

Topics in Model-Based Reasoning

Towards Integration of Proving and Solving

Maria Paola Bonacina

Dipartimento di Informatica
Università degli Studi di Verona
Verona, Italy

March, 2014

Outline

Introduction: Towards model-based reasoning

I part: A classic from the literature: DPLL-CDCL

II part: Solver + prover in DPLL($\Gamma + \mathcal{T}$)

III part: Discussion of current trends in the field

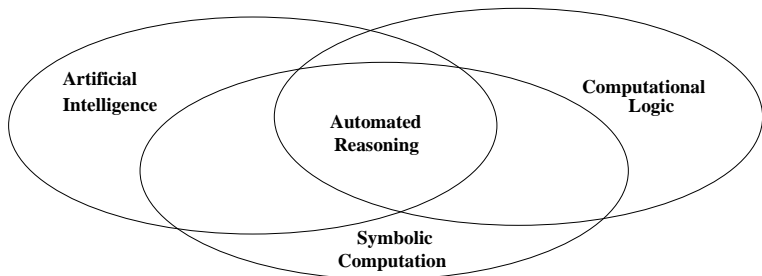
Introduction: Towards model-based reasoning

I part: A classic from the literature: DPLL-CDCL

II part: Solver + prover in DPLL($\Gamma + \mathcal{T}$)

III part: Discussion of current trends in the field

Automated reasoning



- ▶ In AI we work with symbols: automated reasoning is symbolic reasoning
- ▶ Symbolic reasoning: Logico-deductive, Probabilistic ...

The gist of this lecture I

- ▶ Automated reasoning from **proofs** to **models**
- ▶ Models are relevant to applications, e.g.:
 - ▶ Program verification: a program state is a model
 - ▶ Program testing: model as “mole” in automated test generation
 - ▶ Program synthesis: model as example in example-driven synthesis

The gist of this lecture II

- ▶ Proofs mean **proving**
- ▶ Models mean **solving**
- ▶ Towards **integrations** of proving and solving

Symbolic reasoning: Proving

- ▶ Validity: $\mathcal{T} \models \varphi$
- ▶ Refutationally: $\mathcal{T} \cup \{\neg\varphi\}$ unsatisfiable
- ▶ **Inference:** $\mathcal{T} \cup \{\neg\varphi\} \vdash \perp$ (Success!)
- ▶ If not: \mathcal{T} -model of $\neg\varphi$, counter-example for φ

Symbolic reasoning: Solving

- ▶ Satisfiability: is there a \mathcal{T} -model of φ ?
- ▶ Search: solution found (Success!)
- ▶ If not: $\mathcal{T} \cup \{\varphi\}$ unsatisfiable, $\mathcal{T} \models \neg\varphi$

Theorem proving strategies (Semi-decision procedures)

- ▶ First-order logic with equality
- ▶ Unsatisfiability is semi-decidable, satisfiability is not
- ▶ Search for **proof** (refutation)
- ▶ Models for **semantic guidance**, e.g.:
 - ▶ Hyper-resolution
 - ▶ Set of support
 - ▶ Semantic resolution

Algorithmic reasoning (Decision procedures)

- ▶ Satisfiability decidable: **Symmetry restored**
- ▶ Propositional logic
- ▶ Decidable (fragments of) first-order theories, e.g.:
 - ▶ QFF: equality, recursive data structures (e.g., lists), arrays
 - ▶ Linear arithmetic (integers, rationals), arithmetic (algebraic reals)

Towards integration I

- ▶ Integrating solvers (e.g., arithmetic) into first-order reasoners

Towards integration II

Integration in the reasoner's operations:

- ▶ **Deduction** guides **search for model**
- ▶ Candidate **partial model** guides **deduction**
- ▶ How?

I part: A classic from the literature: DPLL-CDCL

- ▶ DPLL: The Davis-Putnam-Logemann-Loveland procedure with
- ▶ Conflict driven clause learning (DPLL-CDCL), or
- ▶ How the integration of search and inference in propositional logic brought Boolean satisfiability from theoretical hardness to practical success

Propositional logic (SAT)

- ▶ Davis-Putnam-Logemann-Loveland (DPLL) procedure
- ▶ Decision procedure:
 - model found: return **sat**;
 - failure: return **unsat**
- ▶ Backtracking search for model

DPLL

- ▶ Build candidate model M
- ▶ State of derivation: $M \parallel F$
 - M : sequence of truth assignments
 - F : clauses to satisfy
- ▶ Depth-first search with backtracking

