CDSAT: Conflict-Driven SATisfiability modulo theories and assignments[1]

Maria Paola Bonacina

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

Invited talk at the Institute of Software, Chinese Academy of Sciences, Beijing,
and at the School of Computer Science and Software Engineering, East China Normal University, Shanghai,
PR China, April-May 2018

(And 1st half of a one-day tutorial on "Conflict-driven reasoning," LORIA Nancy, France, EU, February 2019)

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

Archetype of conflict-driven reasoning: DPLL-CDCL

- SAT: satisfiability of a set of clauses in propositional logic
- Conflict-Driven Clause Learning (CDCL) procedure
  [Marques-Silva, Sakallah: ICCAD 1996]
  [Marques-Silva, Sakallah: 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

A taste of DPLL-CDCL: decisions and propagations

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

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

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

A taste of CDCL: conflict-solving

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

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 level with $\neg b$ false and $\neg e$ unassigned:
   $\Gamma\ =\ a,\ \ b,\ \ \neg e$
5. Continue until it finds a satisfying assignment (model) or none can be found (conflict at level 0)

Conflict-driven reasoning in fragments of arithmetic

- Early forerunners, e.g.:
  - LPSAT [Wolfman, Weld: IJCAI 1999]
  - Separation logic [Wang, Ivančić, Ganai, Gupta: LPAR 2005]
- Linear rational arithmetic, e.g.:
  - Generalized DPLL [McMillan, Kuehlmann, Sagiv: CAV 2009]
  - Conflict Resolution [Korovin, Tsiskaridze, Voronkov: CP 2009]
  - Natural domain SMT [Cotton: FORMATS 2010]
- Linear integer arithmetic, e.g.:
  Cutting-to-the-chase method [Jovanović, de Moura: CADE 2011]
- Non-linear arithmetic, e.g.:
  NLSAT [Jovanović, de Moura: IJCAR 2012]
- Floating-point binary arithmetic, e.g.:
  Systematic abstraction [Haller, Griggio, Brain, Kroening: FMCAD 2012]

Conflict-driven $\mathcal{T}$-satisfiability procedures

- $\mathcal{T}$-satisfiability procedure: decides satisfiability of a set of literals in the quantifier-free fragment of a theory $\mathcal{T}$
- Conflict-driven $\mathcal{T}$-satisfiability procedures generalize CDCL with at least two key features:
  - Assignments to first-order variables
  - Explanation of conflicts with lemmas containing new atoms (i.e., non-input)

Example in linear rational arithmetic

$$R = \{L_0 : (-2x - y < 0),\ L_1 : (x + y < 0),\ L_2 : (x < -1)\}$$

1. Decide a first-order assignment: $y \leftarrow 0$;
2. Deduce: L_0 yields $x > 0$
3. Conflict between $x > 0$ and L_2
4. Explanation: infer $-y < -2$ by the linear combination of L_0 and L_2 that eliminates x
   $-y < -2$ is a new (non-input) atom
   that excludes not only $y \leftarrow 0$, but all assignments $y \leftarrow \mathfrak{c}$ where $\mathfrak{c} \leq 2$

From sets of literals to arbitrary QF formulas

- How to combine a conflict-driven $\mathcal{T}$-satisfiability procedure with DPLL-CDCL to decide the satisfiability of an arbitrary formula in the quantifier-free fragment of theory $\mathcal{T}$?
- Using the standard DPLL($\mathcal{T}$) framework? [Nieuwenhuis, Oliveras, Tinelli: JACM 2006]
  No: it allows neither first-order assignment nor new atoms on the trail
- MCSAT [de Moura, Jovanović: VMCAI 2013]

Open questions

Problems from applications require combinations of theories:

- How to combine multiple conflict-driven $\mathcal{T}$-satisfiability procedures with DPLL-CDCL?
- Better: How to combine multiple conflict-driven $\mathcal{T}$-satisfiability procedure one of which is DPLL-CDCL?
- Which requirements should theories and procedures satisfy to ensure soundness, completeness, and termination of the conflict-driven combination?

