CDSAT: conflict-driven theory combination[1]

Maria Paola Bonacina

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

17 July 2017

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

Archetype of conflict-driven reasoning: 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 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; 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

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

A taste of CDCL: conflict-solving

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

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)

Conflict-driven reasoning: what is a conflict?

- Conflict: between constraints to be satisfied and a candidate partial model
- 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 of a set of clauses
- 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

Two directions of generalization of CDCL

- Towards first-order logic
- Towards theory reasoning, SMT, and beyond

Towards first-order logic

- The Bernays-Schönfinkel class aka EPR
  - DPLL($\mathcal{SX}$)
    [Piskac, de Moura, Bjørner: JAR 2010]
  - NRCL (Non-Redundant Clause Learning)
    [Alagi, Weidenbach: FroCoS 2015]
- Full first-order logic (without equality)
  - SGGS (Semantically-Guided Goal-Sensitive reasoning)
    [Bonacina, Plaisted: JAR 2016, 2017]
  - Conflict-Resolution
    [Slaney, Woltzenlogel Paleo: JAR to appear]
    [Itegulov, Slaney, Woltzenlogel Paleo: CADE 2017]

Two directions of generalization of CDCL

- Towards first-order logic
- Towards theory reasoning, SMT, and beyond: this talk

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. Propagate: L_0 yields $x > 0$
3. Conflict between $x > 0$ and $L_2 : (x < -1)$
4. Explanation: deduce $-y < -2$ by the linear combination of L_0 and L_2 that eliminates x
   Note that $-y < -2$ 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 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
- Answer: MCSAT (Model-Constructing SATisfiability)
  [de Moura, Jovanović: VMCAI 2013]

Key features of MCSAT

- CDCL-based SAT-solver + conflict-driven $\mathcal{T}$-satisfiability procedure: cooperate on the same level
- Trail M: both L (meaning $L \leftarrow \mathit{true}$) and $x \leftarrow 3$
- Any $\mathcal{T}$ equipped with an inference system to explain theory conflicts
- Such inferences may introduce new atoms
- Beyond input literals: finite basis for termination
- MCSAT lifts CDCL to Satisfiability Modulo one Theory

Instances of MCSAT

- One generic theory
  [de Moura, Jovanović: VMCAI 2013]
- Equality + linear rational arithmetic
  [Jovanović, de Moura, Barrett: FMCAD 2013]
- Bit-vectors
  [Zeljić, Wintersteiger, Rümmer: SAT 2016]
  [Graham-Lengrand, Jovanović: SMT 2017]
- Equality + non-linear arithmetic (mixed integer-real problems)
  [Jovanović: VMCAI 2017]

Open questions

Problems from applications require combinations of theories:

- How to combine multiple conflict-driven $\mathcal{T}$-satisfiability procedures with CDCL?
- Better: How to combine multiple conflict-driven $\mathcal{T}$-satisfiability procedure one of which is CDCL?
- Equivalently: How to generalize MCSAT to generic combinations of theories?
- 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 $\mathcal{T}$-satisfiability procedure for $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
- Disjoint theories: share sorts, $\simeq$, uninterpreted constants
- Mixed terms separated by introducing new constants (e.g., $f(g(a)) \simeq b$ becomes $f(c) \simeq b \wedge g(a) \simeq c$, with c new, if f and g belong to different theories)
- The $\mathcal{T}_k$-satisfiability procedures need to agree on:
  - Shared constants
  - 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 constants are equal and which are not by letting the $\mathcal{T}_k$-satisfiability procedures generate and propagate all entailed (disjunctions of) equalities between shared constants
- Minimize interaction: the $\mathcal{T}_k$-satisfiability procedures are treated as black-boxes
- Integrated in DPLL($\mathcal{T}$) with new atoms only for equalities between shared constants [Barrett, Nieuwenhuis, Oliveras, Tinelli: LPAR 2006] [Krstić, Goel: FroCoS 2007]

More open questions

- Conflict-driven behavior and black-box behavior seem at odds: e.g., in MCSAT the $\mathcal{T}$-satisfiability procedure accesses the central trail and performs deductions to explain conflicts on a par with CDCL
- Can we generalize equality sharing to the case where the $\mathcal{T}_k$-satisfiability procedures are conflict-driven?
- How can we combine multiple $\mathcal{T}_k$-satisfiability procedures some conflict-driven and some black-boxes?

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 conflict-driven $\mathcal{T}_k$-satisfiability procedures into a conflict-driven $\mathcal{T}$-satisfiability procedure for $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
- CDSAT reduces to MCSAT if there are two theories:
  propositional logic with CDCL
  a $\mathcal{T}$ with a conflict-driven $\mathcal{T}$-satisfiability procedure

Basic features of CDSAT

- CDSAT treats propositional and theory reasoning uniformly: formulas are terms of sort prop; all theories have sort prop
- Propositional logic is one of $\mathcal{T}_1, \ldots, \mathcal{T}_n$
  CDCL is one of the $\mathcal{T}_k$-satisfiability procedures
- With formulas reduced to terms, assignments become the basic data for inferences
- Key abstraction: CDSAT combines inference systems called theory modules $\mathcal{I}_1, \ldots, \mathcal{I}_n$ for $\mathcal{T}_1, \ldots, \mathcal{T}_n$
- CDSAT is sound, complete, and terminating

How about black-box procedures?

