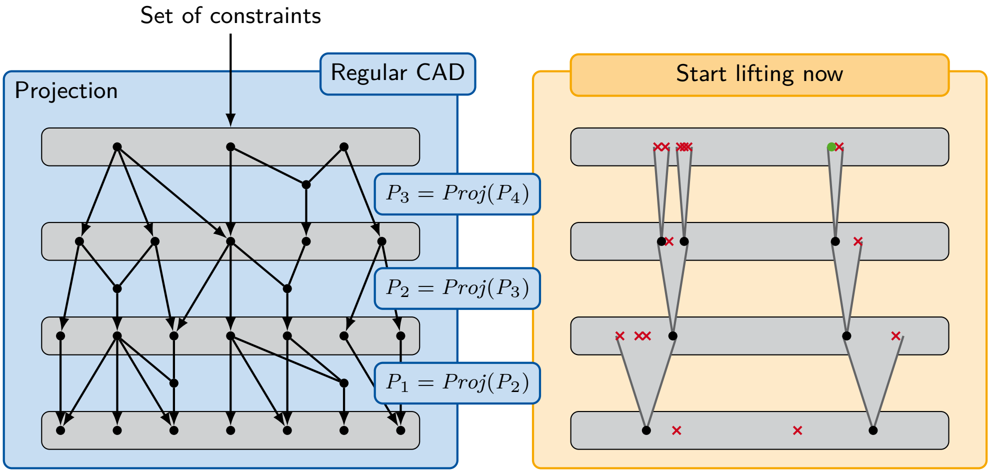
Incremental projection

[JSC 2020]



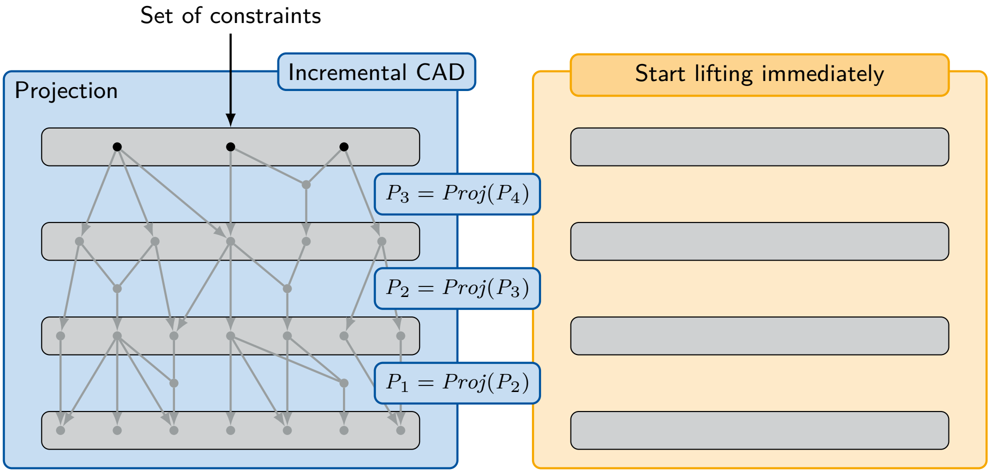
Incremental projection

[JSC 2020]



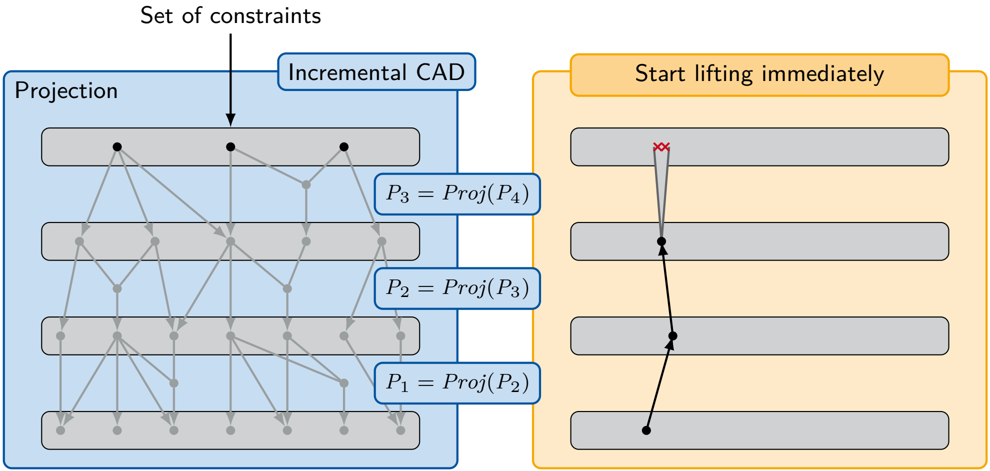
Incremental projection

[JSC 2020]



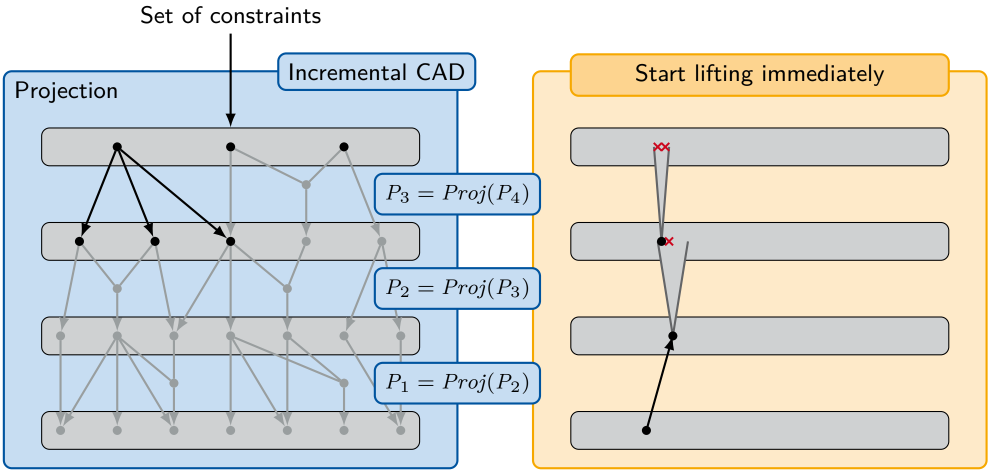
Incremental projection

[JSC 2020]



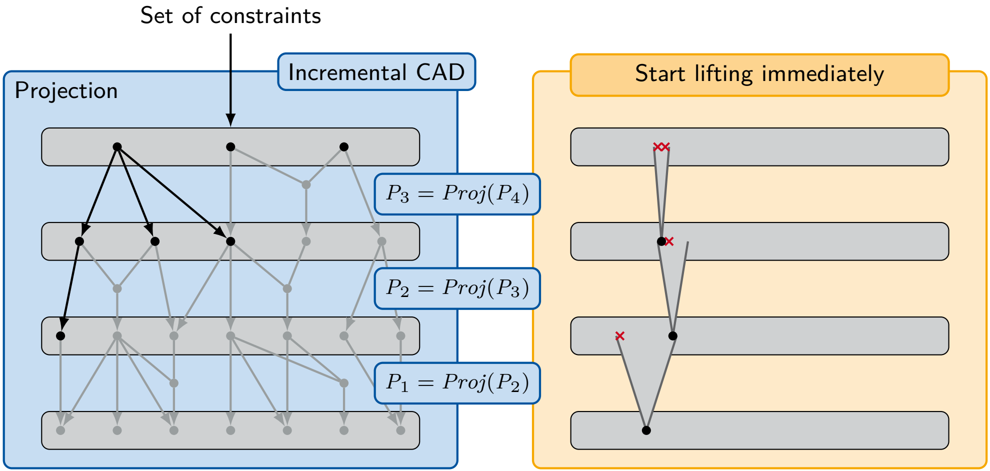
Incremental projection

[JSC 2020]



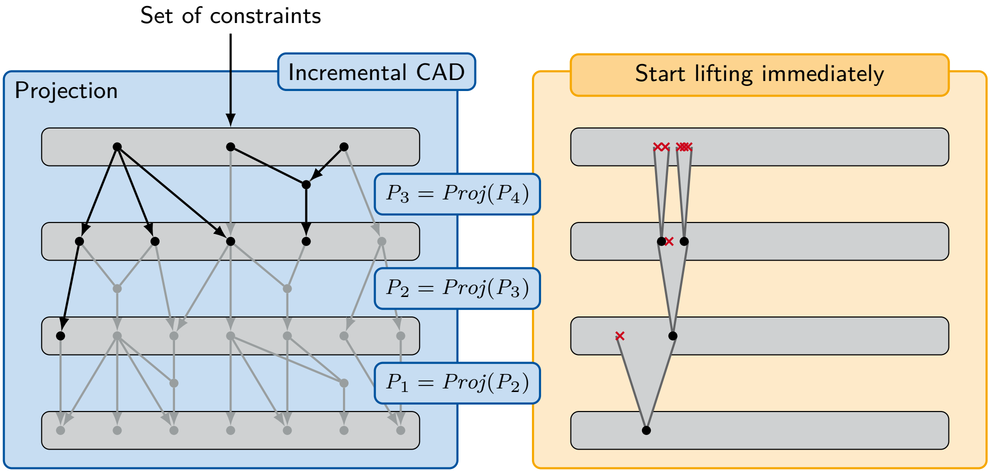
Incremental projection

[JSC 2020]



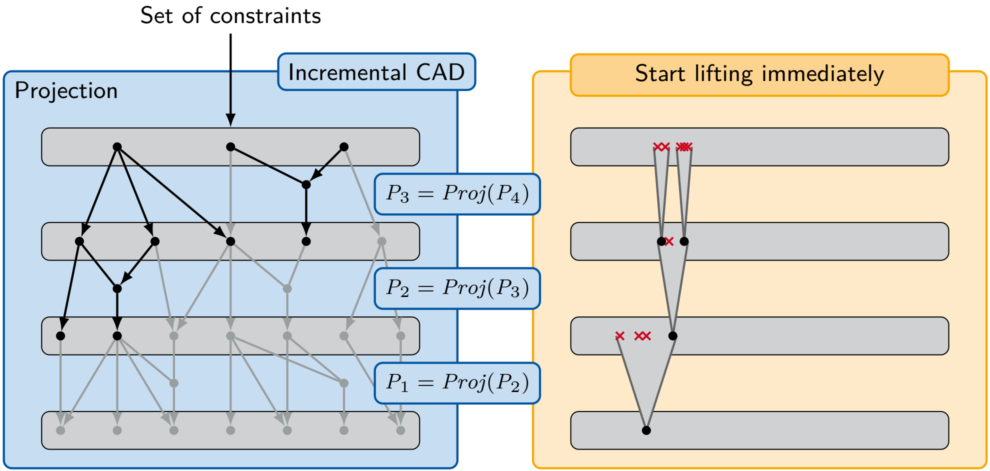
Incremental projection

[JSC 2020]



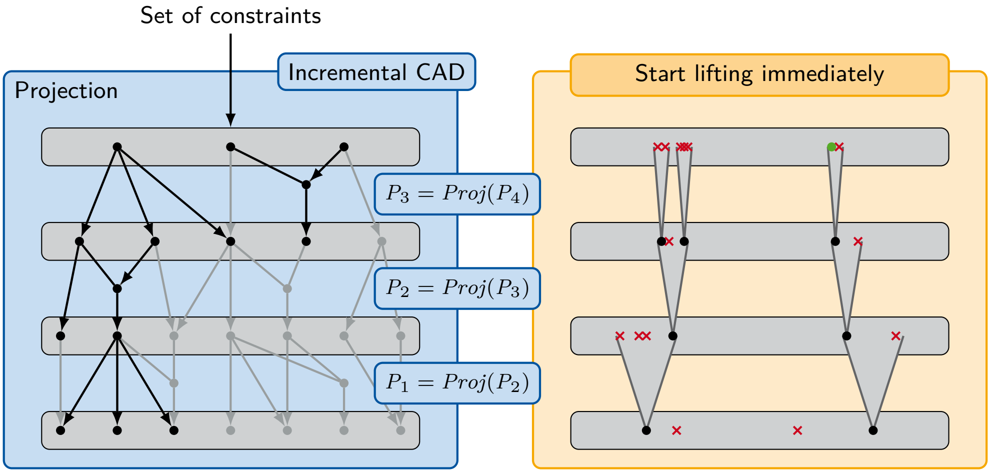
Incremental projection

[JSC 2020]



Incremental projection

[JSC 2020]



Heuristics & Additions

[JSC 2020]

- Level of incrementality
  Trade-off between potential gains and bookkeeping
- Projection operators
  Collins, Hong, McCallum, Lazard, Brown
  [SC^2 2017]
- Variable ordering
  [England + 2014] [Huang + 2014]
- Scheduling of projection and lifting
  Which polynomials and sample points to use first
- Equational constraints
  Exploits equalities to reduce projection
  [McCallum 1999] [SC^2 2018]
- Factorization of polynomials
  Ensures correctness, can reduce projection
- Minimal infeasible subsets
  Allow SAT solver to learn smaller conflicts
  [Jaroschek + 2015][Hentze 2017]

Experimental results

- Timeout: 120s
- Benchmarks: `QF_NRA` from SMT-LIB

Experimental results

- Timeout: 120s
- Benchmarks: `QF_NRA` from SMT-LIB

Incrementality

Solver	SAT	UNSAT	overall
`Naive`	4285	3496	7781
`None`	4293	3507	7800
`Simple`	4281	3889	8170
`Full`	4328	4250	8578
`Full-hide`	4328	4255	8583

- Incrementality is worth the effort
- Bookkeeping not that costly

Experimental results

- Timeout: 120s
- Benchmarks: `QF_NRA` from SMT-LIB

Incrementality

Solver	SAT	UNSAT	overall
`Naive`	4285	3496	7781
`None`	4293	3507	7800
`Simple`	4281	3889	8170
`Full`	4328	4250	8578
`Full-hide`	4328	4255	8583