Answer: the new system CDSAT (Conflict-Driven SATisfiability)

Classical approach to theory combination: equality sharing

Equality sharing aka Nelson-Oppen method
[Nelson, Oppen: ACM TOPLAS 1979]

- Given theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$ with $\mathcal{T}_k$-satisfiability procedures
- Get $\mathcal{T}_\infty$-satisfiability procedure for $\mathcal{T}_\infty = \bigcup_{k=1}^{n} \mathcal{T}_k$
- Disjoint theories: share only $\simeq$ (and sorts)
- Mixed terms handled by introducing new variables or viewing as variables maximal subterms with foreign root symbol
- The $\mathcal{T}_k$-satisfiability procedures need to agree on:
  - Which shared variables are equal
  - Cardinalities of shared sorts

Theory combination by equality sharing

- For cardinality: assume stably infinite: every $\mathcal{T}_k$-satisfiable ground formula has $\mathcal{T}_k$-model with infinite cardinality
- For equality: compute an arrangement saying which shared variables are equal and which are not by letting the $\mathcal{T}_k$-satisfiability procedures generate and propagate all entailed (disjunctions of) equalities between shared variables
- Minimize interaction: the $\mathcal{T}_k$-satisfiability procedures are treated as black-boxes
- Integrated in DPLL($\mathcal{T}$) with new atoms on the trail only for equalities between shared variables [Barrett, Nieuwenhuis, Oliveras, Tinelli: LPAR 2006] [Krstić, Goel: FroCoS 2007]

More open questions

- Conflict-driven behavior and black-box integration are at odds: a conflict-driven $\mathcal{T}_k$-satisfiability procedure needs to access the trail and performs inferences to explain conflicts on a par with DPLL-CDCL
- How can we combine multiple $\mathcal{T}_k$-satisfiability procedures some conflict-driven and some not?

Answer: the new system CDSAT (Conflict-Driven SATisfiability)

What is CDSAT (Conflict-Driven SATisfiability)

- CDSAT is a new method for theory combination
- CDSAT generalizes conflict-driven reasoning to generic combinations of disjoint theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$
- CDSAT solves the problem of combining multiple $\mathcal{T}_k$-satisfiability procedures some conflict-driven and some not into a conflict-driven $\mathcal{T}$-satisfiability procedure for $\mathcal{T}_\infty = \bigcup_{k=1}^{n} \mathcal{T}_k$
- CDSAT reduces to equality sharing if no $\mathcal{T}_k$-satisfiability procedure is conflict-driven

Basic features of CDSAT

- CDSAT treats propositional and theory reasoning uniformly: formulas are terms of sort prop
- Propositional logic is one of $\mathcal{T}_1, \ldots, \mathcal{T}_n$
  DPLL-CDCL is one of the $\mathcal{T}_k$-satisfiability procedures
- With formulas reduced to terms, assignments become the basic data for inferences
- CDSAT combines inference systems called theory modules $\mathcal{I}_1, \ldots, \mathcal{I}_n$ for $\mathcal{T}_1, \ldots, \mathcal{T}_n$
- CDSAT treats a non-conflict-driven $\mathcal{T}_k$-satisfiability procedure as a theory module whose only inference rule invokes the procedure to detect $\mathcal{T}_k$-unsatisfiability
- CDSAT is sound, complete, and terminating

In CDSAT everything is assignment

- $P = \{f(select(store(a,i,v),j)) \simeq w,\ f(u) \simeq w-2,\ i \simeq j,\ u \simeq v\}$
- $P = \{\ f(select(store(a,i,v),j)) \simeq w\ \leftarrow true$
  $f(u) \simeq w{-}2\ \leftarrow true$
  $i \simeq j\ \leftarrow true$
  $u \simeq v\ \leftarrow true\ \}$
- Combination of the theories of Equality (EUF), Linear Rational Arithmetic (LRA), and Arrays (Arr)
- EUF and Arr share the sort of array values
- EUF and LRA share the sort of rational numbers

Beyond propositional variables and Boolean values

- Assignments to propositional variables: $L \leftarrow \mathit{true}$
- Assignments to first-order variables: $x \leftarrow 3$, $y \leftarrow \sqrt{2}$
- 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: $a \geq b \ \leftarrow \mathit{true}$
  $P(a, b) \ \leftarrow \mathit{false} \qquad a \geq b \vee P(a, b) \ \leftarrow \mathit{true}$
  all theories feature sort prop
- L stands for $L \leftarrow \mathit{true}$, $t_1 \not\simeq t_2$ stands for $t_1 \simeq t_2 \leftarrow \mathit{false}$
  $\overline{L}$ is the flip of L
- What are values? 3, $\sqrt{2}$ are not in the signature of any theory