DPLL-CDCL I

State of derivation: $M \parallel F$

- ▶ **Decide**: guess L is true, add it to M (decided literal)
- ▶ **UnitPropagate**: propagate consequences of assignment (implied literals)
- ▶ **Conflict**: detect $L_1 \vee \dots \vee L_n$ all false
- ▶ **Unsat**: conflict clause is \square (nothing else to try)
- ▶ **Sat**: all variables assigned

DPLL-CDCL II

State of derivation: $M \parallel F$

- ▶ **Explain**: unfold implied literals in conflict clause by resolution
- ▶ **Learn** conflict clause $C \vee L$
- ▶ **Backjump**: when only L assigned at current decision level, jump back to least recent level where C false and L unassigned, undo at least one decision, make L true (implied by $C \vee L$)

Conflict-Driven Clause Learning (CDCL)

- ▶ **Conflict**: M falsifies clause $L_1 \vee \dots \vee L_n$: conflict clause
- ▶ **Explain**: resolve and get another conflict clause
$$L_1 \vee \dots \vee L_n$$
$$\neg L_1 \vee Q_2 \dots \vee Q_k$$
- ▶ **Learn**: may add resolvent(s)
- ▶ **Backjump**: undoes at least an assignment, jumps back as far as possible to state where learnt resolvent can be satisfied

Example of CDCL

$$F = \{\neg a \vee b, \neg c \vee d, \neg e \vee \neg f, f \vee \neg e \vee \neg b\}$$

$$M = a \ b \ c \ d \ e \ \neg f$$

blue: assignments; violet: propagations

Conflict: $f \vee \neg e \vee \neg b$

Explain by resolving $f \vee \neg e \vee \neg b$ and $\neg e \vee \neg f$: $\neg e \vee \neg b$

Learn $\neg e \vee \neg b$: no model with e and b true

Jump back to earliest state with $\neg b$ false and $\neg e$ unassigned:

$$M = a \ b \ \neg e$$

Chronological backtracking: $M = a \ b \ c \ d \ \neg e$

Decision procedures

- ▶ Davis-Putnam-Logemann-Loveland (DPLL) procedure for SAT
- ▶ \mathcal{T} -solver: Decision procedure for \mathcal{T}
Equality: congruence closure (CC)
- ▶ DPLL(\mathcal{T})-based SMT-solver: Decision procedure for $\mathcal{T} = \bigcup_{i=1}^n \mathcal{T}_i$ with
- ▶ Combination of \mathcal{T}_i -sat procedures by a method called **equality sharing**

Satisfiability modulo theories (SMT)

- ▶ DPLL(\mathcal{T}) procedure
- ▶ Integrate \mathcal{T} -satisfiability procedure in DPLL
- ▶ Ground first-order literals abstracted to propositional variables
- ▶ CDCL: same

DPLL(\mathcal{T})

State of derivation: $M \parallel F$

- ▶ \mathcal{T} -Propagate: add to M an L that is \mathcal{T} -consequence of M
- ▶ \mathcal{T} -Conflict: detect that L_1, \dots, L_n in M are \mathcal{T} -inconsistent

Theory combination by equality sharing I

- ▶ Theories $\mathcal{T}_1, \dots, \mathcal{T}_n$
- ▶ $\mathcal{T} = \bigcup_{i=1}^n \mathcal{T}_i$
- ▶ \mathcal{T}_i -satisfiability procedures
- ▶ Disjoint: share only \simeq and uninterpreted constants
- ▶ Mixed terms **separated** by introducing new constants
- ▶ Need to agree on:
 - ▶ Shared constants
 - ▶ Cardinalities of shared sorts

Theory combination by equality sharing II

- ▶ Compute **arrangement**: which shared constants are equal and which are not
- ▶ \mathcal{T}_i -solvers generate and propagate all entailed (disjunctions of) equalities between shared constants
- ▶ For cardinalities: assume **stably infinite**: every \mathcal{T}_i -sat ground formula has \mathcal{T}_i -model with infinite cardinality

Model-based theory combination (MBTC)

- ▶ Assume \mathcal{T}_i -satisfiability procedure that builds a \mathcal{T}_i -model (e.g., linear arithmetic)
- ▶ Optimistic approach: propagate equalities **true in \mathcal{T}_i -model**
- ▶ If not entailed: conflict + backjumping with CDCL + update \mathcal{T}_i -model
- ▶ Rationale: few equalities matter in practice