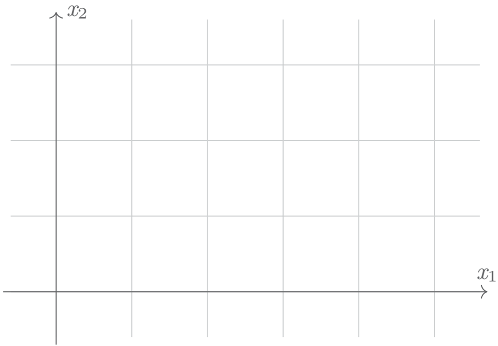
- Incrementality is worth the effort
- Bookkeeping not that costly

Projection operators

Solver	SAT	UNSAT	overall
`Collins`	4292	4150	8442
`Hong`	4301	4189	8490
`McCallum`	4320	4216	8536
`Lazard`	4322	4229	8551
`Brown`	4328	4250	8578

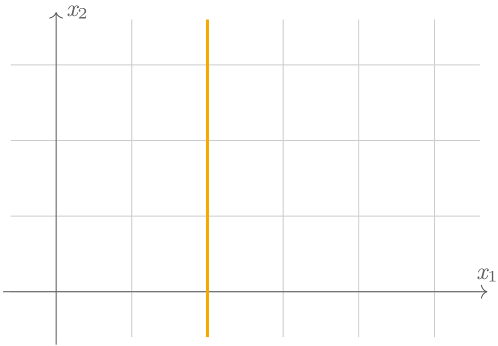
- Confirms conventional wisdom
- Margins may be surprising

Model-Constructing Satisfiability Calculus



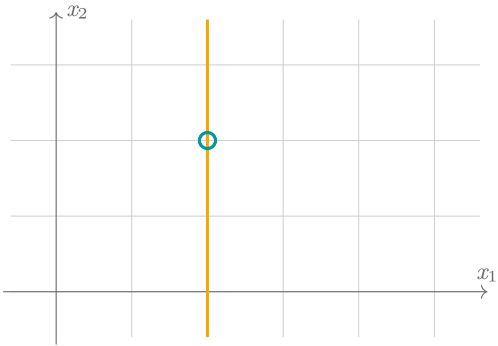
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



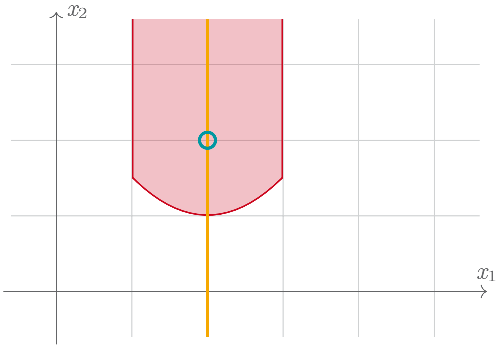
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



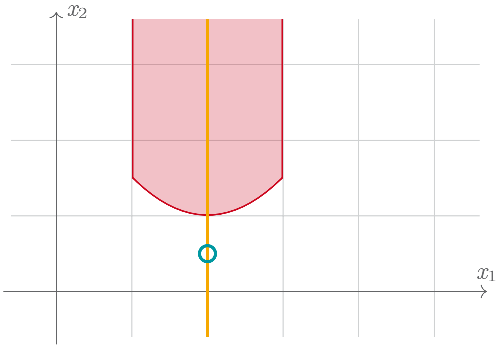
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



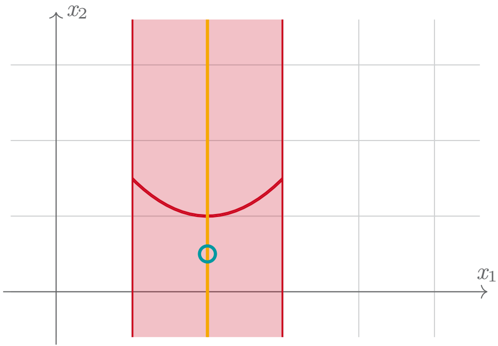
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



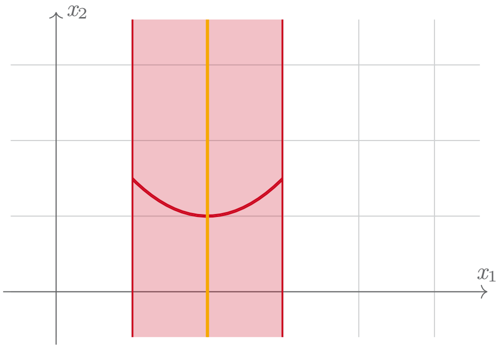
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



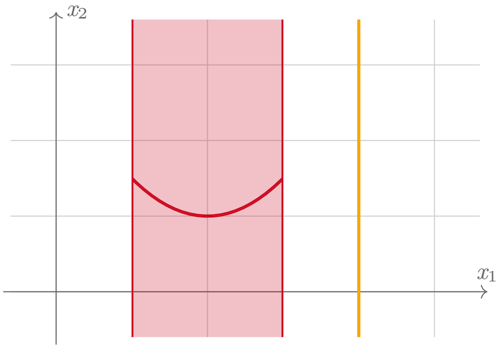
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



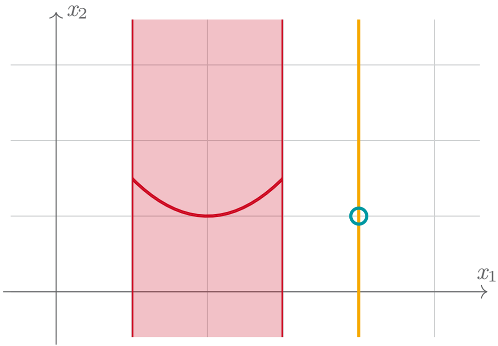
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



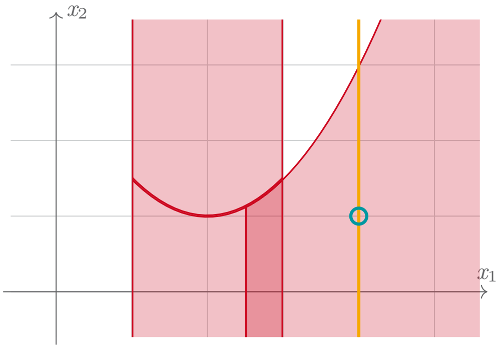
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



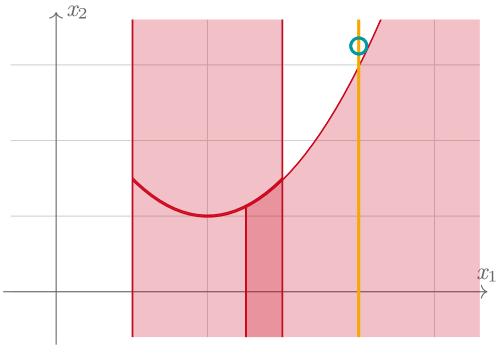
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



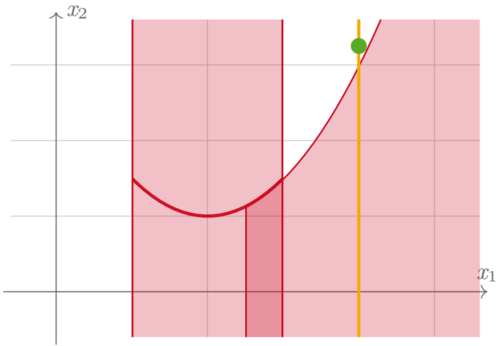
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



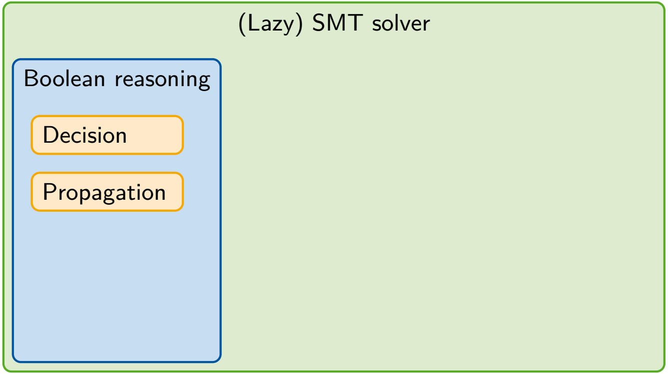
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus

(Lazy) SMT solver

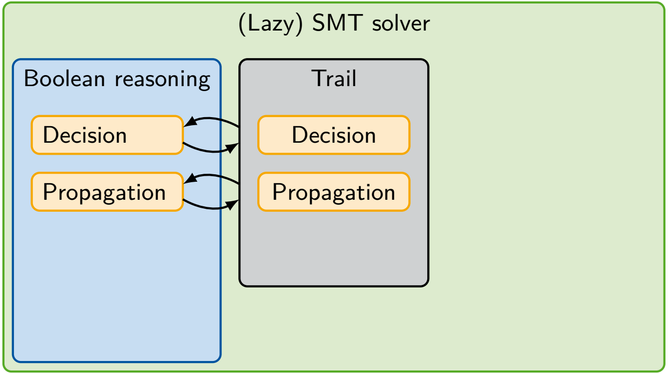
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



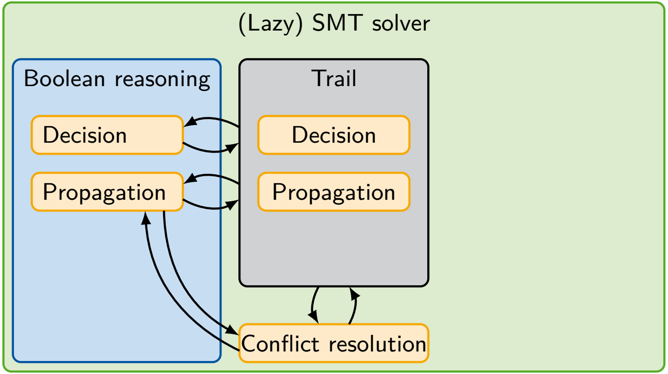
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



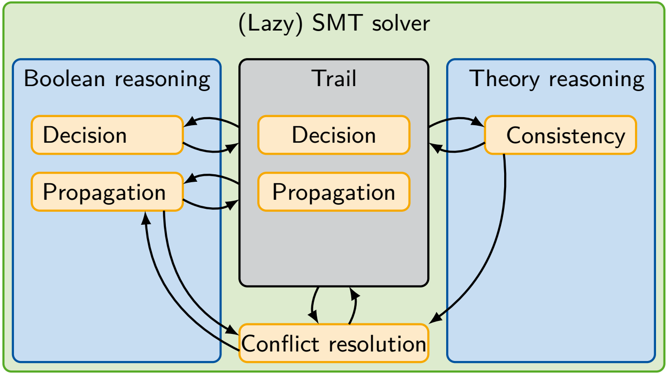
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



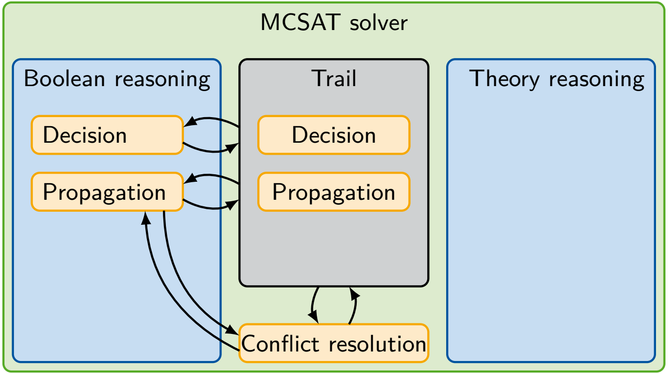
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



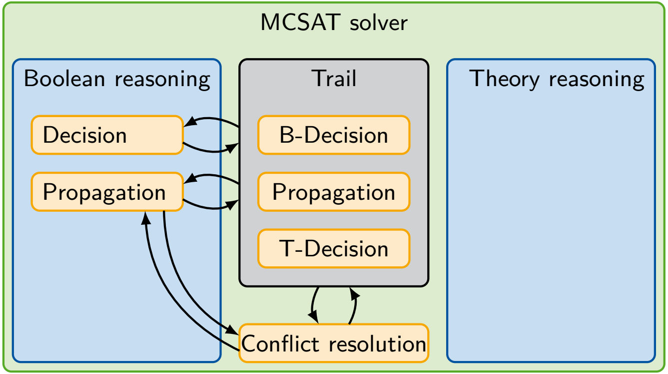
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



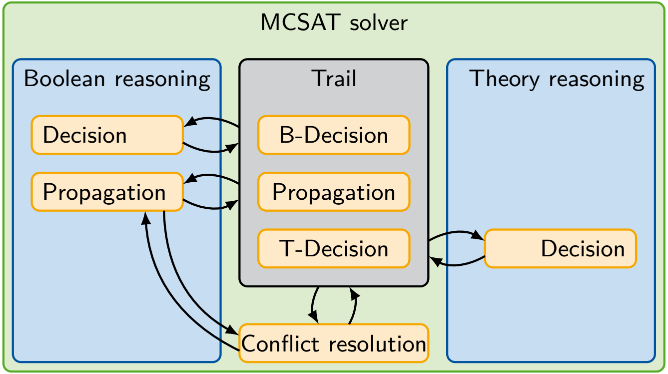
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



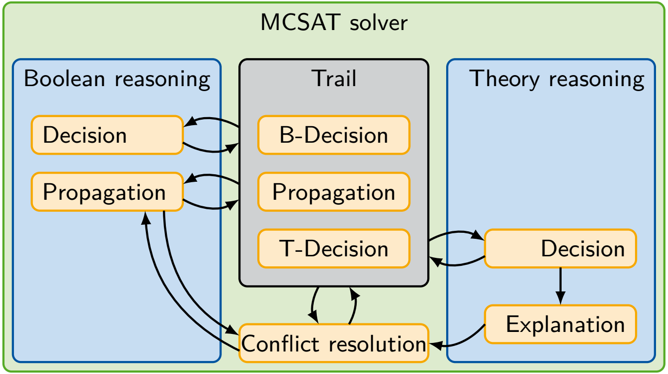
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