Theory extension

- Theory extension $\mathcal{T}_k^+$ of theory $\mathcal{T}_k$: add new constant symbols (and possibly new axioms)
- Example: add a constant symbol for every number (e.g., integers, rationals, algebraic reals) $\sqrt{2}$ is a constant symbol interpreted as $\sqrt{2}$
- The values in assignments are these constant symbols, called $\mathcal{T}_k$-values (*true* and *false* are values for all theories)
- Conservative theory extension: a $\mathcal{T}_k^+$-unsatisfiable set of $\mathcal{T}_k$-formulas is $\mathcal{T}_k$-unsatisfiable
- $\mathcal{T}_\infty^+ = \bigcup_{k=1}^n \mathcal{T}_k^+$ extension of $\mathcal{T}_\infty = \bigcup_{k=1}^n \mathcal{T}_k$

Assignment

- $\{t_1 \leftarrow \mathfrak{c}_1, \ldots, t_m \leftarrow \mathfrak{c}_m\}$
- $t_1, \ldots, t_m$: $\mathcal{T}_\infty$-terms
- $\mathfrak{c}_1, \ldots, \mathfrak{c}_m$: values
- $\mathfrak{c}_i$ has the same sort as t_i
- $t_i \leftarrow \mathfrak{c}_i$ is a $\mathcal{T}_k$-assignment if $\mathfrak{c}_i$ is a $\mathcal{T}_k$-value
- An assignment must be plausible: it does not contain $L \leftarrow \mathit{true}$ and $L \leftarrow \mathit{false}$
- All theories may contribute: e.g., $t_i \leftarrow \mathit{true}$ is a $\mathcal{T}_1$-assignment, $t_j \leftarrow 3$ is a $\mathcal{T}_2$-assignment, $t_h \leftarrow \sqrt{2}$ is a $\mathcal{T}_3$-assignment

Problems as assignments

- Boolean assignment: Boolean values
- First-order assignment: non-Boolean values
- Satisfiability Modulo Theory problem: a plausible Boolean assignment
- Satisfiability Modulo theory and Assignment problem: a plausible assignment with both Boolean and first-order assignments

Theory view of an assignment

Let $\mathcal{T}$ stand for either $\mathcal{T}_k$, for any k, $1 \leq k \leq n$, or $\mathcal{T}_\infty$

$\mathcal{T}_\infty$-assignment: $H = \{t_1 \leftarrow \mathfrak{c}_1, \ldots, t_m \leftarrow \mathfrak{c}_m\}$

The $\mathcal{T}$-view of H is the $\mathcal{T}$-assignment made of:

- The $\mathcal{T}$-assignments in H
- $u \simeq t$ if H includes $\mathcal{T}_j$-assignments $(1 \leq j \leq n)$ $u \leftarrow \mathfrak{c}$ and $t \leftarrow \mathfrak{c}$ of a sort known to $\mathcal{T}$
- $u \not\simeq t$ if H includes $\mathcal{T}_j$-assignments $(1 \leq j \leq n)$ $u \leftarrow \mathfrak{c}$ and $t \leftarrow \mathfrak{q}$ of a sort known to $\mathcal{T}$ $(\mathfrak{c} \neq \mathfrak{q})$

Examples of theory views

$H = \{y \leftarrow -1,\ z \leftarrow 2,\ x > 1,\ store(a,i,v) \simeq b,$
$select(a,j) \leftarrow \mathrm{red}\}$

- Bool-view: $\{x > 1,\ store(a,i,v) \simeq b\}$
- Arr-view: $\{x > 1,\ store(a,i,v) \simeq b,\ select(a,j) \leftarrow \mathrm{red}\}$
- LRA-view:
  $\{x > 1,\ store(a,i,v) \simeq b,\ y \leftarrow -1,\ z \leftarrow 2,\ y \neq z\}$
- EUF-view: $\{x > 1,\ store(a,i,v) \simeq b,\ y \neq z\}$ assuming EUF has the sort of the rational numbers
- Global view: $H \cup \{y \neq z\}$