II part: Solver + prover in DPLL($\Gamma + \mathcal{T}$)

- ▶ Γ : first-order inference system
- ▶ DPLL(\mathcal{T}): SMT-solver with DPLL-CDCL and equality sharing
- ▶ A tight integration: the DPLL($\Gamma + \mathcal{T}$) method

Motivation

- ▶ Decision procedures are most desirable, but ...
- ▶ Formulæ from SW verification tools (verifying compiler, static analyzer, test generator, synthesizer, model checker) use **quantifiers** to write
 - ▶ invariants
 - ▶ axioms of theories without decision procedure
- ▶ Need for **generic first-order inferences**

Shape of problem

- ▶ Background theory \mathcal{T}
 - ▶ $\mathcal{T} = \bigcup_{i=1}^n \mathcal{T}_i$ (linear arithmetic, data structures)
- ▶ Set of formulæ: $\mathcal{R} \cup P$
 - ▶ \mathcal{R} : set of **non-ground** clauses **without** \mathcal{T} -symbols
 - ▶ P : large **ground** formula (set of ground clauses)
typically **with** \mathcal{T} -symbols
- ▶ Determine whether $\mathcal{R} \cup P$ is satisfiable modulo \mathcal{T}

Superposition-based inference system Γ

- ▶ FOL $_{+=}$ clauses with universally quantified variables
- ▶ Axiomatized theories
- ▶ Deduce clauses from clauses (**expansion**)
- ▶ Remove redundant clauses (**contraction**)
- ▶ **Well-founded** ordering \succ on terms and literals to restrict expansion and define contraction
- ▶ Semi-decision procedure
- ▶ No backtracking

Inference system Γ

State of derivation: set of clauses F

- ▶ **Resolution**
- ▶ **Superposition/Paramodulation**: resolution with equality built-in
- ▶ **Simplification**: by well-founded rewriting
- ▶ **Subsumption**: eliminate less general clauses
- ▶ Other rules: e.g., **Factoring** rules, **Deletion** of trivial clauses

Theorem-proving strategy as decision procedure

- ▶ **Termination** results by analysis of inferences: Γ is \mathcal{T} -satisfiability procedure
- ▶ Covered theories include: **lists**, **arrays** and **records** with or without extensionality, **recursive data structures**

Also for combination of theories

- ▶ If Γ terminates on \mathcal{R}_i -sat problems, it terminates also on \mathcal{R} -sat problems for $\mathcal{R} = \bigcup_{i=1}^n \mathcal{R}_i$, if the \mathcal{R}_i 's are **disjoint** and **variable-inactive**
- ▶ Variable-inactivity: no maximal literals of the form $t \simeq x$ where $x \notin \text{Var}(t)$ (no paramodulation from variables)
- ▶ The only inferences across theories are **paramodulations from shared constants** (correspond to equalities between shared constants in equality sharing)

Variable inactivity implies stable infiniteness

- ▶ If \mathcal{R} is variable-inactive, then it is stably infinite
- ▶ Γ reveals lack of stable infiniteness by generating a **cardinality constraint** (e.g., $y \simeq x \vee y \simeq z$) which is not variable-inactive

Recap on first-order inference systems

- ▶ Resolution/superposition-based engines good for reasoning on formulæ with quantified variables: **automated** instantiation
- ▶ Not for large non-Horn clauses
- ▶ Not for theories such as linear arithmetic or bit-vectors
- ▶ Unexpected: they are satisfiability-procedures for theories such as lists, arrays, records and their combinations

DPLL($\Gamma+\mathcal{T}$): integrate Γ in DPLL(\mathcal{T}) I

- ▶ **Model-based deduction:**
literals in M can be premises of Γ -inferences
- ▶ Stored as **hypotheses** in inferred clause
- ▶ **Hypothetical clause:** $(L_1 \wedge \dots \wedge L_n) \triangleright (L'_1 \vee \dots \vee L'_m)$
interpreted as $\neg L_1 \vee \dots \vee \neg L_n \vee L'_1 \vee \dots \vee L'_m$

DPLL($\Gamma+\mathcal{T}$): integrate Γ in DPLL(\mathcal{T}) II

- ▶ Inferred clauses **inherit** hypotheses from premises
- ▶ **Backjump**: remove hypothetical clauses depending on undone assignments

