Conflict-Driven Reasoning in Unions of Theories[1]

Maria Paola Bonacina

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

Invited Keynote Speech
12th Int. Symposium on Frontiers of Combining Systems (FroCoS)
London, England, UK

4 September 2019

[1] Based on joint work with S. Graham-Lengrand and N. Shankar

Automated reasoning in unions of theories

- Problems from applications: decide $\mathcal{T}$-satisfiability for $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
- Disjoint theories and quantifier-free formulas
- Several approaches
- This talk advertises a general paradigm named CDSAT (Conflict-Driven SATisfiability):
  - Conflict-Driven reasoning in $\mathcal{T}$
  - By combining $\mathcal{T}_k$-inference systems: theory modules

Conflict-driven satisfiability

- Procedure to determine satisfiability of a formula
- Build candidate model
- Assignments + propagation through formulas
- Conflict btw model and formula: explain by inferences
- Learn generated lemma to avoid repetition
- Solve conflict by fixing model to satisfy learned lemma
- Nontrivial inferences on demand to respond to conflicts

CDSAT does this for a generic union $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$

Conflict-driven propositional satisfiability

- CDCL (Conflict-Driven Clause Learning) procedure for SAT
  [Marques Silva, Sakallah: ICCAD 1996, IEEE TOC 1999]
  [Davis, Putnam, Logeman, Loveland: JACM 1960, CACM 1962]:
  - Build candidate propositional model
  - Assignments to propositional variables + BCP
  - Explain conflicts by propositional resolution
  - Learn resolvents made of input atoms
  - Resolution on demand to respond to conflicts
- CDSAT: propositional logic as theory Bool
- CDSAT reduces to CDCL if $\mathcal{T} = \text{Bool}$

Conflict-driven satisfiability procedures in arithmetic

- Decide satisfiability of sets of literals
- Assignments to atoms and first-order variables ($x \leftarrow 3$)
- Explanation of conflicts by theory inferences
- Learn lemmas that may contain new (non-input) atoms
- Nontrivial theory inferences on demand to respond to conflicts

[Korovin, Tsiskaridze, Voronkov: CP 2009] [McMillan, Kuehlmann, Sagiv: CAV 2009] [Cotton: FORMATS 2010] [Jovanović, de Moura: JAR 2013] [Haller, Griggio, Brain, Kroening: FMCAD 2012] [Jovanović, de Moura: IJCAR 2012] [Braußе, Korovin, Korovina, Müller: FroCoS 2019]

Example: linear rational arithmetic

- Propagation as evaluation: $y \leftarrow 0 \vdash_{\mathsf{LRA}} \overline{y > 2}$
- Explanation of conflicts by Fourier-Motzkin (FM) resolution:
  $\{x < -y,\ -y < -2\} \vdash_{\mathsf{LRA}} x < -2$
  It generates new (non-input) atoms
- FM-resolution on demand to respond to conflicts

  [Korovin, Tsiskaridze, Voronkov: CP 2009] [McMillan, Kuehlmann, Sagiv: CAV 2009] [Cotton: FORMATS 2010]

CDSAT integrates LRA-module with inference rules including evaluation and FM-resolution

Standard theory combination: not conflict-driven

- Equality sharing method [Nelson, Oppen: ACM TOPLAS 1979]
- Combines $\mathcal{T}_k$-sat procedures as black-boxes that
  - Exchange entailed (disjunctions of) equalities between shared variables
  - Build arrangement that tells which shared variables are equal
- Stably infinite theories: infinite cardinality for shared sorts
- A $\mathcal{T}_k$-sat procedure could be conflict-driven, not the combination scheme

No conflict-driven $\mathcal{T}_k$-sat procedure: CDSAT emulates equality sharing as it accommodates also black-box procedures

From sets of literals to formulas

DPLL($\mathcal{T}$) aka CDCL($\mathcal{T}$) with $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
[Nieuwenhuis, Oliveras, Tinelli: JACM 2006] [Krstić, Goel: FroCoS 2007]