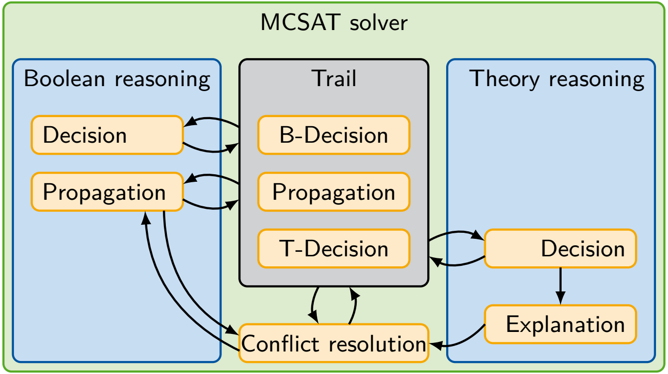
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus

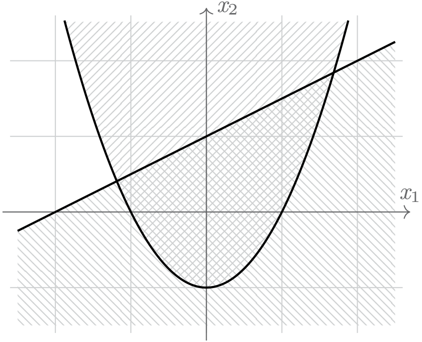


[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus

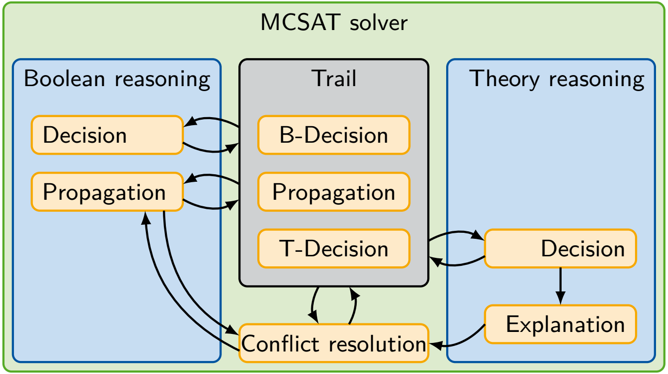


Trail: $[\![c_1, c_2]\!]$



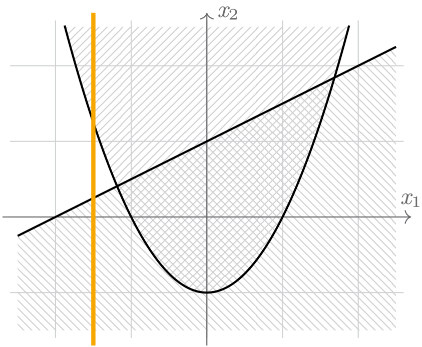
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



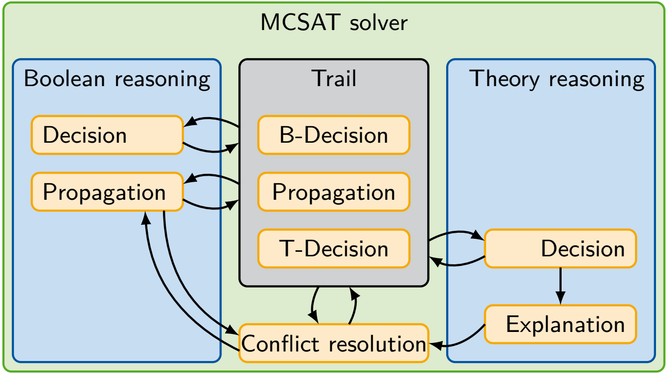
Guess: $x_1 \mapsto -1.5$

Trail: $[\![c_1, c_2, x_1 \mapsto -1.5]\!]$



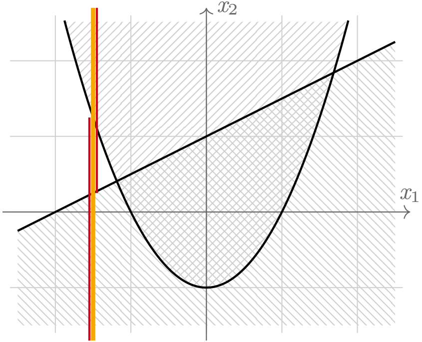
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



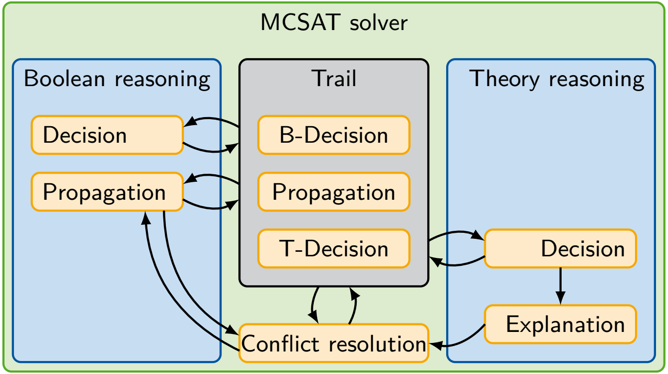
No assignment for x_2. Reason: c_1, c_2

Trail: $[\![c_1, c_2, x_1 \mapsto -1.5]\!]$



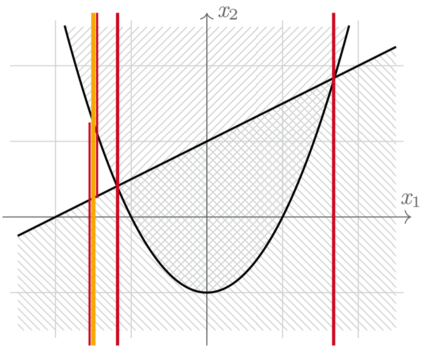
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



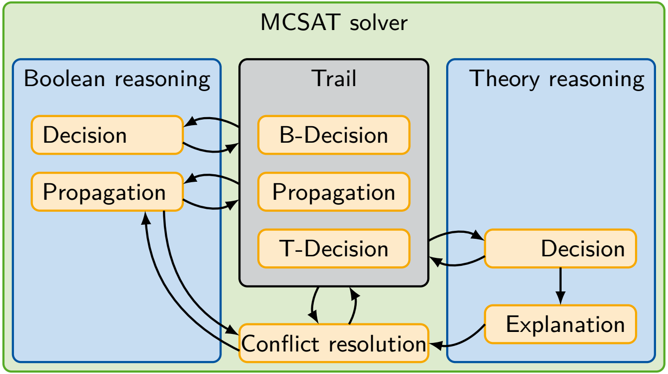
No assignment for x_2. Reason: c_1, c_2

Trail: $[\![c_1, c_2, x_1 \mapsto -1.5]\!]$



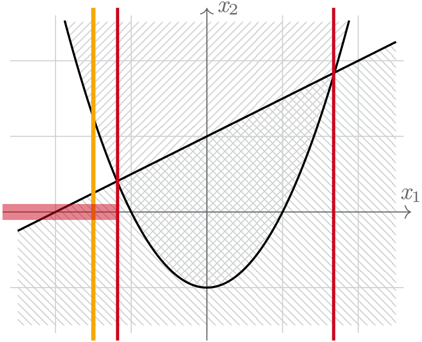
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



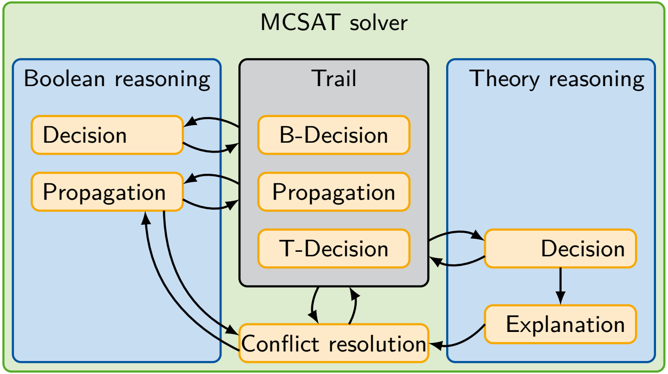
No assignment for x_2. Reason: c_1, c_2

Trail: $[\![c_1, c_2, x_1 \mapsto -1.5]\!]$



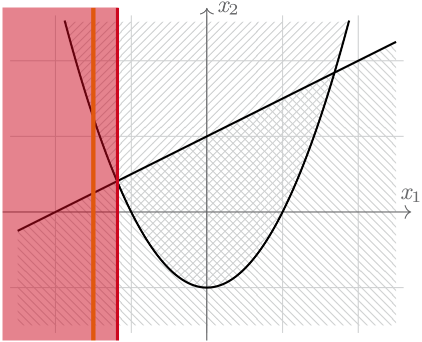
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



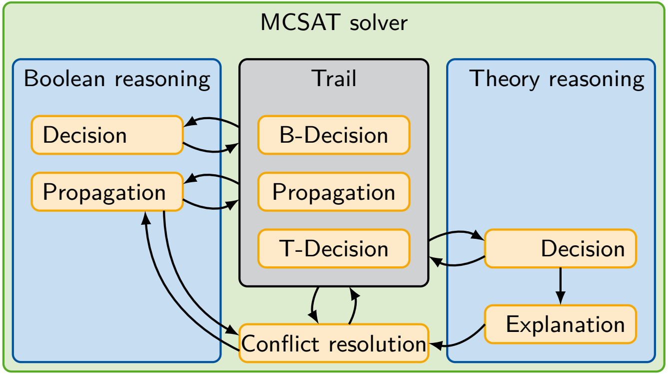
$(c_1 \wedge c_2) \rightarrow (\alpha_2 \leq x_1)$

Trail: $[\![c_1, c_2]\!]$



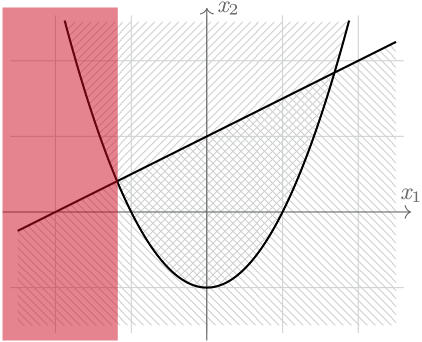
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



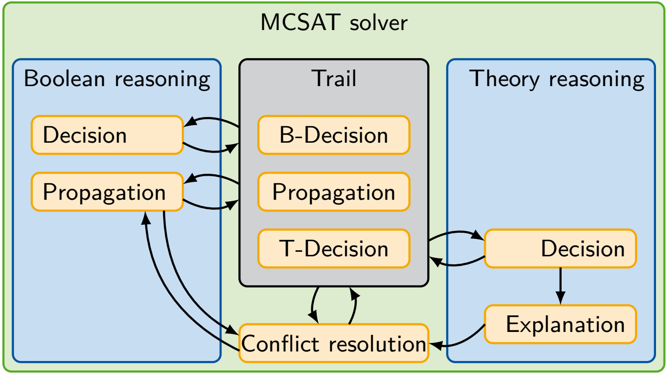
Conflict resolution

Trail: $[\![c_1, c_2]\!]$



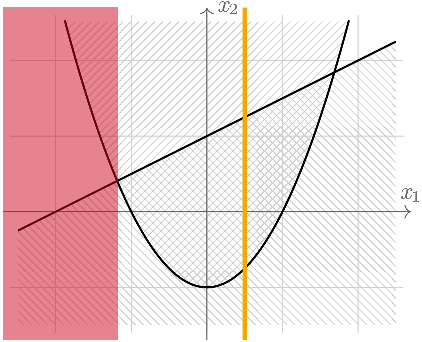
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



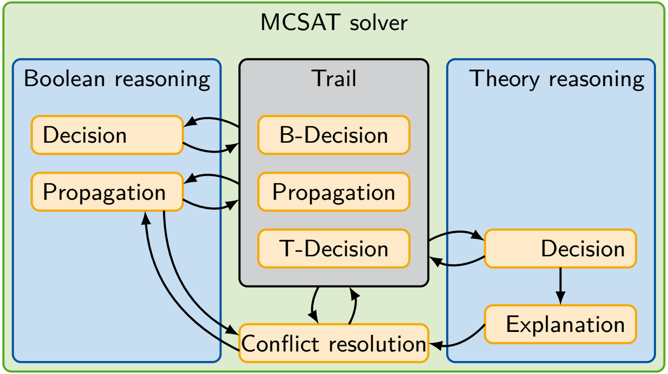
Guess: $x_1 \mapsto 0.5$

Trail: $[\![c_1, c_2, x_1 \mapsto 0.5]\!]$



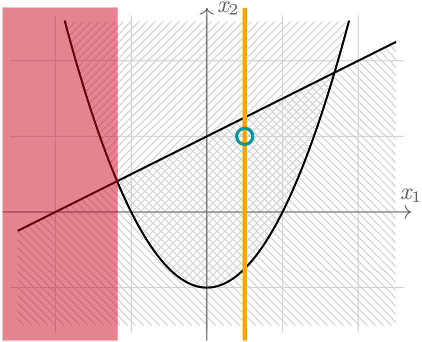
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



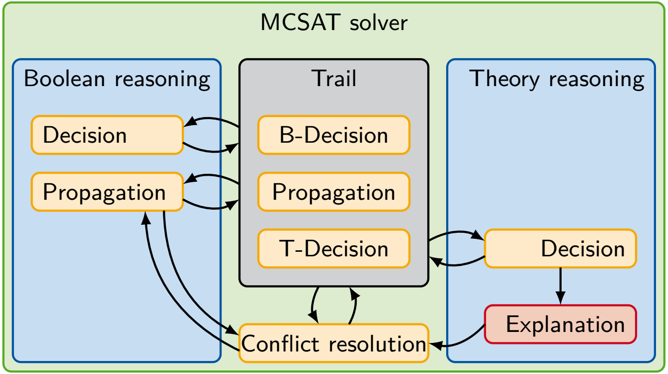
Guess: $x_2 \mapsto 1$

Trail: $[\![c_1, c_2, x_1 \mapsto 0.5, x_2 \mapsto 1]\!]$



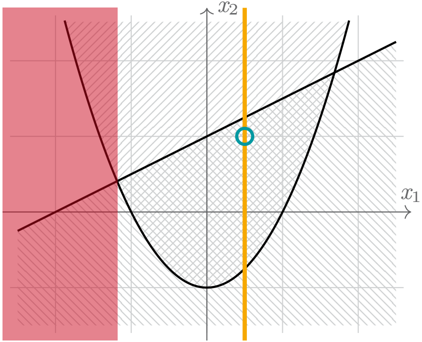
[Jovanović + 2012] [Moura + 2013]

Model-Constructing Satisfiability Calculus



Guess: $x_2 \mapsto 1$

Trail: $[\![c_1, c_2, x_1 \mapsto 0.5, x_2 \mapsto 1]\!]$



[Jovanović + 2012] [Moura + 2013]

Explanation functions

Definition (Explanation function)

Let $\mathcal{A} = \{x_1 \mapsto \alpha_1, \dots, x_k \mapsto \alpha_k\}$ and $C = \{c_1, \dots\}$ constraints over $x_1, \dots x_{k+1}$.
Assume that $\nexists\, \alpha_{k+1}.\{x_1 \mapsto \alpha_1, \dots x_{k+1} \mapsto \alpha_{k+1}\} \models C$.

