On Model-Based Reasoning
Recent Trends and Current Developments

Maria Paola Bonacina

Dipartimento di Informatica
Università degli Studi di Verona
Verona, Italy

September, 2013

1 Joint work with Leonardo de Moura
Model-based reasoning

\text{DPLL}(\Gamma + \mathcal{T})$: algorithmic reasoner + first-order prover

\text{DPLL}(\Gamma + \mathcal{T}) + speculative inferences: Decision procedures

Current and future work
The gist of this talk

- Automated reasoning from proofs to models
- Models are relevant to applications (e.g., program testing, program synthesis)
- Theorem provers that terminate on satisfiable inputs (Decision procedures)
- Trade-off between decidability and expressivity
Automated reasoning

- Logico-deductive reasoning
- Other kinds: Probabilistic ...

Outline
Model-based reasoning
DPLL(Γ + T): algorithmic reasoner + first-order prover
DPLL(Γ + T) + speculative inferences: Decision procedures
Current and future work
Logico-deductive reasoning

- **Proofs and Models**
- **Theorem Proving**
  - Validity: $\mathcal{T} \models \varphi$
  - Refutationally: $\mathcal{T} \cup \{\neg \varphi\}$ unsatisfiable
  - If not: $\mathcal{T}$-model of $\neg \varphi$, counter-example for $\varphi$
- **Model Building**
  - Satisfiability: is there a $\mathcal{T}$-model of $\varphi$?
  - 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:
  - Hyper-resolution [Alan Robinson 1965]
  - Set of support [Larry Wos et al. 1965]
  - Semantic resolution [James Slagle 1967]
Algorithmic reasoning (Decision procedures)

- Satisfiability decidable: Symmetry restored
- Propositional logic
- Decidable (fragments of) first-order theories
  - QFF: equality, recursive data structures, arrays
  - Linear arithmetic (integers, rationals), arithmetic (reals)
Symmetry in the reasoner’s operations

- Deduction guides search for model
- Candidate partial model guides deduction
- How?
Propositional logic (SAT)

- Davis-Putnam-Logemann-Loveland (DPLL) procedure
  - [Martin Davis and Hilary Putnam 1960]
  - [Martin Davis and George Logemann and Donald Loveland 1962]
- Backtracking search for model
- State of derivation: $M \| F$
  - $M$: sequence of truth assignments
  - $F$: clauses to satisfy
Conflict-Driven Clause Learning (CDCL)

- **Conflict**: $M$ falsifies clause $L_1 \lor \ldots \lor L_n$: conflict clause
- **Explain**: resolve and get another conflict clause
  
  
  $L_1 \lor \ldots \lor L_n$
  
  $\neg L_1 \lor Q_2 \ldots \lor Q_k$

- **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


[Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang and Sharad Malik 2001]
Example of CDCL

\[ F = \{ \neg a \lor b, \neg c \lor d, \neg e \lor \neg f, f \lor \neg e \lor \neg b \} \]
\[ M = a \ b \ c \ d \ e \ \neg f \]
blue: assignments; violet: propagations

Conflict: \( f \lor \neg e \lor \neg b \)
Explain by resolving \( f \lor \neg e \lor \neg b \) and \( \neg e \lor \neg f \): \( \neg e \lor \neg b \)
Learn \( \neg e \lor \neg b \): 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: \( M = a \ b \ c \ d \ \neg e \)
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

[Robert Nieuwenhuis, Albert Oliveras and Cesare Tinelli 2006]
Theory combination by equality sharing

- Theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$
- $\mathcal{T} = \bigcup_{i=1}^{n} \mathcal{T}_i$
- $\mathcal{T}_i$-satisfiability procedures
- Disjoint: share only $\simeq$ and uninterpreted constants
- Need to compute **arrangement**: which shared constants are equal and which are not
- Conservative approach: propagate all entailed (disjunctions of) equalities between shared constants

[Greg Nelson and Derek C. Oppen 1979]
Model-based theory combination (MBTC)

- Every $\mathcal{T}_i$-satisfiability procedure builds a $\mathcal{T}_i$-model
- 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

[Leonardo de Moura and Nikolaj Bjørner 2007]
CDCL for $\exists$-fragments of arithmetic

- Linear arithmetic (rationals)
  [Ken McMillan, A. Kuehlmann and Mooly Sagiv 2009]
  [Konstantin Korovin, Nestan Tsiskaridze and Andrei Voronkov 2009] [Scott Cotton 2010]

- Linear arithmetic (integers)
  [Dejan Jovanović and Leonardo de Moura 2011]

- Non-linear arithmetic (reals)
  [Dejan Jovanović and Leonardo de Moura 2012]