DPLL($\Gamma+\mathcal{T}$): expansion inferences

- ▶ If non-ground clauses C_1, \dots, C_m and ground \mathcal{R} -literals L_{m+1}, \dots, L_n generate C :
 $H_1 \triangleright C_1, \dots, H_m \triangleright C_m$ and L_{m+1}, \dots, L_n in M generate $H_1 \cup \dots \cup H_m \cup \{L_{m+1}, \dots, L_n\} \triangleright C$
- ▶ Only \mathcal{R} -literals: Γ -inferences ignore \mathcal{T} -literals
- ▶ Take ground unit \mathcal{R} -clauses from M as MBTC puts them there

DPLL($\Gamma+\mathcal{T}$): contraction inferences

- ▶ Don't delete clause if clauses that make it redundant gone by backjumping
 - ▶ Level of a literal in M : its decision level
 - ▶ Level of a set of literals: the maximum

DPLL($\Gamma+\mathcal{T}$): contraction inferences

- ▶ If non-ground clauses C_1, \dots, C_m and ground \mathcal{R} -literals L_{m+1}, \dots, L_n simplify C to C' :
 $H_1 \triangleright C_1, \dots, H_m \triangleright C_m$ and L_{m+1}, \dots, L_n in M simplify $H \triangleright C$
 to $H \cup H_1 \cup \dots \cup H_m \cup \{L_{m+1}, \dots, L_n\} \triangleright C'$
 - ▶ If $level(H) \geq level(H')$: delete
 - ▶ If $level(H) < level(H')$: disable
 (re-enable when backjumping $level(H')$)

Completeness of DPLL($\Gamma + \mathcal{T}$)

- ▶ **Refutational completeness** of the inference system:
 - ▶ From that of Γ , DPLL(\mathcal{T}) and equality sharing
 - ▶ Combines both built-in and axiomatized theories
- ▶ **Fairness** of the search plan:
 - ▶ Depth-first search fair only for ground SMT problems;
 - ▶ Add **iterative deepening** on inference depth:
 k -bounded DPLL($\Gamma + \mathcal{T}$)

DPLL($\Gamma+\mathcal{T}$): Summary

Use each engine for what is best at:

- ▶ DPLL(\mathcal{T}) works on ground clauses and built-in theory
- ▶ Γ works on non-ground clauses and ground unit clauses taken from M : Γ works on \mathcal{R} -satisfiability problem
- ▶ Γ -inferences **guided by current partial model**

Can DPLL($\Gamma + \mathcal{T}$) still be a decision procedure?

Problematic axioms do occur in relevant inputs:

1. $\neg(x \sqsubseteq y) \vee f(x) \sqsubseteq f(y)$ (Monotonicity)
2. $a \sqsubseteq b$ generates by resolution
3. $\{f^i(a) \sqsubseteq f^i(b)\}_{i \geq 0}$

When $f(a) \sqsubseteq f(b)$ or $f^2(a) \sqsubseteq f^2(b)$ often suffice to show satisfiability

Idea: Allow speculative inferences

1. $\neg(x \sqsubseteq y) \vee f(x) \sqsubseteq f(y)$

2. $a \sqsubseteq b$

3. $a \sqsubseteq f(c)$

4. $\neg(a \sqsubseteq c)$

1. Add $f(x) \simeq x$

2. Rewrite $a \sqsubseteq f(c)$ into $a \sqsubseteq c$ and get \square : backtrack!

3. Add $f(f(x)) \simeq x$

4. $a \sqsubseteq b$ yields only $f(a) \sqsubseteq f(b)$

5. $a \sqsubseteq f(c)$ yields only $f(a) \sqsubseteq c$

6. Terminate and detect satisfiability

Speculative inferences in DPLL($\Gamma+\mathcal{T}$)

- ▶ Speculative inference: add **arbitrary** clause C
- ▶ To induce termination on satisfiable input
- ▶ What if it makes problem unsatisfiable?!
- ▶ Detect conflict and backjump:
 - ▶ $\lceil C \rceil$: new propositional variable (a “name” for C)
 - ▶ Add $\lceil C \rceil \triangleright C$ to clauses and $\lceil C \rceil$ to M
 - ▶ Speculative inferences are **reversible**

Example as done by system

1. $\neg(x \sqsubseteq y) \vee f(x) \sqsubseteq f(y)$
2. $a \sqsubseteq b$
3. $a \sqsubseteq f(c)$
4. $\neg(a \sqsubseteq c)$