Explanation functions

Definition (Explanation function)

Let $\mathcal{A} = \{x_1 \mapsto \alpha_1, \ldots, x_k \mapsto \alpha_k\}$ and $C = \{c_1, \ldots\}$ constraints over $x_1, \ldots x_{k+1}$.
Assume that $\nexists\, \alpha_{k+1}.\{x_1 \mapsto \alpha_1, \ldots x_{k+1} \mapsto \alpha_{k+1}\} \models C$.
An explanation function constructs a lemma:

$$C \rightarrow (\overline{x} \notin \text{Cell around } \mathcal{A})$$

Explanation functions

Definition (Explanation function)

Let $\mathcal{A} = \{x_1 \mapsto \alpha_1, \dots, x_k \mapsto \alpha_k\}$ and $C = \{c_1, \dots\}$ constraints over $x_1, \dots x_{k+1}$.
Assume that $\nexists\, \alpha_{k+1}.\{x_1 \mapsto \alpha_1, \dots x_{k+1} \mapsto \alpha_{k+1}\} \models C$.
An explanation function constructs a lemma:

$$C \rightarrow (\overline{x} \notin \text{Cell around } \mathcal{A})$$

$$\text{Conflict in } \mathbb{R}^{k+1} \rightarrow \text{Lemma in } \mathbb{R}^{k}$$

Explanation functions

Definition (Explanation function)

Let $\mathcal{A} = \{x_1 \mapsto \alpha_1, \dots, x_k \mapsto \alpha_k\}$ and $C = \{c_1, \dots\}$ constraints over $x_1, \dots x_{k+1}$.
Assume that $\nexists\, \alpha_{k+1}.\{x_1 \mapsto \alpha_1, \dots x_{k+1} \mapsto \alpha_{k+1}\} \models C$.
An explanation function constructs a lemma:

$$C \rightarrow (\overline{x} \notin \text{Cell around } \mathcal{A})$$

$$\text{Conflict in } \mathbb{R}^{k+1} \rightarrow \text{Lemma in } \mathbb{R}^{k}$$

We need to get rid of x_{k+1}: Quantifier elimination

Explanation functions

Definition (Explanation function)

Let $\mathcal{A} = \{x_1 \mapsto \alpha_1, \dots, x_k \mapsto \alpha_k\}$ and $C = \{c_1, \dots\}$ constraints over $x_1, \dots x_{k+1}$.
Assume that $\nexists\, \alpha_{k+1}.\{x_1 \mapsto \alpha_1, \dots x_{k+1} \mapsto \alpha_{k+1}\} \models C$.
An explanation function constructs a lemma:

$$C \rightarrow (\overline{x} \notin \text{Cell around } \mathcal{A})$$

$$\text{Conflict in } \mathbb{R}^{k+1} \rightarrow \text{Lemma in } \mathbb{R}^{k}$$

We need to get rid of x_{k+1}: Quantifier elimination

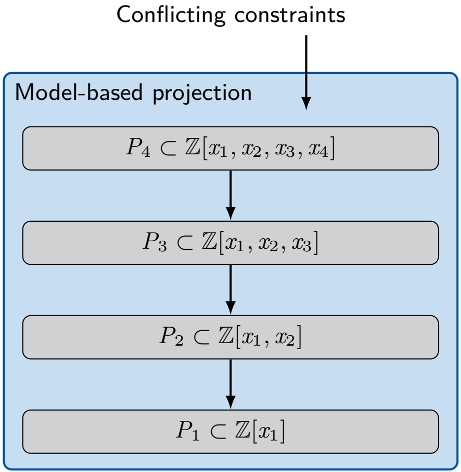
Model-based CAD [Jovanović + 2012], Fourier-Motzkin [Jovanović + 2013]

CAD-based / NLSAT

Conflicting constraints

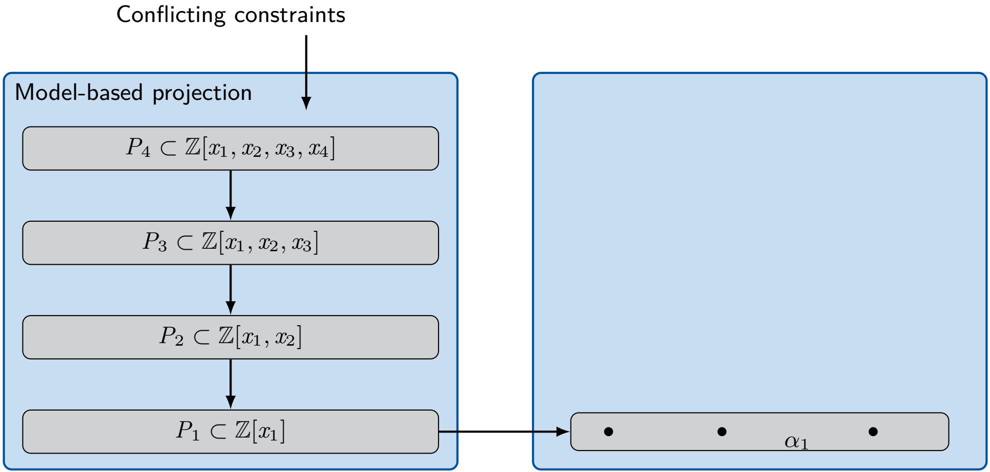
[Jovanović + 2012]

CAD-based / NLSAT



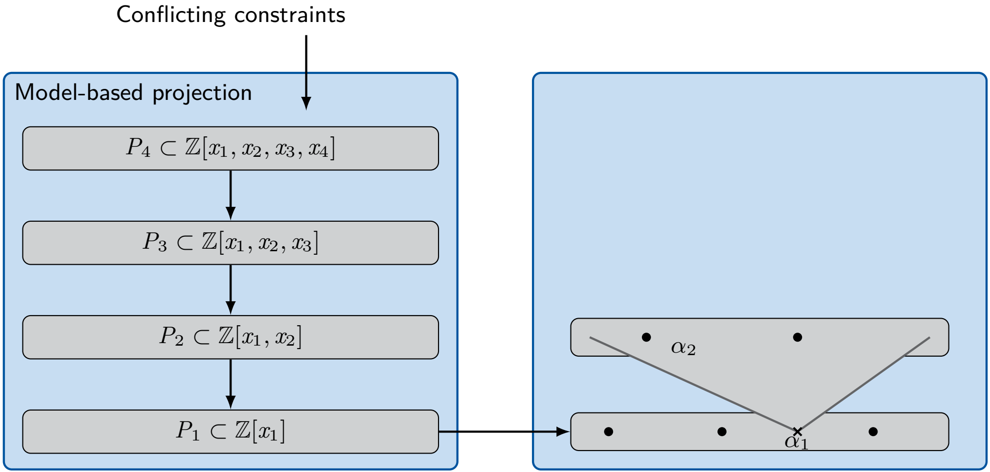
[Jovanović + 2012]

CAD-based / NLSAT



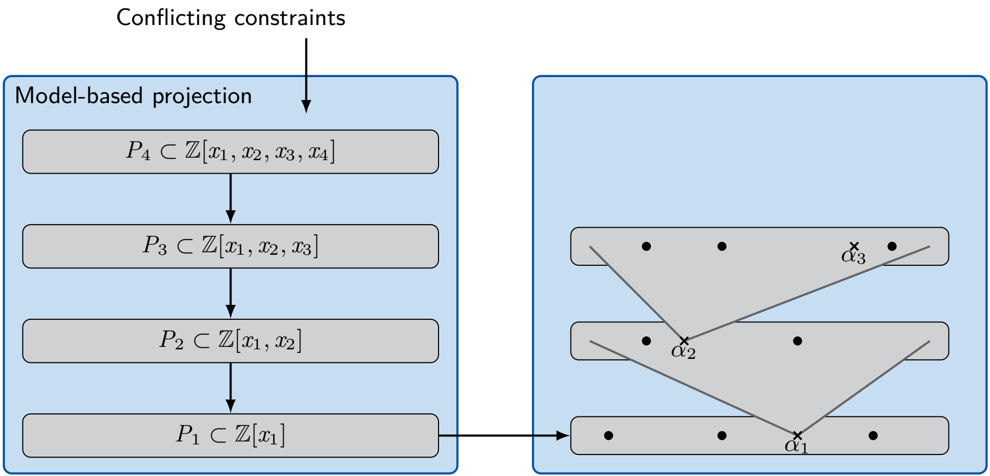
[Jovanović + 2012]

CAD-based / NLSAT



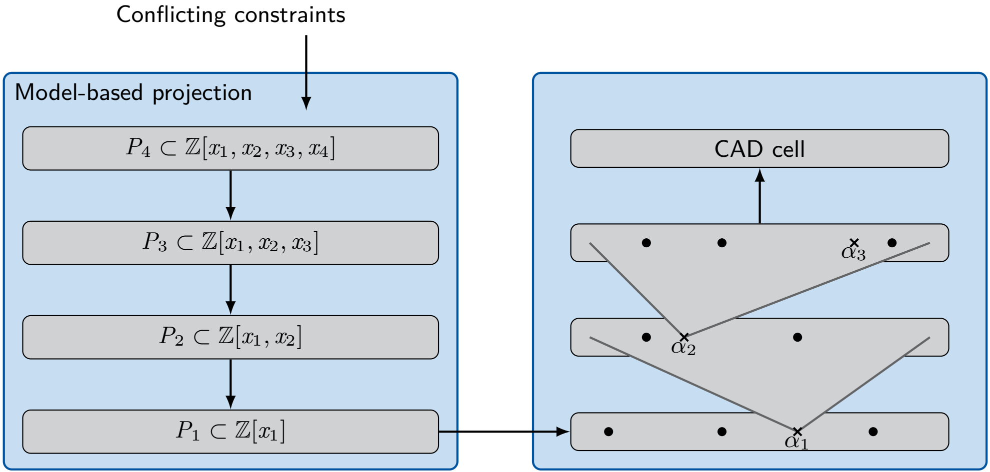
[Jovanović + 2012]

CAD-based / NLSAT



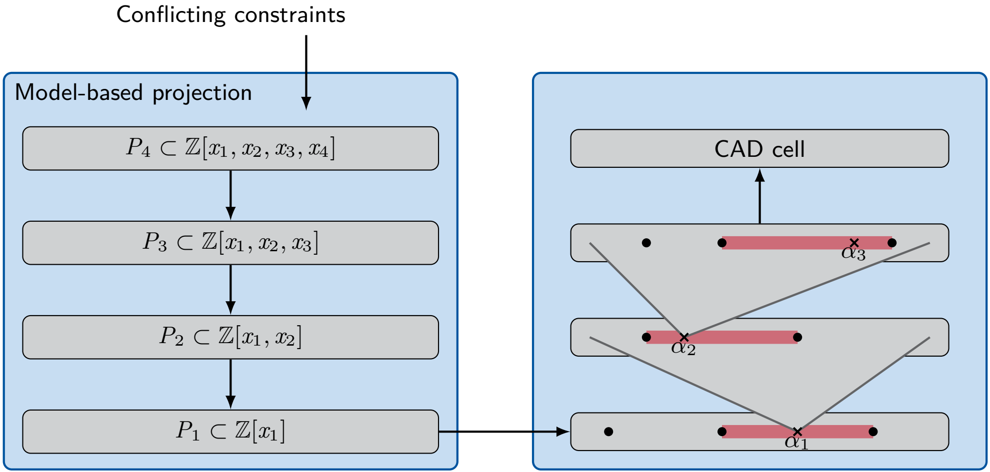
[Jovanović + 2012]

CAD-based / NLSAT



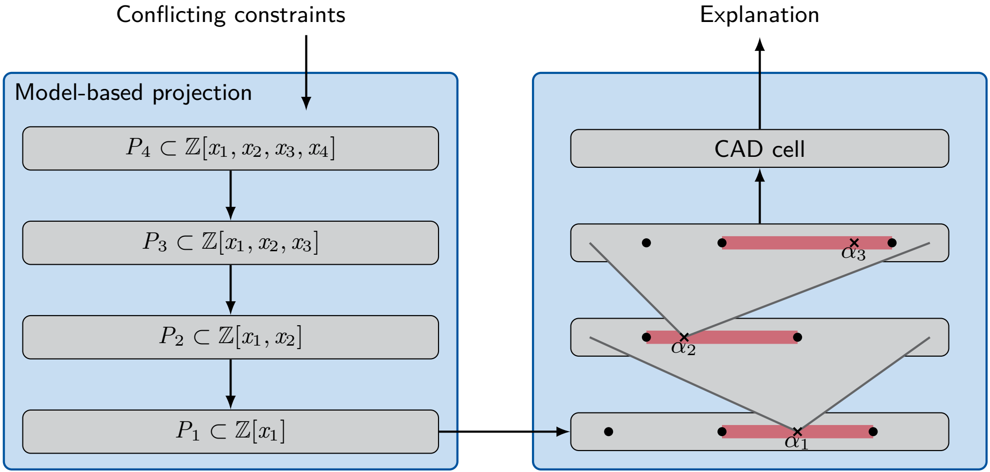
[Jovanović + 2012]

CAD-based / NLSAT



[Jovanović + 2012]

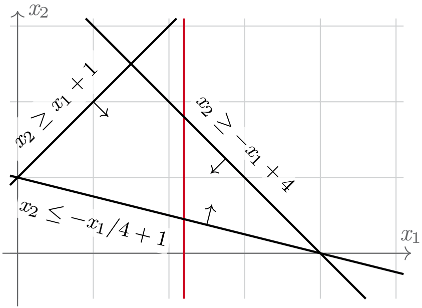
CAD-based / NLSAT



[Jovanović + 2012]