- Floating-point binary arithmetic
  [Leopold Haller, Alberto Griggio, Martin Brain and Daniel Kroening 2012]
Model-constructing satisfiability procedures (MCsat)

- Satisfiability *modulo assignment* (SMA)
  - $M$: both $L$ (means $L \leftarrow \text{true}$) 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)

[Leonardo de Moura and Dejan Jovanović 2013]

[Dejan Jovanović, Clark Barrett and Leonardo de Moura 2013]
Example of theory explanation (equality)

\[ F = \{\ldots, v \simeq f(a), w \simeq f(b), \ldots\} \]

\[ M = \ldots a \leftarrow \alpha \quad b \leftarrow \alpha \quad w \leftarrow \beta_1 \quad v \leftarrow \beta_2 \ldots \]

Conflict!

Explain by \( a \simeq b \supset f(a) \simeq f(b) \)

(instance of substitutivity)
Summary: Recent trends in model-based reasoning

- Deduction guides search for model
- Candidate model guides deduction

- Propositional CDCL (both DPLL and DPLL(Γ))
- Model-based theory combination (MBTC)
- CDCL for arithmetic (aka Natural domain SMT)
- Model-constructing satisfiability procedures (MCsat)
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: $\mathcal{R} \cup \mathcal{P}$
  - $\mathcal{R}$: set of non-ground clauses without $\mathcal{T}$-symbols
  - $\mathcal{P}$: large ground formula (set of ground clauses) typically with $\mathcal{T}$-symbols
- Determine whether $\mathcal{R} \cup \mathcal{P}$ is satisfiable modulo $\mathcal{T}$
DPLL(Γ+T): integrate Γ in DPLL(T)

- Superposition-based inference system Γ:
  - **FOL+=** clauses with universally quantified variables
  - **Expansion**: generate clauses (resolution, superposition)
  - **Contraction**: delete redundant clauses (subsumption, simplification)
  - **Well-founded** ordering and literal **selection**
  - Decision procedure for several theories of data structures (e.g., lists, arrays, records)

- **Model-based deduction**: literals in M as premises of Γ-inferences!

[Alessandro Armando, Maria Paola Bonacina, Silvio Ranise and Stephan Schulz 2009]

[Leonardo de Moura and Nikolaj Bjørner 2008]
Hypothetical clauses

- Literals from \( M \) used as premises of \( \Gamma \)-inferences stored as hypotheses in inferred clause:
  \[(L_1 \land \ldots \land L_n) \triangleright (L'_1 \lor \ldots \lor L'_m)\]
  interpreted as
  \[\neg L_1 \lor \ldots \lor \neg L_n \lor L'_1 \lor \ldots \lor L'_m\]
- Inferred clauses inherit hypotheses from premises
- Backjump: remove hypothetical clauses depending on undone assignments
DPLL(Γ+T): expansion inferences

- If non-ground clauses \( C_1, \ldots, C_m \) and ground \( \mathcal{R} \)-literals \( L_{m+1}, \ldots, L_n \) generate \( C \):
  \[
  H_1 \triangleright C_1, \ldots, H_m \triangleright C_m \quad \text{and} \quad L_{m+1}, \ldots, L_n \quad \text{in } M \quad \text{generate} \quad H_1 \cup \ldots \cup H_m \cup \{L_{m+1}, \ldots, L_n\} \triangleright C
  \]
- Only \( \mathcal{R} \)-literals: \( \Gamma \)-inferences ignore \( \mathcal{T} \)-literals
- Take ground unit \( \mathcal{R} \)-clauses from \( M \) as MBTC puts them there
DPLL(Γ+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
- If non-ground clauses $C_1, \ldots, C_m$ and ground $\mathcal{R}$-literals $L_{m+1}, \ldots, L_n$ simplify $C$ to $C'$
  - $H_1 \triangleright C_1, \ldots, H_m \triangleright C_m$ and $L_{m+1}, \ldots, L_n$ in $M$ simplify $H \triangleright C$ to $H \cup H_1 \cup \ldots \cup H_m \cup \{L_{m+1}, \ldots, L_n\} \triangleright C'$
    - If $\text{level}(H) \geq \text{level}(H')$: delete
    - If $\text{level}(H) < \text{level}(H')$: disable
      (re-enable when backjumping $\text{level}(H')$)
Completeness of DPLL($\Gamma + \mathcal{T}$)

- **Refutational completeness** of the inference system:
  - From that of $\Gamma$, 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($\Gamma + \mathcal{T}$)
Outline
Model-based reasoning
DPLL(Γ+Σ): algorithmic reasoner + first-order prover
DPLL(Γ+Σ) + speculative inferences: Decision procedures
Current and future work

