Conflict-driven reasoning[1]

Maria Paola Bonacina

Dipartimento di Informatica, Università degli Studi di Verona,
Verona, Italy, EU

Invited talk
24th UK Automated Reasoning Workshop (ARW)
Department of Computer Science, The University of Bristol
Bristol, England, UK

3rd April 2017

[1] Joint work with Stéphane Graham-Lengrand and Natarajan Shankar

Background: Theorem proving

- Assumptions: H
- Conjecture: φ
- Problem: $H \models^? \varphi$
  Refutation: is $H \cup \{\neg\varphi\}$ unsatisfiable?
- $H \cup \{\neg\varphi\} \rightsquigarrow S$ set of clauses (machine format)
- Yes, with proof $S \vdash \bot$ that reveals inconsistency
  $\neg\varphi$ unsatisfiable in H, φ valid in H
- No, with model of S, counter-example for φ
  $\neg\varphi$ satisfiable in H, φ invalid in H

Background: Model building/constraint solving

- Set of constraints: H
- Additional constraint: φ
- Problem: is there a model/solution of $H \cup \{\varphi\}$?
- $H \cup \{\varphi\} \rightsquigarrow S$ set of clauses (machine format)
- Yes, with model of S
  φ satisfiable in H, $\neg\varphi$ invalid in H
- No, with proof $S \vdash \bot$
  φ unsatisfiable in H, $\neg\varphi$ valid in H

Background: Proofs and models

- Theorem proving and model building/constraint solving
- Proofs and models
- Are two sides of the same coin
- Both involve inference and search

Background: applications

- Verification: a program state is a model, proof of verification conditions
- Testing: models as "moles" in automated test generation
- Synthesis: proof of synthesis conditions, models as examples in example-driven synthesis
- Reasoning support to model checkers (e.g., abstraction refinement), static analyzers (e.g., invariant generation)
- Reasoning as a back-end enabling technology

Background: Decision procedures

- A procedure that takes as input the set of clauses S and is guaranteed to return
  - Yes with a model, if S is satisfiable
  - No with a proof, if S is unsatisfiable
- Is a decision procedure for satisfiability/validity
- Decision procedures are needed for applications where reasoner is invoked by another software

The quest

- SAT: satisfiability of a set of clauses in propositional logic
- Conflict-Driven Clause Learning (CDCL) procedure [Marques-Silva, Sakallah: ICCAD 1996, IEEE Trans. on Computers 1999], [Moskewicz, Madigan, Zhao, Zhang, Malik: DAC 2001] [Marques-Silva, Lynce, Malik: SAT Handbook 2009]
- CDCL is conflict-driven SAT-solving
- CDCL brought SAT-solving from theoretical hardness to practical success
- Quest: conflict-driven reasoning beyond SAT-solving?

What is a conflict?

- Conflict: between a candidate partial model and constraints
- Methods that build a candidate partial model: model-based reasoning

Model-based reasoning

- A reasoning method is model-based if it works with a candidate (partial) model
- The state of the derivation includes a representation of the current candidate model
- Inferences transform the candidate model
- The candidate model drives the inferences

Conflict-driven reasoning

- Conflict: one of the clauses is false in the current candidate model
- A model-based reasoning method is conflict-driven if inferences
  - Explain the conflict
  - Solve the conflict repairing the model

A taste of CDCL: decide and propagate

$\{\neg a \vee b,\ \neg c \vee d,\ \neg e \vee \neg f,\ f \vee \neg e \vee \neg b\} \subseteq S$

1. Decide: a is true; Propagate: b must be true
2. Decide: c is true; Propagate: d must be true
3. Decide: e is true; Propagate: $\neg f$ must be true

- $M = a,\ b,\ c,\ d,\ e,\ \neg f$
- Conflict: $f \vee \neg e \vee \neg b$ is false

A taste of CDCL: explain, learn, backjump

$\{\neg a \vee b,\ \neg c \vee d,\ \neg e \vee \neg f,\ f \vee \neg e \vee \neg b\} \subseteq S$
$M \ = \ a, \ \ b, \ \ c, \ \ d, \ \ e, \ \ \neg f$

1. Conflict: $f \vee \neg e \vee \neg b$
2. Explain by resolving $f \vee \neg e \vee \neg b$ with $\neg e \vee \neg f$: $\neg e \vee \neg b$
3. Learn $\neg e \vee \neg b$: no model with e and b true
4. Backjump to earliest state with $\neg b$ false and $\neg e$ unassigned:
   $M \ = \ a, \ \ b, \ \ \neg e$
5. Continue until it finds a satisfying assignment (model) or none can be found (conflict at level 0)