Fourier-Motzkin

[Bartolomé 2018]

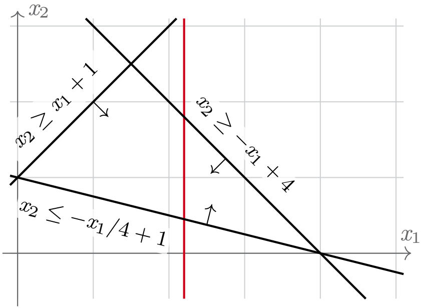


[Jovanović + 2013]

Fourier-Motzkin

[Bartolomé 2018]

Conflicting constraints



[Jovanović + 2013]

Fourier-Motzkin

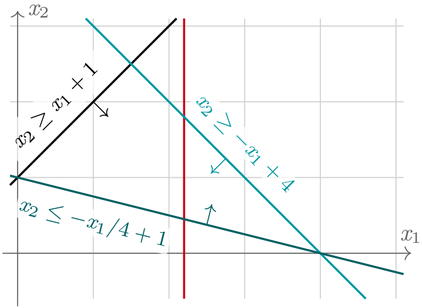
[Bartolomé 2018]

Conflicting constraints

↓

Identify lower and upper bounds

$-x_1 + 4 \leq \quad x_2 \quad \leq -x_1/4 + 1$



[Jovanović + 2013]

Fourier-Motzkin

[Bartolomé 2018]

Conflicting constraints

↓

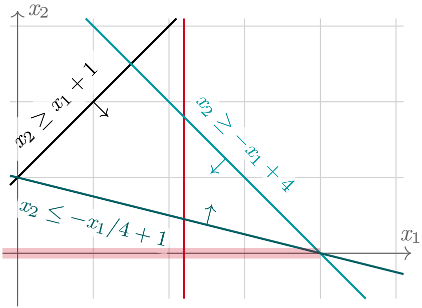
Identify lower and upper bounds

$-x_1 + 4 \leq \quad x_2 \quad \leq -x_1/4 + 1$

↓

Combine bounds

$4 \leq x_1$



[Jovanović + 2013]

Fourier-Motzkin

[Bartolomé 2018]

Conflicting constraints

↓

Identify lower and upper bounds

$-x_1 + 4 \leq \quad x_2 \quad \leq -x_1/4 + 1$

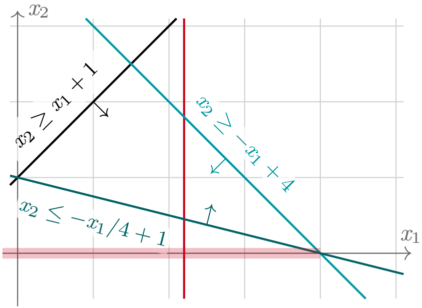
↓

Combine bounds

$4 \leq x_1$

↓

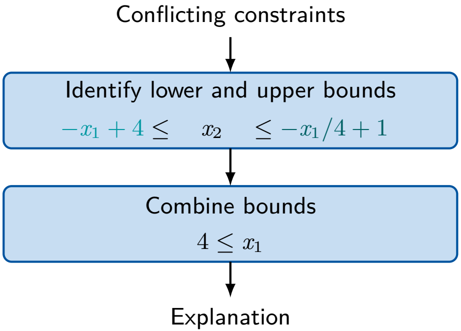
Explanation

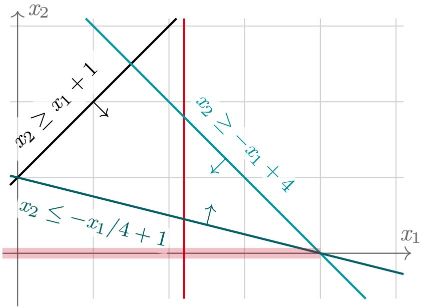


[Jovanović + 2013]

Fourier-Motzkin

[Bartolomé 2018]





- Strict inequalities $\quad 2x_1 + 1 < x_2 < x_1 + 2 \quad x_1 \mapsto 2$
- Disequalities $\quad 2x_1 + 1 \leq x_2 \leq x_1 + 2 \land x_2 \neq 3 \quad x_1 \mapsto 1$
- Nonlinear coefficients $\quad x_1^2 + 3 \leq x_2 \leq 2x_1 \quad x_1 \mapsto 1$

[Jovanović + 2013]

Other explanation functions

Virtual cubstitution [SC^2 2017]

- Nonlinear but low-degree
- Flexible dimensionality of generated cells

[Brown 2013] [Brown + 2015]

Other explanation functions

Virtual cubstitution [SC^2 2017]

- Nonlinear but low-degree
- Flexible dimensionality of generated cells

OneCell [Neuß 2018]

- CAD tailored to construct a single cell
- Larger cells (and thus excludes more)

[Brown 2013] [Brown + 2015]

Other explanation functions

Virtual cubstitution [SC² 2017]

- Nonlinear but low-degree
- Flexible dimensionality of generated cells

OneCell [Neuß 2018]

- CAD tailored to construct a single cell
- Larger cells (and thus excludes more)

Interval constraint propagation

- Novel adaption of ICP for MCSAT
- Disregards theory model

[Brown 2013] [Brown + 2015]

Other explanation functions

Virtual cubstitution [SC^2 2017]

- Nonlinear but low-degree
- Flexible dimensionality of generated cells

OneCell [Neuß 2018]

- CAD tailored to construct a single cell
- Larger cells (and thus excludes more)

Interval constraint propagation

- Novel adaption of ICP for MCSAT
- Disregards theory model

Composition of explanation functions

- `SMT-RAT`: modular composition
- Sequential & parallel

[Brown 2013] [Brown$^+$ 2015]

Other explanation functions

Virtual cubstitution [SC² 2017]

- Nonlinear but low-degree
- Flexible dimensionality of generated cells

OneCell [Neuß 2018]

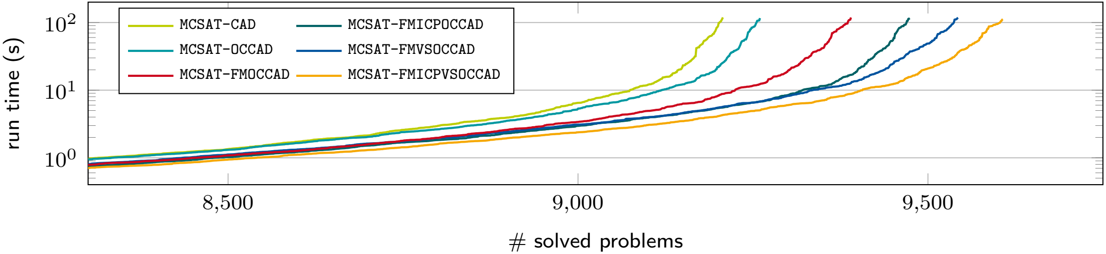
- CAD tailored to construct a single cell
- Larger cells (and thus excludes more)

Interval constraint propagation

- Novel adaption of ICP for MCSAT
- Disregards theory model

Composition of explanation functions

- SMT-RAT: modular composition
- Sequential & parallel



[Brown 2013] [Brown + 2015]

Theory variable ordering

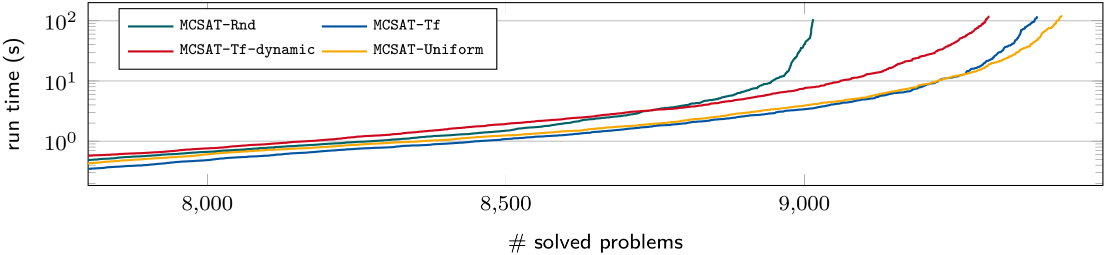
[SC^2 2019]

- Well-known: variable ordering is crucial for CAD, VS, FM, ...
- Usually: static ordering for theory reasoning
- Goal: use dynamic ordering for MCSAT [Jovanović$^+$ 2013]

Theory variable ordering

[SC^2 2019]

- Well-known: variable ordering is crucial for CAD, VS, FM, ...
- Usually: static ordering for theory reasoning
- Goal: use dynamic ordering for MCSAT [Jovanović + 2013]
- Observation: `MCSAT-Tf` and `MCSAT-Uniform` perform best
- But: `MCSAT-Tf-dynamic` is worse



Comparison of lazy SMT and MCSAT

[SC² 2019]

- Two similar proof systems: lazy SMT and MCSAT
- Experience: MCSAT better for nonlinear
- Goal: theoretic comparison of lazy SMT and MCSAT

Comparison of lazy SMT and MCSAT

[SC² 2019]

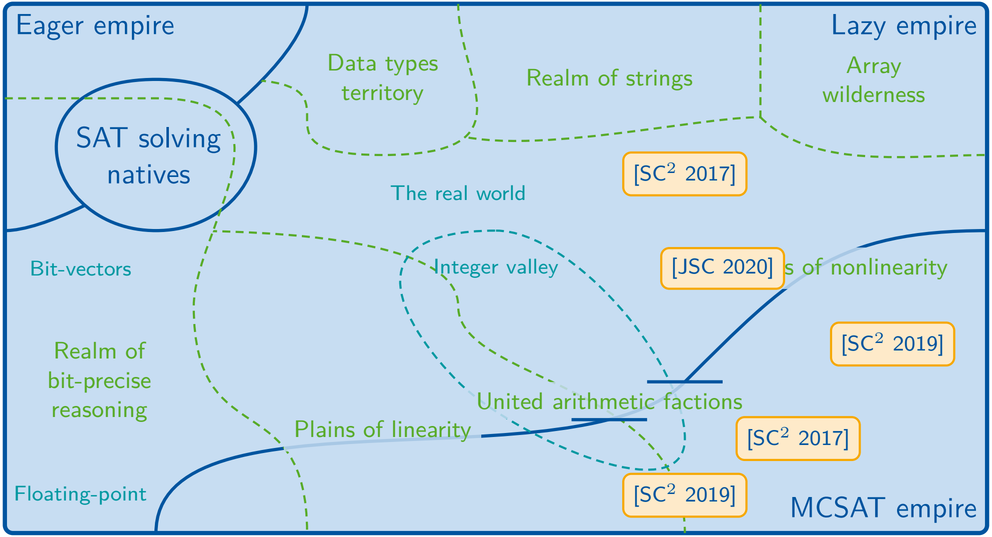
- Two similar proof systems: lazy SMT and MCSAT
- Experience: MCSAT better for nonlinear
- Goal: theoretic comparison of lazy SMT and MCSAT

Theorem

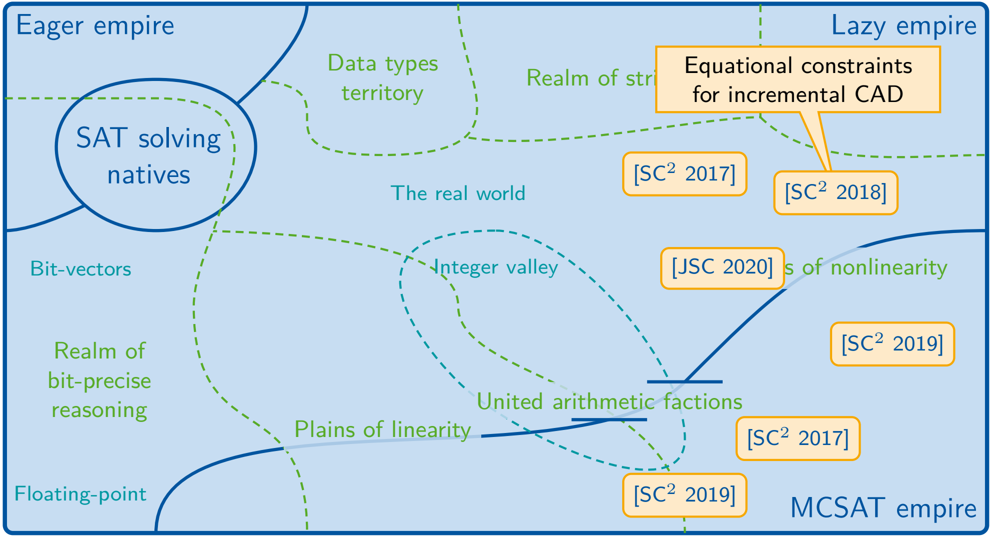
Lazy SMT and MCSAT are algorithmically equivalent.*

*: terms and conditions apply

Summary

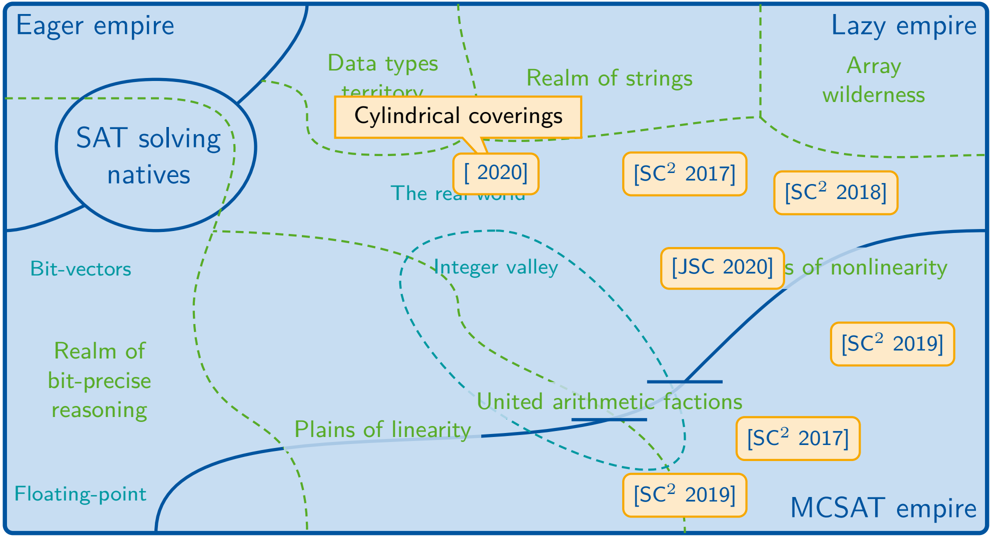

Eager empire
Lazy empire
Data types territory
Realm of strings
Array wilderness
SAT solving natives
The real world
[SC² 2017]
Bit-vectors
Integer valley
[JSC 2020]
s of nonlinearity
Realm of bit-precise reasoning
[SC² 2019]
United arithmetic factions
Plains of linearity
[SC² 2017]
Floating-point
[SC² 2019]
MCSAT empire