1. Add $\lceil f(x) \simeq x \rceil \triangleright f(x) \simeq x$
2. Rewrite $a \sqsubseteq f(c)$ into $\lceil f(x) \simeq x \rceil \triangleright a \sqsubseteq c$
3. Generate $\lceil f(x) \simeq x \rceil \triangleright \square$; Backtrack, learn $\neg\lceil f(x) \simeq x \rceil$
4. Add $\lceil f(f(x)) \simeq x \rceil \triangleright f(f(x)) \simeq x$
5. $a \sqsubseteq b$ yields only $f(a) \sqsubseteq f(b)$
6. $a \sqsubseteq f(c)$ yields only $\lceil f(f(x)) = x \rceil \triangleright f(a) \sqsubseteq c$
7. Terminate and detect satisfiability

Decision procedures with speculative inferences

To decide satisfiability modulo \mathcal{T} of $\mathcal{R} \cup P$:

- ▶ Find sequence of **speculative axioms** U
- ▶ Show that there exists k s.t. k -bounded DPLL($\Gamma+\mathcal{T}$) is guaranteed to terminate
 - ▶ returning Unsat if $\mathcal{R} \cup P$ is \mathcal{T} -unsatisfiable
 - ▶ in a state which is not stuck at k otherwise

Decision procedures

- ▶ \mathcal{R} has single monadic function symbol f
- ▶ **Essentially finite**: if $\mathcal{R} \cup P$ is satisfiable, has model where range of f is **finite**
- ▶ Such a model satisfies $f^j(x) \simeq f^k(x)$ for some $j \neq k$
- ▶ Add **pseudo-axioms** $f^j(x) \simeq f^k(x)$, $j > k$
- ▶ Use $f^j(x) \simeq f^k(x)$ as rewrite rule to **limit term depth**
- ▶ **Clause length limited** by properties of Γ and \mathcal{R}
- ▶ Only finitely many clauses generated: termination

Situations where clause length is limited

Γ : Superposition, Resolution + negative selection, Simplification

Negative selection: only positive literals in positive clauses resolve or superpose

- ▶ \mathcal{R} is Horn: number of literals in each clause is bounded
- ▶ \mathcal{R} is **ground-preserving**: all variables appear also in negative literals
the only positive clauses are ground
only finitely many clauses generated

Axiomatizations of type systems

$$\text{Reflexivity} \quad x \sqsubseteq x \quad (1)$$

$$\text{Transitivity} \quad \neg(x \sqsubseteq y) \vee \neg(y \sqsubseteq z) \vee x \sqsubseteq z \quad (2)$$

$$\text{Anti-Symmetry} \quad \neg(x \sqsubseteq y) \vee \neg(y \sqsubseteq x) \vee x \simeq y \quad (3)$$

$$\text{Monotonicity} \quad \neg(x \sqsubseteq y) \vee f(x) \sqsubseteq f(y) \quad (4)$$

$$\text{Tree-Property} \quad \neg(z \sqsubseteq x) \vee \neg(z \sqsubseteq y) \vee x \sqsubseteq y \vee y \sqsubseteq x \quad (5)$$

Multiple inheritance: $MI = \{(1), (2), (3), (4)\}$

Single inheritance: $SI = MI \cup \{(5)\}$

Concrete examples of decision procedures

DPLL($\Gamma+\mathcal{T}$) with addition of $f^j(x) \simeq f^k(x)$ for $j > k$ decides the satisfiability modulo \mathcal{T} of problems

- ▶ $MI \cup P$
- ▶ $SI \cup P$
- ▶ $MI \cup TR \cup P$ and $SI \cup TR \cup P$

where $TR = \{\neg(g(x) \simeq null), h(g(x)) \simeq x\}$ has only infinite models!

(because g is injective, since it has left inverse, but not surjective, since there is no pre-image for *null*)

III part: Discussion of current trends in the field

- ▶ Integration of search and inference in first-order theories
- ▶ CDCL beyond propositional logic?
- ▶ MBTC beyond linear integer arithmetic?

Model-constructing satisfiability procedures (MCsat)

- ▶ Satisfiability **modulo assignment** (SMA)
- ▶ M : both L (means $L \leftarrow true$) and $x \leftarrow 3$
- ▶ CDCL + MBTC
- ▶ Theory CDCL: **explain** theory conflicts and theory propagations
- ▶ Beyond input literals: finite bag for termination
- ▶ Equality, lists, arrays, linear arithmetic (rationals)

Example of theory explanation (equality)

$$F = \{\dots, v \simeq f(a), w \simeq f(b), \dots\}$$

$$M = \dots a \leftarrow \alpha \quad b \leftarrow \alpha \quad w \leftarrow \beta_1 \quad v \leftarrow \beta_2 \dots$$

Conflict!