More general conflict-driven reasoning

Conflict-driven reasoning from SAT to arithmetic

Conflict-driven reasoning in fragments of arithmetic

- $\mathcal{T}$-satisfiability procedure: decides satisfiability of a set of ground literals in theory $\mathcal{T}$
- Conflict-driven $\mathcal{T}$-satisfiability procedures for fragments of arithmetic:
  - Linear rational arithmetic: [McMillan, Kuehlmann, Sagiv: CAV 2009], [Korovin, Tsiskaridze, Voronkov: CP 2009], [Cotton: FORMATS 2010]
  - Linear integer arithmetic: [Jovanović, de Moura: CADE 2011]
  - Non-linear arithmetic: [Jovanović, de Moura: IJCAR 2012]
  - Floating-point binary arithmetic: [Haller, Griggio, Brain, Kroening: FMCAD 2012]

First-order assignments

- CDCL: the trail is a sequence of literals
- Example: $M = a, \;\; b, \;\; \neg e$
- Equivalently: $M = a \leftarrow \mathit{true}, \;\; b \leftarrow \mathit{true}, \;\; \neg e \leftarrow \mathit{true}$
- Conflict-driven $\mathcal{T}$-satisfiability procedures for fragments of arithmetic: assignments to first-order variables
- Example: $M = x \leftarrow 3, \;\; y \leftarrow -2, \;\; z \leftarrow 0$

More general conflict-driven reasoning

Conflict-driven reasoning from SAT to SMT: MCSAT

Conflict-driven reasoning for SMT

- SMT: Satisfiability Modulo Theories
- $\mathcal{T}$-decision procedure: decides satisfiability of an arbitrary quantifier-free formula, or equivalently a set of ground clauses, in theory $\mathcal{T}$
- SAT-solving + theory reasoning in a quantifier-free fragment
- Conflict-driven $\mathcal{T}$-decision procedures: Model Constructing Satisfiability (MCSAT)
  - One generic theory [Jovanović, de Moura: VMCAI 2013]
  - A specific combination: propositional logic + linear rational arithmetic + equality [Jovanović, Barrett, de Moura: FMCAD 2013]

Model-constructing satisfiability: MCSAT

- CDCL-based SAT-solver + conflict-driven $\mathcal{T}$-satisfiability procedure: cooperate on the same level
- M: both L (means $L \leftarrow true$) and $x \leftarrow 3$
- Any $\mathcal{T}$ equipped with clausal inference rules to explain theory conflicts
- Such inferences may introduce new atoms
- Beyond input literals: finite basis for termination

Example of theory explanation (equality)

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

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

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 $\neg(x = 2),\ \neg(x = 3),\ \neg(x = 4) \ldots$!

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

More general conflict-driven reasoning

Conflict-driven reasoning for combinations of theories: CDSAT

Conflict-driven satisfiability: CDSAT

- A framework for conflict-driven $\mathcal{T}$-decision procedures
- For $\mathcal{T}$ a generic combination of theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$
- Disjoint theories: share only $\simeq$ and uninterpreted constants
- Propositional logic is one of them
- CDSAT generalizes both
  - MCSAT: combination by explicit model construction, and
  - Equality sharing (aka Nelson-Oppen): combination of $\mathcal{T}$-satisfiability procedures as black-boxes

Let's start with an example

- $\{f(select(store(a,i,v),j)) \simeq w,\ f(u) \simeq w-2,\ i \simeq j,\ u \simeq v\}$
- Combination of
  - Linear rational arithmetic (LRA)
  - Equality (EUF)
  - Arrays (Arr)

Example (continued)

- LRA has sorts $\{prop, Q\}$; $\simeq$ on each sort; $0, 1\colon Q$; $+\colon Q \times Q \to Q$; $c\cdot\colon Q \to Q$ for all rational number c
- EUF has sorts $\{prop, Q, V\}$; $\simeq$ on each sort; $f\colon V \to Q$
- Arr has sorts $\{prop, V, I, A\}$; $\simeq$ on each sort; $select\colon A \times I \to V$; $store\colon A \times I \times V \to A$

Everything is assignment

$f(select(store(a,i,v),j)) \simeq w \ \leftarrow true$
$f(u) \simeq w{-}2 \ \leftarrow true$
$i \simeq j \ \leftarrow true$
$u \simeq v \ \leftarrow true$

Assignment

- Assignments to propositional variables: $L \leftarrow \mathit{true}$
- Assignments to first-order variables: $x \leftarrow 3$
- Assignments to first-order terms: $\mathit{select}(a, i) \leftarrow 3$
- Assignments to first-order atoms, literals, clauses ... all seen as first-order terms of sort *prop*: $a \geq b \leftarrow \mathit{true}$, $P(a, b) \leftarrow \mathit{false}$