Summary

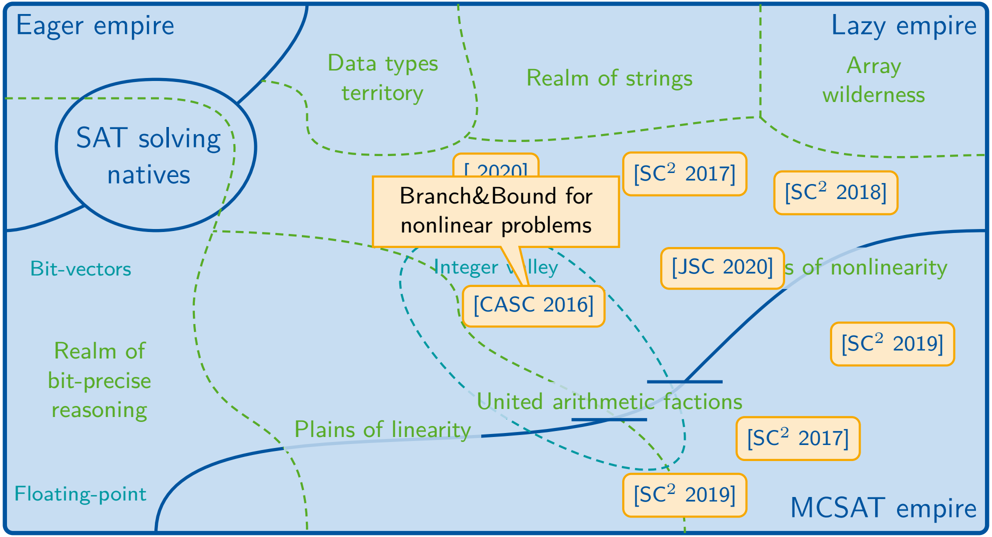


Eager empire
Lazy empire
Data types territory
Realm of stri
Equational constraints for incremental CAD
SAT solving natives
[SC^2 2017]
[SC^2 2018]
The real world
Bit-vectors
Integer valley
[JSC 2020]
s of nonlinearity
[SC^2 2019]
Realm of bit-precise reasoning
United arithmetic factions
Plains of linearity
[SC^2 2017]
Floating-point
[SC^2 2019]
MCSAT empire

Summary

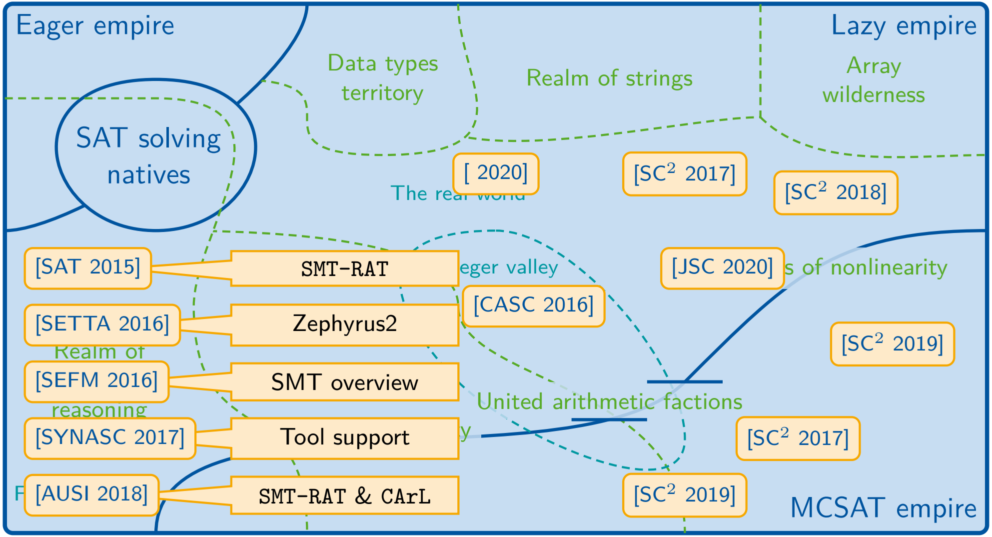

Eager empire
Lazy empire
Data types territory
Realm of strings
Array wilderness
SAT solving natives
Cylindrical coverings
[2020]
[SC² 2017]
[SC² 2018]
The real world
Bit-vectors
Integer valley
[JSC 2020]
s of nonlinearity
[SC² 2019]
Realm of bit-precise reasoning
United arithmetic factions
Plains of linearity
[SC² 2017]
Floating-point
[SC² 2019]
MCSAT empire

Summary

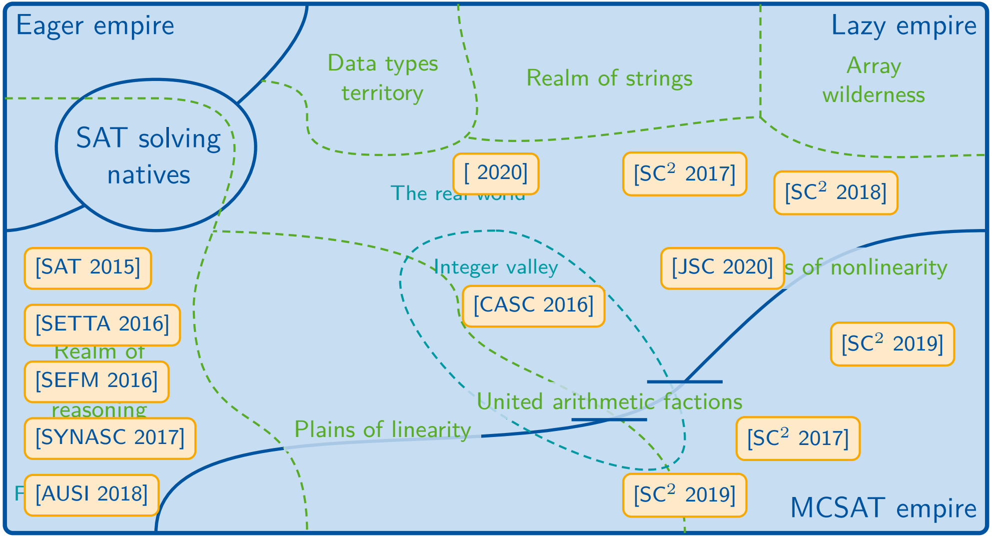


Eager empire
Lazy empire
Data types territory
Realm of strings
Array wilderness
SAT solving natives
[2020]
[SC² 2017]
[SC² 2018]
Branch&Bound for nonlinear problems
Bit-vectors
Integer valley
[JSC 2020]
s of nonlinearity
[CASC 2016]
[SC² 2019]
Realm of bit-precise reasoning
United arithmetic factions
Plains of linearity
[SC² 2017]
Floating-point
[SC² 2019]
MCSAT empire

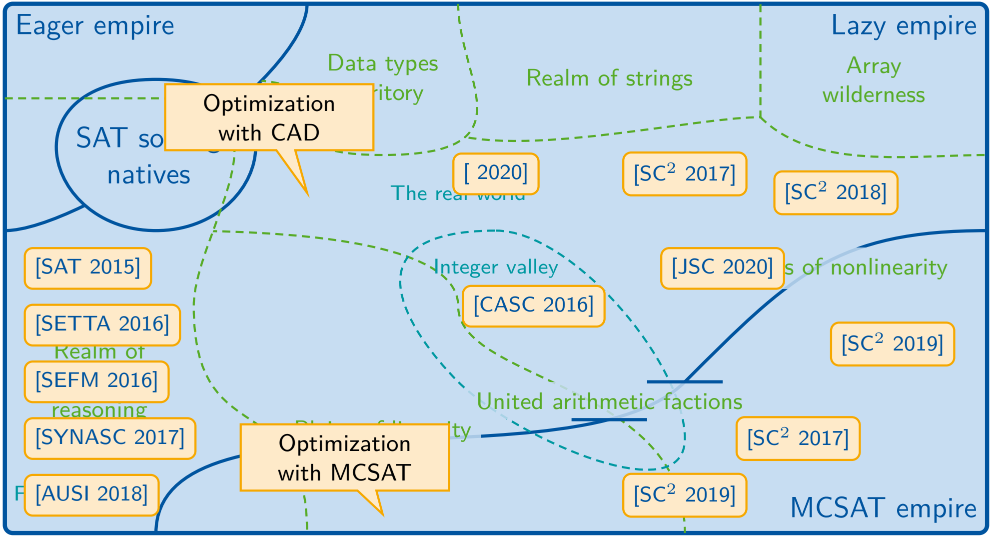
Summary


Eager empire
Lazy empire
Data types territory
Realm of strings
Array wilderness
SAT solving natives
[2020]
[SC^2 2017]
[SC^2 2018]
The real world
eger valley
[JSC 2020]
s of nonlinearity
[SAT 2015]
SMT-RAT
[SETTA 2016]
Zephyrus2
[CASC 2016]
Realm of
[SEFM 2016]
SMT overview
reasoning
[SC^2 2019]
United arithmetic factions
[SYNASC 2017]
Tool support
[SC^2 2017]
[AUSI 2018]
SMT-RAT & CArL
[SC^2 2019]
MCSAT empire

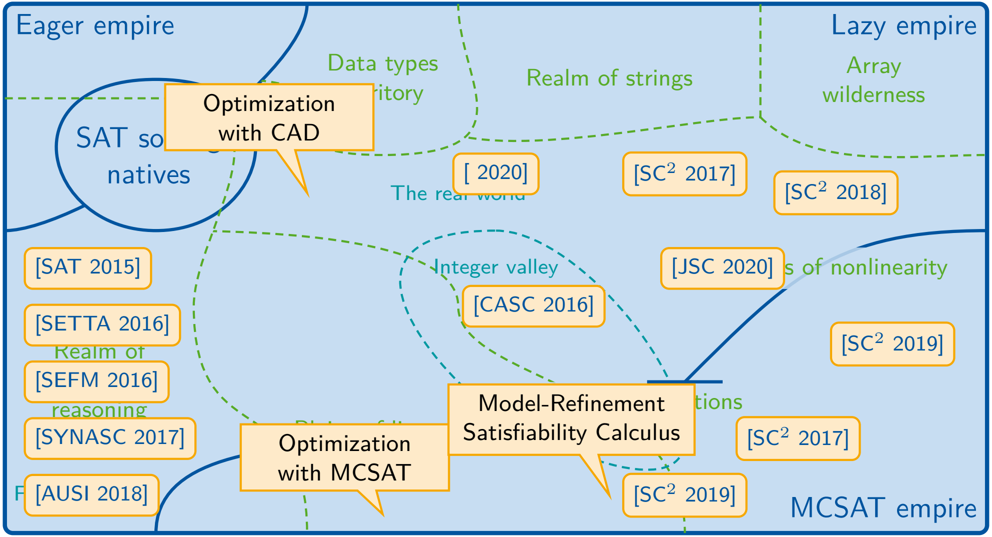
Summary


Eager empire
Lazy empire
Data types territory
Realm of strings
Array wilderness
SAT solving natives
[2020]
[SC² 2017]
[SC² 2018]
The real world
[SAT 2015]
Integer valley
[JSC 2020]
s of nonlinearity
[CASC 2016]
[SETTA 2016]
Realm of
[SC² 2019]
[SEFM 2016]
reasoning
United arithmetic factions
Plains of linearity
[SYNASC 2017]
[SC² 2017]
[AUSI 2018]
[SC² 2019]
MCSAT empire

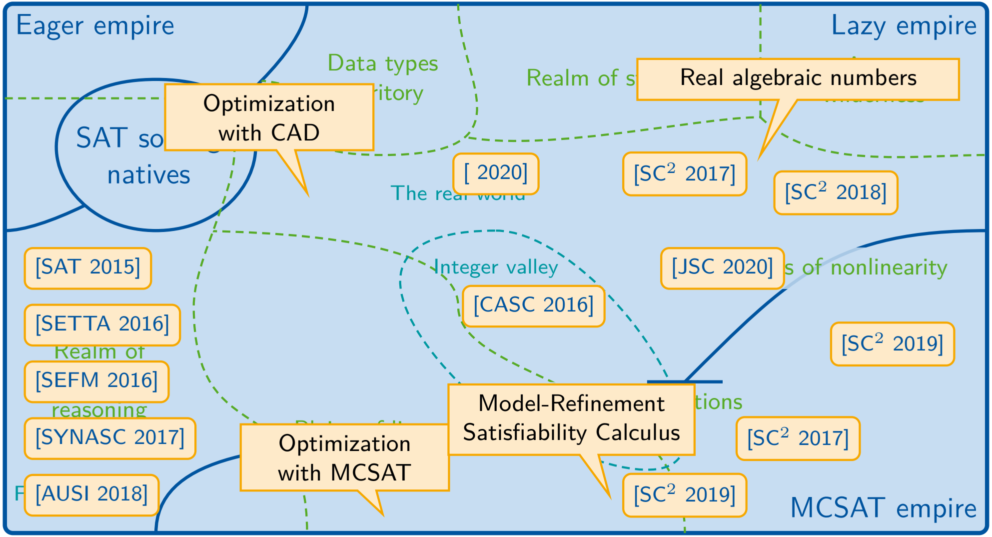
Summary


Eager empire
Lazy empire
Data types
Realm of strings
Array wilderness
Optimization with CAD
SAT so
natives
[2020]
[SC² 2017]
[SC² 2018]
The real world
[SAT 2015]
[SETTA 2016]
Realm of
[SEFM 2016]
reasoning
[SYNASC 2017]
[AUSI 2018]
Integer valley
[JSC 2020]
s of nonlinearity
[CASC 2016]
[SC² 2019]
United arithmetic factions
Optimization with MCSAT
[SC² 2017]
[SC² 2019]
MCSAT empire