- CDCL builds candidate propositional model $\mathcal{M}$
- Satellite $\mathcal{T}_k$-satisfiability procedures
  - Combined by equality sharing as black-boxes
  - Signal $\mathcal{T}$-conflicts in $\mathcal{M}$ and contribute $\mathcal{T}$-lemmas
- Conflict-driven inferences: only propositional (resolution)

CDCL only conflict-driven procedure: CDSAT reduces to CDCL($\mathcal{T}$) with equality sharing

Model-based theory combination (MBTC)

- Model-based equality sharing [de Moura, Bjørner: SMT 2007]
  - $\mathcal{T}_k$-sat procedures build candidate models $\mathcal{M}_k$
  - Exchange equalities true in $\mathcal{M}_k$ (btw. terms occuring in the problem)
  - Not entailed: conflict, undo, update $\mathcal{M}_k$
- Model-based conflict-driven arrangement construction
- $\mathcal{M}_k$ and conflict-driven steps inside a black-box procedure

CDSAT lets model-constructing conflict-driven procedures cooperate to build a $\mathcal{T}$-model

Conflict-driven reasoning from sets of literals to formulas

- MCSAT (Model-Constructing SATisfiability) [de Moura, Jovanović: VMCAI 2013] [Jovanović, Barrett, de Moura: FMCAD 2013]
  - Integrates CDCL and one model-constructing conflict-driven $\mathcal{T}$-sat procedure (theory plugin)
  - CDCL and the $\mathcal{T}$-plugin cooperate in model construction
  - Both propositional and $\mathcal{T}$-reasoning are conflict-driven
- CDSAT generalizes MCSAT to generic $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
- CDSAT reduces to MCSAT if there are CDCL and one conflict-driven model-constructing $\mathcal{T}$-sat procedure

CDSAT: Conflict-driven reasoning from a theory to many

- Conflict-driven behavior and black-box integration are at odds: each conflict-driven $\mathcal{T}_k$-sat procedure needs to access the trail, post assignments, perform inferences, explain $\mathcal{T}_k$-conflicts, export lemmas on a par with CDCL
- Key abstraction in CDSAT: open the black-boxes, pull out the $\mathcal{T}_k$-inference systems used to explain $\mathcal{T}_k$-conflicts, and combine them in a conflict-driven way
- If $\mathcal{T}_k$ has no conflict-driven $\mathcal{T}_k$-sat procedure: black-box inference rule $L_1, \ldots, L_m \vdash_k \bot$ invokes the $\mathcal{T}_k$-procedure to detect $\mathcal{T}_k$-unsat

More about CDSAT

- SMA: Satisfiability Modulo theories and Assignments
  (allows first-order assignments such as $x \leftarrow 3$ in input)
- CDSAT does not require model-constructing $\mathcal{T}_k$-sat procedures in the strong sense of MBTC and MCSAT
- CDSAT does not require the theories to be stably infinite
  it suffices a leading theory that knows all sorts
- CDSAT is
  - Sound if all theory modules are
  - Terminating if all new terms come from a finite global basis
  - Complete if the theory modules are complete relative to the leading theory

Assignments of values to terms

- CDSAT treats propositional and theory reasoning similarly: formulas as terms of sort prop (from proposition)
- Assignments take center stage:
  - Boolean assignments to formulas
    first-order assignments to first-order terms
  - Mixed assignments: $(x > 1) \leftarrow$ false,
    $(x > 1) \vee (y < 0) \leftarrow$ true,
    $(store(a, i, v) \simeq b) \leftarrow$ true,
    $y \leftarrow -1$,
    $select(a, j) \leftarrow 3$
- What are values? 3, $\sqrt{2}$ are not in the signature

Theory extensions to define values

- From theory $\mathcal{T}_k$ to theory extension $\mathcal{T}_k^+$:
  - Add new constant symbols (and possibly axioms)
  - Ex.: add a constant symbol for every number (e.g., integers, rationals, algebraic reals)
    $\sqrt{2}$ is a constant symbol interpreted as $\sqrt{2}$
- Values in assignments are these constant symbols, called $\mathcal{T}_k$-values (true and false are values for all theories)
- $\mathcal{T}_k$-assignment: assigns $\mathcal{T}_k$-values
- Conservative theory extension: a $\mathcal{T}_k^+$-unsatisfiable set of $\mathcal{T}_k$-formulas is $\mathcal{T}_k$-unsatisfiable