Assignment

- $\{t_1 \leftarrow \alpha_1, \ldots, t_m \leftarrow \alpha_m\}$
- $t_1, \ldots, t_m$: terms
- $\alpha_1, \ldots, \alpha_m$: values
- α_i has the same sort as t_i
- $t_i \leftarrow \alpha_i$ is a $\mathcal{T}_1$-assignment
- $t_j \leftarrow \alpha_j$ is a $\mathcal{T}_2$-assignment
- What are values? 3, $\sqrt{2}$ are not in the signature of the theory

Theory extension

- Theory $\mathcal{T}$
- Theory extension $\mathcal{T}^+$: add new constant symbols
- Example: add a constant symbol for every number; $\sqrt{2}$ is a constant symbol interpreted as $\sqrt{2}$
- The values in assignments are these constant symbols (also for *true* and *false*)
- Conservative theory extension: a $\mathcal{T}^+$-unsatisfiable set of $\mathcal{T}$-formulas is $\mathcal{T}$-unsatisfiable

Public sorts

- A sort s is public for theory $\mathcal{T}$ ($\mathcal{T}$-public)
- If $\mathcal{T}^+$ adds new constants of sort s
- There are values of sort s that can appear on the right hand side of an assignment in the trail shared by all theories

More on assignments

- Does not contain $L \leftarrow true$ and $L \leftarrow false$
- Abbreviations: L for $L \leftarrow true$, $\overline{L}$ for $L \leftarrow false$, $t_1 \not\simeq t_2$ for $t_1 \simeq t_2 \leftarrow false$
- Flipping an assignment: from L to $\overline{L}$ or vice versa

Theory view of an assignment

- Theory $\mathcal{T}$
- Assignment: $\{t_1 \leftarrow \alpha_1, \ldots, t_m \leftarrow \alpha_m\}$
- $\mathcal{T}$-view:
  - The $\mathcal{T}$-assignments
  - $t_1 \simeq t_2$ if there are $t_1 \leftarrow \alpha$ and $t_2 \leftarrow \alpha$ by any theory
  - $t_1 \not\simeq t_2$ if there are $t_1 \leftarrow \alpha$ and $t_2 \leftarrow \beta$ by any theory

Theory modules

- Theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$
- Equipped with theory modules $\mathcal{I}_1, \ldots, \mathcal{I}_n$
- Abstraction of theory solver, theory plugin
- $\mathcal{I}_k$ is the inference system for $\mathcal{T}_k$
- $\mathcal{I}_k$-inferences transforms assignments

Examples of inferences

- Theory of arithmetic on the reals (RA)
- $(x \leftarrow \sqrt{2}),\ (y \leftarrow \sqrt{2}) \vdash (x \times y \simeq 1 + 1)$
- $(y \leftarrow \sqrt{2}),\ (x \leftarrow \sqrt{2}) \vdash (y \simeq x)$
- $(y \leftarrow \sqrt{2}),\ (x \leftarrow \sqrt{3}) \vdash (y \not\simeq x)$

Inferences in theory modules

- $J \vdash L$
- J is an assignment
- L is a singleton Boolean assignment
- Only Boolean assignments are inferred
- Getting $y \leftarrow 2$ from $x \leftarrow 1$ and $(x + y) \leftarrow 3$ is not an inference

Equality inferences

- All theory modules include equality inferences
- $t_1 \leftarrow \alpha,\ t_2 \leftarrow \alpha \vdash t_1 \simeq t_2$
- $t_1 \leftarrow \alpha,\ t_2 \leftarrow \beta \vdash t_1 \not\simeq t_2$
- $\vdash t \simeq t$
- $t_1 \simeq t_2 \vdash t_2 \simeq t_1$
- $t_1 \simeq t_2,\ t_2 \simeq t_3 \vdash t_1 \simeq t_3$

We have theory modules for

- Propositional logic
- Linear rational arithmetic (LRA)
- Equality (EUF)
- Arrays (Arr)
- Any stably infinite theory $\mathcal{T}$ equipped with a $\mathcal{T}$-satisfiability procedure:
  - Stably infinite: requirement for equality sharing
  - $\{t_1 \leftarrow \alpha_1, \ldots, t_m \leftarrow \alpha_m\} \vdash_{\mathcal{T}} \bot$