Summary


Eager empire
Lazy empire
Data types
ritory
Realm of strings
Array
wilderness
SAT so
natives
Optimization
with CAD
[2020]
[SC^2 2017]
[SC^2 2018]
The rea
[SAT 2015]
Integer valley
[JSC 2020]
s of nonlinearity
[CASC 2016]
[SETTA 2016]
Realm of
[SEFM 2016]
reasoning
[SC^2 2019]
tions
Model-Refinement
Satisfiability Calculus
[SYNASC 2017]
Optimization
with MCSAT
[SC^2 2017]
[AUSI 2018]
[SC^2 2019]
MCSAT empire

Summary


Eager empire
Lazy empire
MCSAT empire
Data types
Realm of s
Real algebraic numbers
Optimization with CAD
SAT so
natives
[2020]
[SC² 2017]
[SC² 2018]
The real world
[SAT 2015]
[SETTA 2016]
Realm of
[SEFM 2016]
reasoning
[SYNASC 2017]
[AUSI 2018]
Integer valley
[CASC 2016]
[JSC 2020]
s of nonlinearity
[SC² 2019]
Model-Refinement Satisfiability Calculus
tions
[SC² 2017]
Optimization with MCSAT
[SC² 2019]

Conclusion

- CAD is suitable for lazy SMT solving
  - Incrementality is very beneficial
  - Backtracking is efficient
  - Infeasible subsets can be computed

Conclusion

- CAD is suitable for lazy SMT solving
  - Incrementality is very beneficial
  - Backtracking is efficient
  - Infeasible subsets can be computed
- `MCSAT` is an effective alternative
- `MCSAT` benefits from multiple explanations and variable orderings

Conclusion

- CAD is suitable for lazy SMT solving
  - Incrementality is very beneficial
  - Backtracking is efficient
  - Infeasible subsets can be computed
- MCSAT is an effective alternative
- MCSAT benefits from multiple explanations and variable orderings

Directions for future research:

- Optimize CAD integration: better scheduling, NuCAD, parallelization, ...
- Novel adaptations of CAD: CAC [2020], NuCAD [Brown 2015], ...
- New methods for MCSAT: explanations, orderings, assignment finders, ...
- More applications: optimization, dedicated heuristics, ...

References I

- Erika Ábrahám, Florian Corzilius, Einar Broch Johnsen, Gereon Kremer, and Jacopo Mauro. "Zephyrus2: On the Fly Deployment Optimization Using SMT and CP Technologies". In: Dependable Software Engineering: Theories, Tools, and Applications (SETTA 2016).
- Erika Ábrahám, James H. Davenport, Matthew England, and Gereon Kremer. "Deciding the Consistency of Non-Linear Real Arithmetic Constraints with a Conflict Driven Search Using Cylindrical Algebraic Coverings". In: arXiv e-prints (2020). Accepted at Journal of Logical and Algebraic Methods in Programming. arXiv: 2003.05633.
- Erika Ábrahám and Gereon Kremer. "Satisfiability Checking: Theory and Applications". In: Software Engineering and Formal Methods (SEFM 2016).
- Erika Ábrahám and Gereon Kremer. "SMT Solving for Arithmetic Theories: Theory and Tool Support". In: Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2017).
- Erika Ábrahám, Jasper Nalbach, and Gereon Kremer. "Embedding the Virtual Substitution Method in the Model Constructing Satisfiability Calculus Framework". In: Satisfiability Checking and Symbolic Computation (SC^2 2017) at ISSAC.
- Lorena Calvo Bartolomé. "Using Fourier-Motzkin Variable Elimination for MCSAT Explanations in SMT-RAT". Bachelor's thesis. RWTH Aachen University, 2018.
- François Bobot, Stéphane Graham-Lengrand, Bruno Marre, and Guillaume Bury. "Centralizing equality reasoning in MCSAT". In: Satisfiability Modulo Theories (SMT 2018) at FLoC.
- Christopher W. Brown. "QEPCAD B: A program for computing with semi-algebraic sets using CADs". In: ACM SIGSAM Bulletin 37 (4 2003).

References II

- Christopher W. Brown. "Constructing a Single Open Cell in a Cylindrical Algebraic Decomposition". In: International Symposium on Symbolic and Algebraic Computation (ISSAC 2013).
- Christopher W. Brown. "Open Non-uniform Cylindrical Algebraic Decompositions". In: International Symposium on Symbolic and Algebraic Computation (ISSAC 2015).
- Christopher W. Brown and Marek Košta. "Constructing a single cell in cylindrical algebraic decomposition". In: Journal of Symbolic Computation 70 (2015).
- Vasek Chvátal. "A Greedy Heuristic for the Set-Covering Problem". In: Mathematics of Operations Research 4.3 (1979).
- George E. Collins. "Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition". In: Automata Theory and Formal Languages (2nd GI Conference 1975).
- George E. Collins and Hoon Hong. "Partial Cylindrical Algebraic Decomposition for Quantifier Elimination". In: Journal of Symbolic Computation 12 (3 1991).
- Florian Corzilius, Gereon Kremer, Sebastian Junges, Stefan Schupp, and Erika Ábrahám. "SMT-RAT: An Open Source C++ Toolbox for Strategic and Parallel SMT Solving". In: Theory and Applications of Satisfiability Testing (SAT 2015).
- Florian Corzilius, Ulrich Loup, Sebastian Junges, and Erika Ábrahám. "SMT-RAT: An SMT-Compliant Nonlinear Real Arithmetic Toolbox (Tool Presentation)". In: Theory and Applications of Satisfiability Testing (SAT 2012).

References III

- Matthew England, Russell Bradford, James H. Davenport, and David Wilson. "Choosing a Variable Ordering for Truth-Table Invariant Cylindrical Algebraic Decomposition by Incremental Triangular Decomposition". In: International Congress on Mathematical Software (ICMS 2014).
- Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, and Harald Zankl. "SAT Solving for Termination Analysis with Polynomial Interpretations". In: Theory and Applications of Satisfiability Testing (SAT 2007).
- Stéphane Graham-Lengrand and Dejan Jovanović. "An MCSAT treatment of Bit-Vectors (preliminary report)". In: Satisfiability Modulo Theories (SMT 2017) at CAV.
- Rebecca Haehn, Gereon Kremer, and Erika Ábrahám. "Evaluation of Equational Constraints for CAD in SMT Solving". In: Satisfiability Checking and Symbolic Computation (SC^2 2018) at FLoC.
- Wanja Hentze. "Computing minimal infeasible subsets for the Cylindrical Algebraic Decomposition". Bachelor's thesis. RWTH Aachen University, 2017.
- Zongyan Huang, Matthew England, David Wilson, James H. Davenport, Lawrence C. Paulson, and James Bridge. "Applying Machine Learning to the Problem of Choosing a Heuristic to Select the Variable Ordering for Cylindrical Algebraic Decomposition". In: Intelligent Computer Mathematics (CICM 2014).
- Maximilian Jaroschek, Pablo Federico Dobal, and Pascal Fontaine. "Adapting Real Quantifier Elimination Methods for Conflict Set Computation". In: Frontiers of Combining Systems (FroCoS 2015).
- Dejan Jovanović. "Solving Nonlinear Integer Arithmetic with MCSAT". In: Verification, Model Checking, and Abstract Interpretation (VMCAI 2017).

References IV

- Dejan Jovanović, Clark Barrett, and Leonardo de Moura. "The Design and Implementation of the Model Constructing Satisfiability Calculus". In: Formal Methods in Computer-Aided Design (FMCAD 2013).
- Dejan Jovanović and Leonardo de Moura. "Solving Non-linear Arithmetic". In: Automated Reasoning (IJCAR 2012).
- Gereon Kremer and Erika Ábrahám. "Modular strategic SMT solving with SMT-RAT". In: Acta Universitatis Sapientiae, Informatica 10 (1 2018).
- Gereon Kremer and Erika Ábrahám. "Fully Incremental Cylindrical Algebraic Decomposition". In: Journal of Symbolic Computation 100 (2020).
- Gereon Kremer, Erika Ábrahám, and Vijay Ganesh. "On the Proof Complexity of MCSAT". In: Satisfiability Checking and Symbolic Computation (SC^2 2019) at SIAM AG.
- Gereon Kremer, Florian Corzilius, and Erika Ábrahám. "A Generalised Branch-and-Bound Approach and Its Application in SAT Modulo Nonlinear Integer Arithmetic". In: Computer Algebra in Scientific Computing (CASC 2016).
- Ulrich Loup and Erika Ábrahám. "GiNaCRA: A C++ Library for Real Algebraic Computations". In: NASA Formal Methods (NFM 2011).
- Ulrich Loup, Karsten Scheibler, Florian Corzilius, Erika Ábrahám, and Bernd Becker. "A Symbiosis of Interval Constraint Propagation and Cylindrical Algebraic Decomposition". In: Automated Deduction (CADE-24 2013).

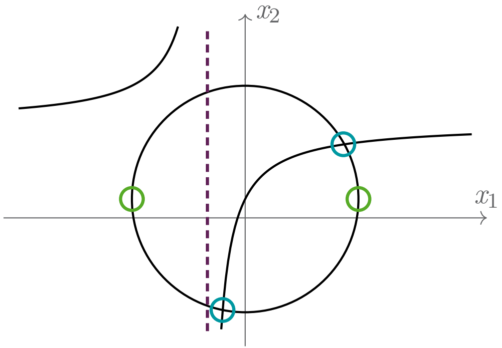
References V

- Scott McCallum. "An Improved Projection Operation for Cylindrical Algebraic Decomposition". In: European Conference on Computer Algebra (EUROCAL 1985).
- Scott McCallum. "On Projection in CAD-based Quantifier Elimination with Equational Constraint". In: International Symposium on Symbolic and Algebraic Computation (ISSAC 1999).
- Leonardo de Moura and Dejan Jovanović. "A Model-Constructing Satisfiability Calculus". In: Verification, Model Checking, and Abstract Interpretation (VMCAI 2013).
- Jasper Nalbach, Gereon Kremer, and Erika Ábrahám. "On Variable Orderings in MCSAT for Non-linear Real Arithmetic (extended abstract)". In: Satisfiability Checking and Symbolic Computation (SC^2 2019) at SIAM AG.
- Tom Neuhäuser. "Quantifier Elimination by Cylindrical Algebraic Decomposition". Bachelor's thesis. RWTH Aachen University, 2018.
- Malte Neuß. "Using Single CAD Cells as Explanations in MCSAT-style SMT Solving". Master's thesis. RWTH Aachen University, 2018.
- Tarik Viehmann, Gereon Kremer, and Erika Ábrahám. "Comparing Different Projection Operators in the Cylindrical Algebraic Decomposition for SMT Solving". In: Satisfiability Checking and Symbolic Computation (SC^2 2017) at ISSAC.
- Aleksandar Zeljić, Christoph M. Wintersteiger, and Philipp Rümmer. "Deciding Bit-Vector Formulas with mcSAT". In: Theory and Applications of Satisfiability Testing (SAT 2016).

CAD projection

Example: McCallum

$$Proj(P) = \{\mathrm{coeffs}(p), \mathrm{disc}(p) \mid p \in P\} \cup \{\mathrm{res}(p, q) \mid p, q \in P\}$$



[McCallum 1985]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \ldots\}$ is infeasible

[Jaroschek + 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \dots$

[Jaroschek + 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \dots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)

[Jaroschek + 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

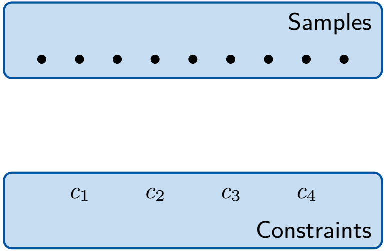
- Set of constraints $\{c_1, c_2, \ldots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \ldots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)
- How to construct?

[Jaroschek + 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \dots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)
- How to construct?

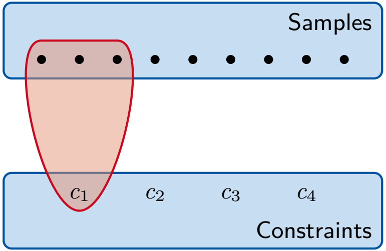


[Jaroschek + 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \dots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics…)
- How to construct?

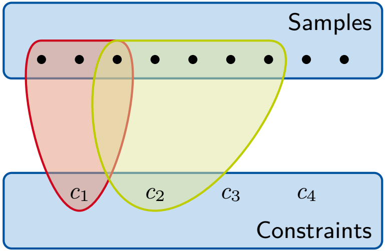


[Jaroschek + 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \dots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)
- How to construct?

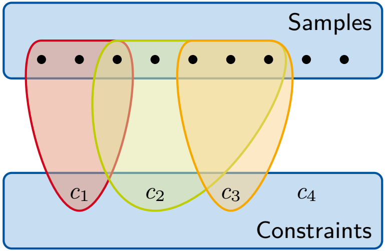


[Jaroschek+ 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \lor \neg c_2 \lor \dots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)
- How to construct?

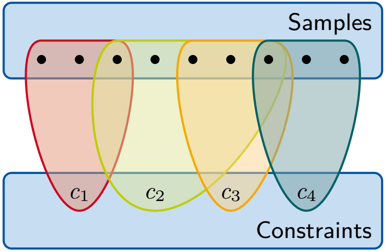


[Jaroschek + 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \dots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)
- How to construct?

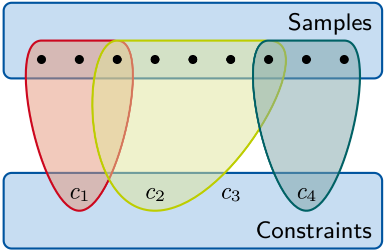


[Jaroschek+ 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \dots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)
- How to construct?