- CDSAT treats a non-conflict-driven $\mathcal{T}_k$-satisfiability procedure as a theory module whose only inference rule invokes the procedure to detect the $\mathcal{T}_k$-unsatisfiability of a set of assignments
- Thus CDSAT generalizes equality sharing:
  CDSAT reduces to equality sharing, if none of the theories has a conflict-driven $\mathcal{T}$-satisfiability procedure

Running example

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

Combination of

- Equality (EUF)
- Linear rational arithmetic (LRA)
- Arrays (Arr)

Running example

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

Everything is assignment

Initial state of the trail:

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

abbreviated

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

One central public trail shared by all theories

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*:
  $a \geq b \ \leftarrow \mathit{true} \qquad P(a, b) \ \leftarrow \mathit{false}$
  $a \geq b \vee P(a, b) \ \leftarrow \mathit{true}$
- Abbreviations: L for $L \leftarrow \mathit{true}$, $\bar{L}$ for $L \leftarrow \mathit{false}$
  $t_1 \not\simeq t_2$ for $t_1 \simeq t_2 \leftarrow \mathit{false}$
- Flipping a Boolean assignment: from L to $\bar{L}$ or vice versa

Assignment

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

Theory extension

- Theory $\mathcal{T}_k$
- Theory extension $\mathcal{T}_k^+$: 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 $\mathcal{T}_k^+$-unsatisfiable set of $\mathcal{T}_k$-formulas is $\mathcal{T}_k$-unsatisfiable

Public sorts

- A sort s is public for theory $\mathcal{T}_k$ ($\mathcal{T}_k$-public)
- If $\mathcal{T}_k^+$ 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 central trail shared by all theories

Plausibility

- An assignment is plausible if
  it does not contain $L \leftarrow \mathit{true}$ and $L \leftarrow \mathit{false}$
- Assignments are required to be plausible
- A plausible assignment may contain
  $\{t \leftarrow 3.1,\ u \leftarrow 5.4,\ t \leftarrow \text{green},\ u \leftarrow \text{yellow}\}$
  two by $\mathcal{T}_1$ and two by $\mathcal{T}_2$:
  the sort of t and u is both $\mathcal{T}_1$-public and $\mathcal{T}_2$-public
  When building a model from this assignment
  3.1 is identified with green and 5.4 with yellow

Theory view of an assignment

Theory $\mathcal{T}$

Assignment: $H = \{t_1 \leftarrow \mathfrak{c}_1, \ldots, t_m \leftarrow \mathfrak{c}_m\}$

$\mathcal{T}$-view of H:

- The $\mathcal{T}$-assignments
- $t \simeq s$ if there are e.g. $t \leftarrow 3$ and $s \leftarrow 3$ by another theory
- $t \not\simeq s$ if there are e.g. $t \leftarrow 3$ and $s \leftarrow 4$ by another theory

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 the inference system for $\mathcal{T}_k$
- $\mathcal{I}_k$-inferences transforms assignments

Examples of inferences

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

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 $(x + y) \leftarrow 3$ does not need an inference

Equality inferences

All theory modules include equality inferences:

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

Acceptability

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

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

1. J does not already assign a value to t:
   - No repetition
   - No contradiction if $t \leftarrow \mathfrak{c}$ is Boolean
2. It does not happen $J' \cup \{t \leftarrow \mathfrak{c}\} \vdash_{\mathcal{I}_k} L$ where $J' \subseteq J$ and $\bar{L} \in J$

Relevance

Given assignment H (e.g., the trail) and theory $\mathcal{T}_k$
A term is $\mathcal{T}_k$-relevant if

- It appears in H (also as subterm) and has a $\mathcal{T}_k$-public sort
  E.g., $H = \{x \leftarrow \sqrt{5},\ f(x) \leftarrow \sqrt{2},\ f(y) \leftarrow \sqrt{3}\}$
  x of sort real is RA-relevant
- Or it is an equality $t \simeq s$ whose sides appear in H and have a sort which is a sort of $\mathcal{T}_k$ but it is not $\mathcal{T}_k$-public
  E.g., $H = \{x \leftarrow \sqrt{5},\ f(x) \leftarrow \sqrt{2},\ f(y) \leftarrow \sqrt{3}\}$
  $x \simeq y$ is EUF-relevant

Meaning of relevance

- $H = \{x \leftarrow \sqrt{5},\ f(x) \leftarrow \sqrt{2},\ f(y) \leftarrow \sqrt{3}\}$
- x and y of sort real are RA-relevant
- $x \simeq y$ is EUF-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 $x \simeq y$

We have theory modules for

- Propositional logic
- Linear rational arithmetic (LRA)
- Equality (EUF)
- Arrays (Arr)
- 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
- A justified assignment A has a justification J
- Justification: a set of assignments J that appear before A in the trail and yields A, e.g., by an inference $J \vdash_{\mathcal{I}_k} A$