Assignments and models: endorsement

- Let $\mathcal{T}$ stand for either $\mathcal{T}_k$, for any k, $1 \leq k \leq n$, or $\mathcal{T}_\infty$
- What does it mean that a $\mathcal{T}^+$-model $\mathcal{M}$ satisfies a $\mathcal{T}$-assignment?
- $\mathcal{T}^+$-model $\mathcal{M}$ endorses $\mathcal{T}$-assignment $u \leftarrow \mathfrak{c}$ if $\mathcal{M}$ interprets u and $\mathfrak{c}$ as the same element
- $\mathcal{T}^+$-model $\mathcal{M}$ satisfies $\mathcal{T}$-assignment J if $\mathcal{M}$ endorses the $\mathcal{T}$-view of J

Another example

- $\{t \leftarrow 3.1, u \leftarrow 5.4, t \leftarrow \text{red}, u \leftarrow \text{blue}\} \subseteq H$
- $t \leftarrow 3.1$ and $u \leftarrow 5.4$ are $\mathcal{T}_1$-assignments
- $t \leftarrow \text{red}$ and $u \leftarrow \text{blue}$ are $\mathcal{T}_2$-assignments
- $\mathcal{T}_1$ and $\mathcal{T}_2$ share the sort of t and u
- Both $\mathcal{T}_1^+$ and $\mathcal{T}_2^+$ provide values for this sort
- The $\mathcal{T}_1$-view of H includes $\{t \leftarrow 3.1, u \leftarrow 5.4, t \neq u\}$
- The $\mathcal{T}_2$-view of H includes $\{t \leftarrow \text{red}, u \leftarrow \text{blue}, t \neq u\}$
- A combined model that identifies 3.1 with red and 5.4 with blue can satisfy H

Theory modules

- Theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$
- Equipped with theory modules $\mathcal{I}_1, \ldots, \mathcal{I}_n$
- $\mathcal{I}_k$ is an inference system for $\mathcal{T}_k$
- $\mathcal{I}_k$-inferences transforms assignments
- Examples in arithmetic on the reals (RA):
  - $(x \leftarrow \sqrt{2}),\ (y \leftarrow \sqrt{2}) \vdash (x \cdot y \simeq 1 + 1)$
  - $(y \leftarrow \sqrt{2}),\ (x \leftarrow \sqrt{2}) \vdash (y \simeq x)$
  - $(y \leftarrow \sqrt{2}),\ (x \leftarrow \sqrt{3}) \vdash (y \not\simeq x)$

Inferences in theory modules

- Inference $J \vdash L$
- 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 viewed as a forced decision in CDSAT

Equality inferences

All theory modules include equality inferences:

- Same value: $u \leftarrow \mathfrak{c},\ t \leftarrow \mathfrak{c} \vdash u \simeq t$
- Different values: $u \leftarrow \mathfrak{c},\ t \leftarrow \mathfrak{q} \vdash u \not\simeq t$
- Reflexivity: $\vdash t \simeq t$
- Symmetry: $t \simeq u \vdash u \simeq t$
- Transitivity: $t \simeq s,\ s \simeq u \vdash t \simeq u$

How about decisions?

Module $\mathcal{I}_k$ decides a value for term u if u is relevant to theory $\mathcal{T}_k$:

- $H = \{x \leftarrow 5,\ f(x) \leftarrow 2,\ f(y) \leftarrow 3\}$
- Rational variables x and y are LRA-relevant, not EUF-relevant
- $x \simeq y$ is EUF-relevant (assume EUF has sort Q), not LRA-relevant
- LRA can make x and y equal/different by assigning them the same/different value
- EUF can make x and y equal/different by deciding the truth value of $x \simeq y$

Two ways to communicate an equality: making it *true* and assigning the same value to its sides

Acceptability

Given $\mathcal{T}_k$-assignment J (e.g., the $\mathcal{T}_k$-view of the trail)

Assignment $u \leftarrow \mathfrak{c}$ is acceptable for J and the $\mathcal{T}_k$-module $\mathcal{I}_k$ if

1. u is relevant to $\mathcal{T}_k$
2. J does not already assign a $\mathcal{T}_k$-value to u
3. For $u \leftarrow \mathfrak{c}$ first-order, it does not happen $J' \cup \{u \leftarrow \mathfrak{c}\} \vdash_{\mathcal{I}_k} L$, where $J' \subseteq J$ and $\overline{L} \in J$