Acceptability

- Given assignment $J = \{t_1 \leftarrow \alpha_1, \ldots, t_m \leftarrow \alpha_m\}$ and theory module $\mathcal{I}$ for theory $\mathcal{T}$
- Assignment $t \leftarrow \beta$ is acceptable for J and $\mathcal{I}$ if
  - J does not already assign a $\mathcal{T}$-value to t and
  - It does not happen $J \cup \{t \leftarrow \beta\} \vdash_{\mathcal{I}} L$ with $\overline{L}$ in J

Relevance

- Given assignment $J = \{t_1 \leftarrow \alpha_1, \ldots, t_m \leftarrow \alpha_m\}$ and theory $\mathcal{T}$
- A term is $\mathcal{T}$-relevant if
  - it appears in J (also as subterm) and has a $\mathcal{T}$-public sort
  - or it is an equality $t_1 \simeq t_2$ whose sides appear in J and whose sort is a sort of $\mathcal{T}$ but it is not $\mathcal{T}$-public

Examples of relevant terms

- $J = \{x \leftarrow \sqrt{5},\ f(x) \leftarrow \sqrt{2},\ f(y) \leftarrow \sqrt{3}\}$
- x and y of sort real are RA-relevant not EUF-relevant
- $x \simeq y$ is EUF-relevant not RA-relevant
- Subdivision of labor among theories: RA can make x and y equal/different by assigning them the same/different value; EUF decides the truth value of $x \simeq y$

The CDSAT transition system

- Trail: sequence of assignments some of which are marked as decisions
- Explanation function: maps every assignment that is not a decision to a set of preceding assignments: $\mathit{expl}(A) \vdash_{\mathcal{I}} A$

The CDSAT transition system

- Search mode and Conflict resolution mode
- Search rules: Decide, Propagate, Conflict, Fail
- Conflict resolution rules: Resolve, Backjump, SemSplit, Undo
- Finite global basis for termination

Example of CDSAT derivation I

$$F = \{f(select(store(a,i,v),j)) \simeq w,\ \ f(u) \simeq w{-}2,\ \ i \simeq j,\ \ u \simeq v\}$$

- Decisions: $u \leftarrow \alpha, \quad v \leftarrow \alpha$
- Decisions: $select(store(a,i,v),j) \leftarrow \alpha, \quad w \leftarrow 0$
- Decisions: $f(select(store(a,i,v),j)) \leftarrow 0, \quad f(u) \leftarrow -2$
- Propagations:
  $u \simeq select(store(a,i,v),j), \quad f(u) \not\simeq f(select(store(a,i,v),j))$
- Conflict!: $u \simeq x,\ \ f(u) \not\simeq f(x) \vdash_{EUF} \bot$
- Backjump: flip $f(u) \not\simeq f(select(store(a,i,v),j))$ and clears the trail saving the explanation of $u \simeq select(store(a,i,v),j)$

Example of CDSAT derivation II

$$F = \{f(select(store(a,i,v),j)) \simeq w,\ f(u) \simeq w{-}2,\ i \simeq j,\ u \simeq v\}$$

- Decisions: $u \leftarrow \alpha, \quad v \leftarrow \alpha$
- Decision: $select(store(a,i,v),j) \leftarrow \alpha$
- Propagations:
  $u \simeq select(store(a,i,v),j), \quad f(u) \simeq f(select(store(a,i,v),j))$
- Propagations: $f(u) \simeq w, \quad w-2 \simeq w$ by transitivity of equality
- Conflict!: $\vdash_{LRA} w-2 \not\simeq w$

Summary of results

- Soundness: if CDSAT returns unsatisfiable, there is no model
- Termination: CDSAT is guaranteed to terminate if the global basis is finite
- Completeness: if CDSAT terminates without returning unsatisfiable, there is a model
- Satisfiability modulo assignments (SMA): first-order assignments as part of the input
- CDSAT: conflict-driven SMA-solving in generic combinations of theories

Summary of the big picture

- Emergence of a general paradigm of conflict-driven reasoning
- CDCL: conflict-driven SAT-solving
- Conflict-driven $\mathcal{T}$-satisfiability procedures in arithmetic
- MCSAT: conflict-driven SMT-solving
- CDSAT: conflict-driven SMA-solving
- SGGS: conflict-driven theorem proving in first-order logic

References

- Maria Paola Bonacina, Stéphane Graham-Lengrand, and Natarajan Shankar. Satisfiability modulo theories and assignments. Submitted, 1–16, February 2017.
- Maria Paola Bonacina, Stéphane Graham-Lengrand, and Natarajan Shankar. A model-constructing framework for theory combination. Research Report No. 99/2016, Dipartimento di Informatica, Università degli Studi di Verona, and Technical Report, SRI International, and CNRS–INRIA–École Polytechnique, November 2016 (revised February 2017), 1–49.

Thanks

Thank you!