DPLL(Γ+Σ): Summary

Use each engine for what is best at:

- DPLL(Σ) works on ground clauses and built-in theory
- Γ works on non-ground clauses and ground unit clauses taken from M: Γ works on 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. $\neg (x \sqsubseteq y) \lor f(x) \sqsubseteq f(y)$ (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. \( \neg (x \sqsubseteq y) \lor f(x) \sqsubseteq f(y) \)

2. \( a \sqsubseteq b \)

3. \( a \sqsubseteq f(c) \)

4. \( \neg (a \sqsubseteq c) \)
Idea: Allow speculative inferences

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

1. Add $f(x) \simeq x$
2. Rewrite $a \subseteq f(c)$ into $a \subseteq c$ and get $\square$: backtrack!
Idea: Allow speculative inferences

1. \( \neg (x \sqsubseteq y) \lor f(x) \sqsubseteq f(y) \)
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 \( \Box \): 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(Γ+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. \( \neg (x \sqsubseteq y) \lor f(x) \sqsubseteq f(y) \)
2. \( a \sqsubseteq b \)
3. \( a \sqsubseteq f(c) \)
4. \( \neg (a \sqsubseteq c) \)
Example as done by system

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

1. Add \([f(x) \simeq x] \triangleright f(x) \simeq x\)
2. Rewrite \(a \sqsubseteq f(c)\) into \([f(x) \simeq x] \triangleright a \sqsubseteq c\)
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 $\lceil f(x) \simeq x \rceil \triangleright \Box$; Backtrack, learn $\neg \lceil f(x) \simeq x \rceil$
Example as done by system

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

1. Add \( [f(x) \simeq x] \triangleright f(x) \simeq x \)
2. Rewrite \( a \sqsubseteq f(c) \) into \( [f(x) \simeq x] \triangleright a \sqsubseteq c \)
3. Generate \( [f(x) \simeq x] \triangleright \Box \); Backtrack, learn \( \neg[f(x) \simeq x] \)
4. Add \( [f(f(x)) \simeq x] \triangleright f(f(x)) \simeq x \)
5. \( a \sqsubseteq b \) yields only \( f(a) \sqsubseteq f(b) \)
6. \( a \sqsubseteq f(c) \) yields only \( [f(f(x)) = x] \triangleright f(a) \sqsubseteq c \)
7. Terminate and detect satisfiability
Outline
Model-based reasoning
DPLL($\Gamma + T$): algorithmic reasoner + first-order prover
DPLL($\Gamma + T$) + speculative inferences: Decision procedures
Current and future work

Decision procedures with speculative inferences

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

- Find sequence of speculative axioms $U$
- Show that there exists $k$ s.t. $k$-bounded DPLL($\Gamma + T$) is guaranteed to terminate
  - returning Unsat if $R \cup P$ is $T$-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 $j \neq k$
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**
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 $\Gamma$ and $\mathcal{R}$
- Only finitely many clauses generated: termination
Situations where clause length is limited

\( \Gamma \): 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

Reflexivity \[ x \sqsubseteq x \]  (1)
Transitivity \[ \neg(x \sqsubseteq y) \lor \neg(y \sqsubseteq z) \lor x \sqsubseteq z \]  (2)
Anti-Symmetry \[ \neg(x \sqsubseteq y) \lor \neg(y \sqsubseteq x) \lor x \simeq y \]  (3)
Monotonicity \[ \neg(x \sqsubseteq y) \lor f(x) \sqsubseteq f(y) \]  (4)
Tree-Property \[ \neg(z \sqsubseteq x) \lor \neg(z \sqsubseteq y) \lor x \sqsubseteq y \lor y \sqsubseteq x \]  (5)

Multiple inheritance: \( \text{MI} = \{(1), (2), (3), (4)\} \)
Single inheritance: \( \text{SI} = \text{MI} \cup \{(5)\} \)
Concrete examples of decision procedures

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

- $\text{MI} \cup P$
- $\text{SI} \cup P$
- $\text{MI} \cup \text{TR} \cup P$ and $\text{SI} \cup \text{TR} \cup P$

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

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

[Maria Paola Bonacina, Chris Lynch and Leonardo de Moura 2011]
Current and future work

- MCsat procedures for more first-order theories
e.g., Boolean algebra with Presburger arithmetic (BAPA)
- Many-sorted DPLL(Γ+T)
- Weakening conditions for completeness
- More decision procedures by speculative inferences
- MCsat + Γ

[Joint work with Serdar Erbatur]