Explain by $a \simeq b \supset f(a) \simeq f(b)$
(instance of substitutivity)

Example of theory explanation (arithmetic) I

$$F = \{x \geq 2, \neg(x \geq 1) \vee y \geq 1, x^2 + y^2 \leq 1 \vee xy > 1\}$$

- ▶ $M = \emptyset$
- ▶ Propagation: $M = x \geq 2$
- ▶ Theory Propagation: $M = x \geq 2, x \geq 1$
- ▶ Boolean Propagation: $M = x \geq 2, x \geq 1, y \geq 1$
- ▶ Boolean Decision: $M = x \geq 2, x \geq 1, y \geq 1, x^2 + y^2 \leq 1$
- ▶ Semantic Decision:
 $M = x \geq 2, x \geq 1, y \geq 1, x^2 + y^2 \leq 1, x \leftarrow 2$
- ▶ Conflict!: no value for y such that $4 + y^2 \leq 1$

Example of theory explanation (arithmetic) II

$$F = \{x \geq 2, \neg(x \geq 1) \vee y \geq 1, x^2 + y^2 \leq 1 \vee xy > 1\}$$

- ▶ Assume we'd learn $\neg(x = 2)$:

$$M = x \geq 2, x \geq 1, y \geq 1, x^2 + y^2 \leq 1, \neg(x = 2)$$

- ▶ Semantic Decision:

$$M = x \geq 2, x \geq 1, y \geq 1, x^2 + y^2 \leq 1, \neg(x = 2), x \leftarrow 3$$

- ▶ Another conflict!
- ▶ We don't want to learn $\neg(x = 2), \neg(x = 3), \neg(x = 4) \dots$!

Example of theory explanation (arithmetic) III

$$F = \{x \geq 2, \neg(x \geq 1) \vee y \geq 1, x^2 + y^2 \leq 1 \vee xy > 1\}$$

- ▶ Solution: **theory explanation** by **interpolation**
- ▶ $x^2 + y^2 \leq 1$ implies $-1 \leq x \wedge x \leq 1$ which is inconsistent with $x = 2$
- ▶ Learn $\neg(x^2 + y^2 \leq 1) \vee x \leq 1$
- ▶ $M = x \geq 2, x \geq 1, y \geq 1, x^2 + y^2 \leq 1, x \leq 1$

Example of theory explanation (arithmetic) IV

$$F = \{x \geq 2, \neg(x \geq 1) \vee y \geq 1, x^2 + y^2 \leq 1 \vee xy > 1\}$$

- ▶ $M = x \geq 2, x \geq 1, y \geq 1, x^2 + y^2 \leq 1, x \leq 1$
- ▶ Theory conflict: $x \geq 2$ and $x \leq 1$
- ▶ Learn lemma: $\neg(x \geq 2) \vee \neg(x \leq 1)$
- ▶ Boolean Explanation (by resolution): $\neg(x^2 + y^2 \leq 1) \vee x \leq 1$
and $\neg(x \geq 2) \vee \neg(x \leq 1)$ yield $\neg(x^2 + y^2 \leq 1) \vee \neg(x \geq 2)$
- ▶ Boolean Explanation (by resolution):
 $\neg(x^2 + y^2 \leq 1) \vee \neg(x \geq 2)$ and $x \geq 2$ yield $\neg(x^2 + y^2 \leq 1)$
- ▶ $M = x \geq 2, x \geq 1, y \geq 1, \neg(x^2 + y^2 \leq 1)$

Recent trends in model-based reasoning

- ▶ **Deduction** guides **search for model**
- ▶ **Candidate model** guides **deduction**

- ▶ Propositional CDCL (both DPLL and DPLL(\mathcal{T}))
- ▶ Model-based theory combination (MBTC)
- ▶ DPLL($\Gamma + \mathcal{T}$)
- ▶ CDCL for arithmetic (aka Natural domain SMT)
- ▶ Model-constructing satisfiability procedures (MCsat)

Ideas for future work

- ▶ MCsat procedures for more first-order theories
e.g., Boolean algebra with Presburger arithmetic (BAPA)
- ▶ More decision procedures by speculative inferences
- ▶ MCsat + Γ