We have theory modules for

- Propositional logic
- Linear rational arithmetic (LRA)
- Equality (EUF)
- Arrays (Arr) – first time conflict-driven
- Any stably infinite theory $\mathcal{T}_k$ equipped with a $\mathcal{T}_k$-satisfiability procedure that detects the $\mathcal{T}_k$-unsatisfiability of a set of Boolean assignments:
  $\{L_1 \leftarrow \mathfrak{b}_1, \ldots, L_m \leftarrow \mathfrak{b}_m\} \vdash_{\mathcal{T}_k} \bot$

The CDSAT trail

- Trail: sequence of assignments that are either decisions or justified assignments
- Decisions can be either Boolean or first-order
- A justified assignment A has a justification that is a set of assignments that appear before A in the trail:
  - Due to inferences, e.g., $J \vdash_{\mathcal{I}_k} A$
  - Input assignments (empty justification)
  - Due to conflict-solving transitions
  - Boolean except the input first-order assignments of an SMA problem

The CDSAT trail

- Every assignment has a level
- The level of a decision is defined as in CDCL
- The level of a justified assignment is that of its justification
- The level of a justification is the maximum among those of its elements
- The CDSAT trail is not a stack: there may be late propagations

The CDSAT transition system

- Trail rules: Decide, Deduce, Fail, ConflictSolve
- Conflict state rules: UndoClear, Resolve, Backjump, UndoDecide
- Parameter: global basis:
  - A set from which CDSAT can draw new terms
  - Finite to ensure termination
  - Depends on the input and is fixed throughout a CDSAT derivation

Trail rules

- Apply to the trail Γ
- Decide: adds an acceptable assignment
- Deduce: adds L with justification J if $J \vdash_{\mathcal{I}_k} L$
- Conflict: $J \vdash_{\mathcal{I}_k} L$ and $\overline{L}$ is on the trail
  $J \cup \{\overline{L}\}$ is the conflict
- Fail: declares unsatisfiability if the level of the conflict is 0
- ConflictSolve: solves a conflict of level > 0 by calling the conflict state rules

Conflict state rules

- Apply to trail and conflict: $\langle\Gamma, H\rangle$ with $H \subseteq \Gamma$
- If $H = E \uplus \{A\}$ and $level(A) = m$ is greater than $level(E)$:
  - UndoClear: A is a first-order decision
    remove A and all assignments of level $\geq m$
    (i.e., backjump to $m - 1$)
  - Backjump: A is a Boolean L
    backjump to $level(E)$ and add $\overline{L}$ with justification E
    if $E \uplus \{L\} \vdash \bot$ then $E \vdash \overline{L}$

Example of UndoClear

$\Gamma \;=\; -2x - y < 0, \;\; x + y < 0, \;\; x < -1$ (level 0)

1. Decide $y \leftarrow 0$ (level 1)
2. Deduce $-y < -2$ from $-2x - y < 0$ and $x < -1$ (level 0)
3. Conflict is $\{y \leftarrow 0, -y < -2\}$
4. UndoClear removes $y \leftarrow 0$ resulting in
   $\Gamma \;=\; -2x - y < 0, \;\; x + y < 0, \;\; x < -1, \;\; -y < -2$ (level 0)
5. $-y < -2$ is a late propagation

Example of Backjump

$\Gamma = f(select(store(a,i,v),j)) \simeq w, \;\; f(u) \simeq w-2, \;\; i \simeq j, \;\; u \simeq v$ (level 0)

- Decide: $u \leftarrow \mathfrak{c}$ (level 1) $v \leftarrow \mathfrak{c}$ (level 2)
- Decide: $select(store(a,i,v),j) \leftarrow \mathfrak{c}$ (level 3) $w \leftarrow 0$ (level 4)
- Decide: $f(select(store(a,i,v),j)) \leftarrow 0$ (level 5) $f(u) \leftarrow -2$ (level 6)
- Deduce: $u \simeq select(store(a,i,v),j)$ (level 3) $f(u) \not\simeq f(select(store(a,i,v),j))$ (level 6)
- Conflict: the last two yield $\perp$ in $\mathcal{I}_{EUF}$
- Backjumps to level 3 and adds $f(u) \simeq f(select(store(a,i,v),j))$ with $u \simeq select(store(a,i,v),j)$ as justification