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 inference system

- Search rules
- Conflict-resolution rules
- Finite global basis for termination

The search rules of CDSAT

- The search rules apply to the trail
- Decide: adds an acceptable assignment to a relevant term
- Deduce: adds L with justification J if $J \vdash_{\mathcal{I}_k} L$
- Conflict: $J \vdash_{\mathcal{I}_k} L$ and $\bar{L}$ is on the trail
  $J \cup \bar{L}$ 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-resolution rules

The conflict-resolution rules of CDSAT: backjumping rules

- The conflict-resolution rules apply to trail and conflict
- The conflict contains an assignment A of level n greater than that of the rest E of the conflict
- Undo: A is a first-order decision:
  remove A and all assignments of level $\geq n$
  (equivalently: backjump to $n-1$)
- Backjump: A is a Boolean assignment L:
  backjump to the level of E and add $\bar{L}$ with justification E:
  if $E \cup \{L\} \vdash \bot$ then $E \vdash \bar{L}$

Example of CDSAT derivation I

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

- Decide: $u \leftarrow \mathfrak{c},\ v \leftarrow \mathfrak{c}$
- Decide: $select(store(a,i,v),j) \leftarrow \mathfrak{c},\ w \leftarrow 0$
- Decide: $f(select(store(a,i,v),j)) \leftarrow 0,\ f(u) \leftarrow -2$
- Deduce: $u \simeq select(store(a,i,v),j)$, $f(u) \not\simeq f(select(store(a,i,v),j))$
- Conflict: the last two yield $\bot$ in $\mathcal{I}_{EUF}$
- Backjump: flips $f(u) \not\simeq f(select(store(a,i,v),j))$ and clears the trail saving $u \simeq select(store(a,i,v),j)$ and its justification

The conflict-resolution rules of CDSAT: explanation rules

- The explanation rules unfolds the conflict by replacing an assignment in the conflict E with its justification H
- The crux is whether H contains a first-order assignment A of the same level as E
- Resolve applies if it does not
  (to avoid a resolve-undo-decide-deduce loop as first-order assignments do not have a flip)
- UndoDecide applies if it does:
  there are two Boolean assignments L and F both depending on A, and the rule flips arbitrarily one of them

Example of CDSAT derivation II

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

- Decide: $u \leftarrow \mathfrak{c},\ v \leftarrow \mathfrak{c},\ select(store(a,i,v),j) \leftarrow \mathfrak{c}$
- Deduce: $u \simeq select(store(a,i,v),j)$
- Deduce: $f(u) \simeq f(select(store(a,i,v),j))$
- Deduce: $f(u) \simeq w,\ w-2 \simeq w$ by transitivity of equality
- Conflict: $w-2 \simeq w$ yields $\bot$ in $\mathcal{I}_{LRA}$
- Resolve: $f(u) \simeq w,\ f(u) \simeq w-2$
- Resolve: $f(u) \simeq f(select(store(a,i,v),j))$, $f(select(store(a,i,v),j)) \simeq w,\ f(u) \simeq w-2$
- Resolve: $u \simeq select(store(a,i,v),j)$, $f(select(store(a,i,v),j)) \simeq w,\ f(u) \simeq w-2$

Example of CDSAT derivation III

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

- Backjump: flips $u \simeq select(store(a,i,v),j)$ and jumps back to level 0
- $u \not\simeq select(store(a,i,v),j)$
- Decide: $u \leftarrow \mathfrak{c},\ v \leftarrow \mathfrak{c},\ select(store(a,i,v),j) \leftarrow \mathfrak{d}$
- Deduce: $v \not\simeq select(store(a,i,v),j)$
- Conflict: $i \simeq j,\ v \not\simeq select(store(a,i,v),j)$ yield $\bot$ in $\mathcal{I}_{Arr}$

Example of CDSAT derivation IV

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

- $u \not\simeq select(store(a, i, v), j)$
- Backjump: flips $v \not\simeq select(store(a, i, v), j)$ and jumps back to level 0
- $v \simeq select(store(a, i, v), j)$
- Conflict: $u \simeq v$, $u \not\simeq select(store(a, i, v), j)$, and $v \simeq select(store(a, i, v), j)$ yield $\bot$ at level 0
- Fail: P is unsatisfiable

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

Satisfiability Modulo Assignment (SMA)

- Satisfiability modulo assignment (SMA) is the problem of deciding the $\mathcal{T}$-satisfiability of a quantifier-free formula modulo an initial assignment of values to both Boolean and first-order variables:
  - Enumeration of models
  - Parallelization
  - Optimization [de Moura, Passmore: ADDCT 2013]
- CDSAT: conflict-driven SMA-solving in generic combinations of theories

References

- Maria Paola Bonacina, Stéphane Graham-Lengrand, and Natarajan Shankar. Satisfiability modulo theories and assignments. In the Proceedings of CADE-26, LNAI 10395, 42–59, Springer, August 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 June 2017), 1–48.