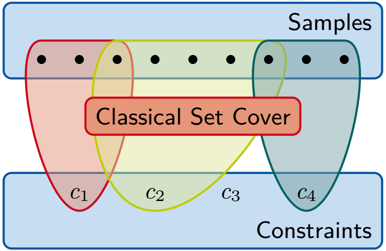


[Jaroschek + 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \dots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)
- How to construct?

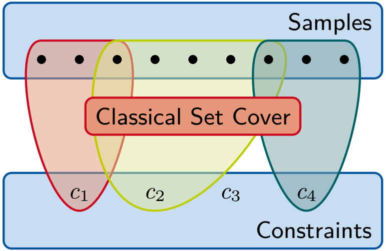


[Jaroschek + 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \dots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \dots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)
- How to construct?
- Significant improvements for MIS ...

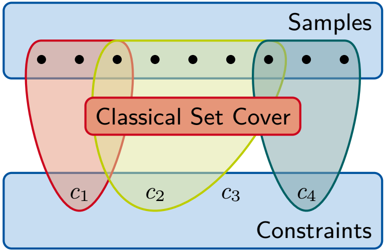


[Jaroschek+ 2015] [Chvátal 1979]

Minimal Infeasible Subsets

[Hentze 2017]

- Set of constraints $\{c_1, c_2, \ldots\}$ is infeasible
- SAT solver learns: $\neg c_1 \vee \neg c_2 \vee \ldots$
- Infeasible subset of these constraints
  - Smaller conflict clause
  - Excludes larger Boolean space
- Huge improvements (in other logics...)
- How to construct?
- Significant improvements for MIS ...
- but no effect on solver performance



Solver	SAT	UNSAT	overall
`Greedy-PP`	4327	4249	8576
`Greedy-Weighted`	4329	4247	8576
`Hybrid`	4328	4248	8576
`Trivial`	4325	4251	8576
`Exact`	4328	4249	8577
`Greedy`	4328	4250	8578

[Jaroschek+ 2015] [Chvátal 1979]

Using CAD for ...

...Quantifier Elimination [Neuhäuser 2018]

$$(\exists x.\ \forall y.\ \varphi(x, y, z)) \Leftrightarrow \varphi'(z)$$

- Validation of our CAD implementation
- Show reuse of implementation
- Results comparable to `QEPCAD B`

[Brown 2003] [Fuhs + 2007]

Using CAD for ...

...Quantifier Elimination [Neuhäuser 2018]

$$(\exists x.\ \forall y.\ \varphi(x, y, z)) \Leftrightarrow \varphi'(z)$$

- Validation of our CAD implementation
- Show reuse of implementation
- Results comparable to `QEPCAD B`

...Integer problems [CASC 2016]

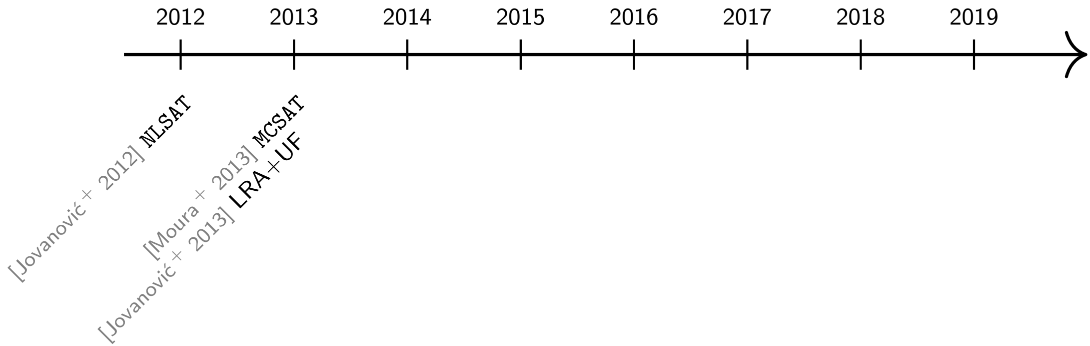
$$\exists x.\ \varphi(x) \qquad x \in \mathbb{Z}$$

- Common approach: bit-blasting
  mostly aims for small domains
- Our approach: Branch & Bound
  mostly aims for algebraically easy problems & UNSAT

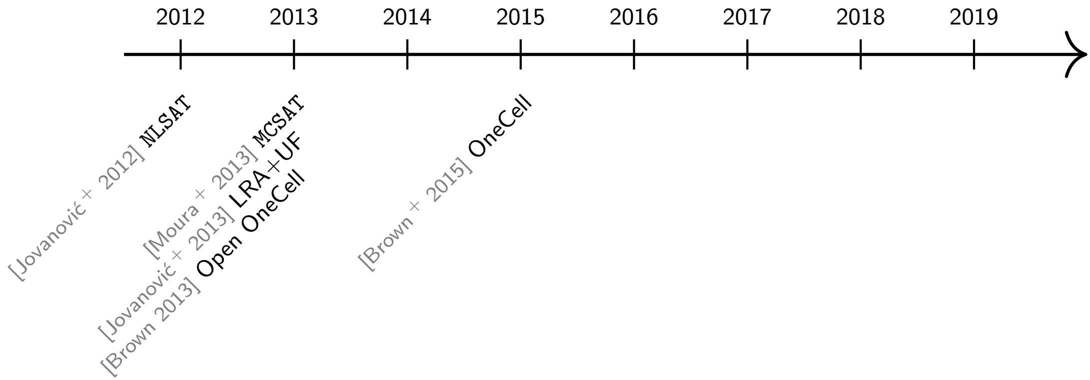
Solver	SAT	UNSAT	overall
`NIA-B&B`	1044	518	1562
`NIA-Blast`	4367	26	4393
`NIA-Full`	4490	508	4998

[Brown 2003] [Fuhs + 2007]

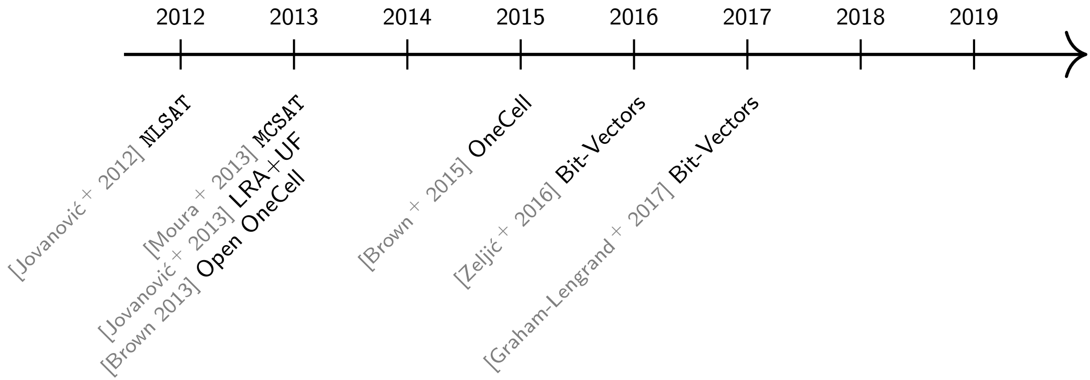
MCSAT history



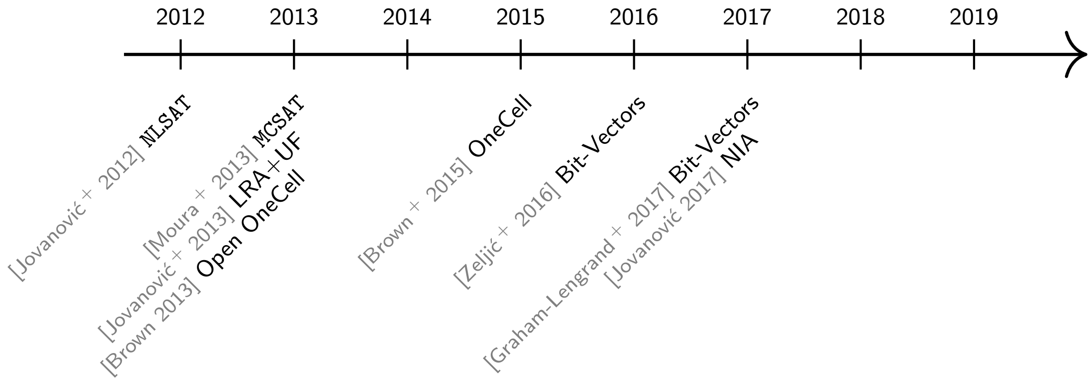
MCSAT history



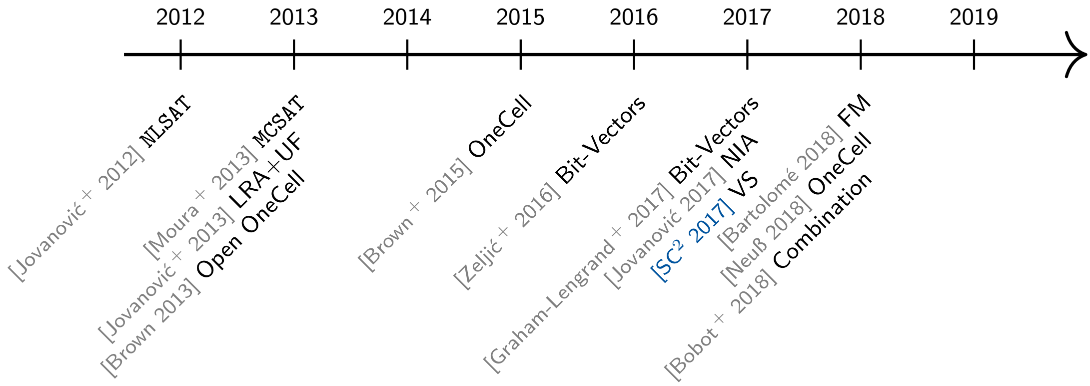
MCSAT history



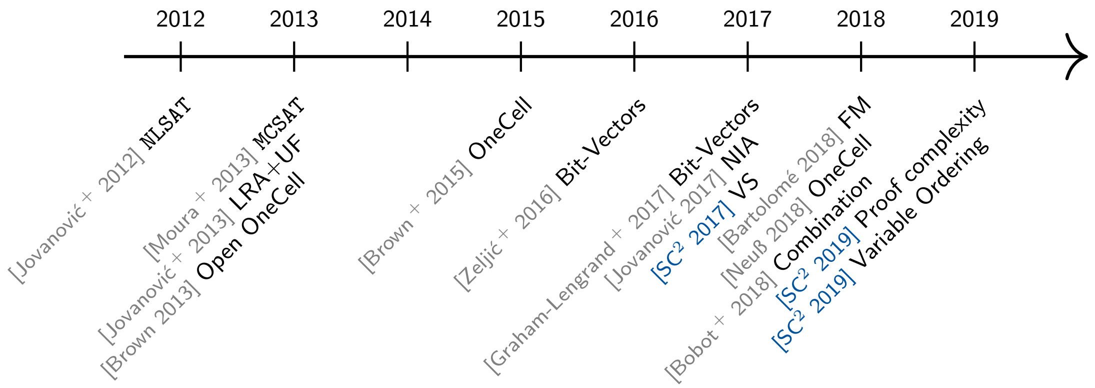
MCSAT history



MCSAT history



MCSAT history



Theory decisions

Regular assignment finder

- Collect univariate constraints
- Compute sign-invariant regions
- Sample from satisfying region
- Essentially a one-dimensional CAD
- Considers only next theory variable

Theory decisions

Regular assignment finder

- Collect univariate constraints
- Compute sign-invariant regions
- Sample from satisfying region
- Essentially a one-dimensional CAD
- Considers only next theory variable

SMT-based assignment finder

- Collect all constraints
- Calls a full SMT solver
- Uses model as new theory assignment
- Considers multiple theory variables at once
- Larger look ahead

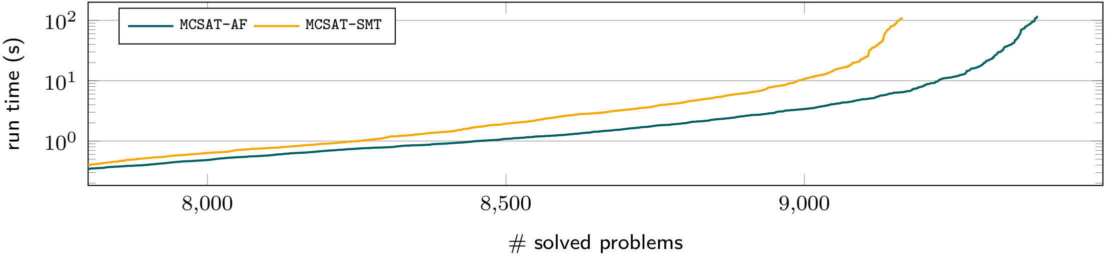
Theory decisions

Regular assignment finder

- Collect univariate constraints
- Compute sign-invariant regions
- Sample from satisfying region
- Essentially a one-dimensional CAD
- Considers only next theory variable

SMT-based assignment finder

- Collect all constraints
- Calls a full SMT solver
- Uses model as new theory assignment
- Considers multiple theory variables at once
- Larger look ahead



Comparison of CDCL*(T) and MCSAT [SC[2] 2019]

- Two similar proof systems: CDCL*(T) and MCSAT
- Experience: MCSAT better for nonlinear
- Goal: theoretic comparison of CDCL*(T) and MCSAT

Comparison of CDCL*(T) and MCSAT

[SC² 2019]

- Two similar proof systems: CDCL*(T) and MCSAT
- Experience: MCSAT better for nonlinear
- Goal: theoretic comparison of CDCL*(T) and MCSAT
- Proof complexity: asymptotic length of shortest proof
- Proof complexity is the same*
- The proof can be completely different

Comparison of CDCL*(T) and MCSAT

[SC² 2019]

- Two similar proof systems: CDCL*(T) and MCSAT
- Experience: MCSAT better for nonlinear
- Goal: theoretic comparison of CDCL*(T) and MCSAT
- Proof complexity: asymptotic length of shortest proof
- Proof complexity is the same*
- The proof can be completely different
- Algorithmic equivalence: polynomial simulation of another method
- CDCL*(T) and MCSAT simulate each other*
- The proofs are logically equivalent

*: terms and conditions apply