Conflict state rules

- Apply to trail and conflict: $\langle \Gamma, H \rangle$ with $H \subseteq \Gamma$
- If $H = E \uplus \{A\}$ and A has justification J
  Resolve transforms H into $E \uplus \{J\}$, provided J does not contain a first-oder decision A' of the same level as H to avoid looping with an UndoClear-Decide-Deduce sequence
- If $H = E \uplus \{L\}$, L is Boolean (no UndoClear), $level(L) = level(E)$ (no Backjump), and L has justification J that contains such an A' (no Resolve)
  UndoDecide undoes A' and decides $\bar{L}$

Example of Resolve

$\Gamma \ = f(select(store(a,i,v),j)) \simeq w, \quad f(u) \simeq w-2, \quad i \simeq j, \quad u \simeq v$
(level 0)
$u \leftarrow \mathfrak{c}$ (level 1)
$v \leftarrow \mathfrak{c}$ (level 2)
$select(store(a,i,v),j) \leftarrow \mathfrak{c}$ (level 3)
$u \simeq select(store(a,i,v),j)$ (level 3)
$f(u) \simeq f(select(store(a,i,v),j))$ (level 3)

- Deduce: $f(u) \simeq w$ (level 3)
  $w-2 \simeq w$ (level 3)
  both by transitivity of equality
- Conflict: $w-2 \simeq w$ yields $\bot$ in $\mathcal{I}_{LRA}$
- Resolve: $f(u) \simeq w, \ f(u) \simeq w-2$

Example of UndoDecide

$\Gamma \;=\; x > 1 \vee y < 0,\;\; x < -1 \vee y > 0$ (level 0)

- Decide: $x \leftarrow 0$ (level 1)
- Deduce: $(x > 1) \leftarrow \mathit{false}$ (level 1)
  $(x < -1) \leftarrow \mathit{false}$ (level 1)
  $y < 0$ (level 1)
  $y > 0$ (level 1)
- Conflict: $0 < 0$
- Resolve: $\{y < 0,\; y > 0\}$
  $\{x > 1 \vee y < 0,\; x < -1 \vee y > 0,\; x > 1 \leftarrow \mathit{false},$
  $x < -1 \leftarrow \mathit{false}\}$

Example of UndoDecide (continued)

$\Gamma \;=\; x > 1 \vee y < 0, \;\; x < -1 \vee y > 0$ (level 0)

- UndoDecide: $x > 1$ (level 1)
- Decide: $x \leftarrow 2$ (level 2)
- Deduce: $(x < -1) \leftarrow \mathit{false}$ (level 2)
  $y > 0$ (level 2)
- Decide: $y \leftarrow 1$ (level 3)
- Deduce: $(y < 0) \leftarrow \mathit{false}$ (level 3)
- Satisfiable

Three main theorems

- 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

Current work

- Lemma learning
- Proof generation
- Completeness of the theory modules
- Construction of a global basis from local bases at the combined theories
  - Size of the global basis as a function of the sizes of the local bases

Current and future work

- CDSAT in C++: forthcoming SMT solver Eos
  (by Giulio Mazzi at U. Verona)
- Heuristic strategies to make decisions and prioritize theory inferences
- Efficient techniques to detect the applicability of theory inference rules and the acceptability of assignments
- More theory modules (e.g., real arithmetic from NLSAT [Jovanović, de Moura: IJCAR 2012])
- Complexity of a combination given the complexities of the theory procedures

References

- Satisfiability modulo theories and assignments. In the Proc. of CADE-26, LNAI 10395, 42–59, Springer, Aug. 2017.
- Proofs in conflict-driven theory combination. In the Proc. of the 7th ACM SIGPLAN Int. Conf. on Certified Programs and Proofs (CPP), ACM Press, 186–200, Jan. 2018.
- Conflict-driven satisfiability for theory combination: transition system and completeness. Journal of Automated Reasoning, volume in press, pages 1–31, published online January 4, 2019.
- Conflict-driven satisfiability for theory combination: modules, lemmas, and proofs. Journal article, in preparation.

Authors: Maria Paola Bonacina, Stéphane Graham-Lengrand, and Natarajan Shankar