Plausible assignment

- 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 $\mathcal{T}_2$
- When building a model from this assignment
  3.1 is identified with green and 5.4 with yellow

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

- The $\mathcal{T}_k$-view of an assignment H written H_k:
  - The $\mathcal{T}_k$-assignments in H: those that assign $\mathcal{T}_k$-values
  - $u \simeq t$ if there are $u \leftarrow \mathfrak{c}$ and $t \leftarrow \mathfrak{c}$ in H
  - $u \not\simeq t$ if there are $u \leftarrow \mathfrak{c}$ and $t \leftarrow \mathfrak{q}$ in H

  u and t of a sort known to $\mathcal{T}_k$
- Global view:
  - The $\mathcal{T}$-view of H for $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
  - $H_{\mathcal{T}}$ has everything

Examples of theory views

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

- $H_{\text{Bool}} = \{x > 1,\ store(a, i, v) \simeq b\}$
- $H_{\text{Arr}} = \{x > 1,\ store(a, i, v) \simeq b,\ select(a, j) \leftarrow \text{red}\}$
- $H_{\text{LRA}} = \{x > 1,\ store(a, i, v) \simeq b,\ y \leftarrow -1,\ z \leftarrow 2,\ y \not\simeq z\}$
- $H_{\text{EUF}} = \{x > 1,\ store(a, i, v) \simeq b,\ y \not\simeq z\}$
  assuming EUF has the sort of the rational numbers
- Global view: $H \cup \{y \not\simeq z\}$

Assignments and models: endorsement

- Model $\mathcal{M}$ endorses $u \leftarrow \mathfrak{c}$:
  $\mathcal{M}$ interprets u and $\mathfrak{c}$ as the same element
- Enough if the assignment is Boolean, otherwise:
- $u \leftarrow \mathfrak{c},\ t \leftarrow \mathfrak{c}$: $\mathcal{M}$ endorses $u \simeq t$
- $u \leftarrow \mathfrak{c},\ t \leftarrow \mathfrak{q}$: $\mathcal{M}$ endorses $u \not\simeq t$
  if $\mathcal{M}$ endorses the theory view
- $\mathcal{T}_k$-satisfiable: a $\mathcal{T}_k^+$-model endorses the $\mathcal{T}_k$-view
- $\mathcal{T}$-satisfiable: a $\mathcal{T}^+$-model endorses the global view
  (global endorsement)

Theory modules

- For theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$ theory modules $\mathcal{I}_1, \ldots, \mathcal{I}_n$
  - Inference $J \vdash_k L$
  - J is a $\mathcal{T}_k$-assignment
  - L is a singleton Boolean assignment:
  - Getting $y \leftarrow 2$ from $x \leftarrow 1$ and $(x + y) \leftarrow 3$ is a forced decision
- Sound inferences: if $J \vdash_k L$ then $J \models L$
- $J \models L$: if $\mathcal{M} \models J_k$ then $\mathcal{M} \models L$
- Local basis: $\text{basis}_k(X)$ contains all terms that $\mathcal{I}_k$ can generate from X

Equality inferences

All theory modules include equality inferences:

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

With first-order assignments, there are two ways to make $t \simeq s$ true: $(t \simeq s){\leftarrow}\text{true}$ and $t{\leftarrow}\mathfrak{c},\ s{\leftarrow}\mathfrak{c}$

Theory module for propositional logic

- $\Sigma_{\text{Bool}}=(\{\text{prop}\},\{\neg,\vee,\wedge,\simeq_{\text{prop}}\})$
- Bool^{+} adds true and false: trivial extension
- Evaluation: $(L_1\leftarrow\mathfrak{b}_1,\ldots,L_m\leftarrow\mathfrak{b}_m)\vdash_{\text{Bool}} L\leftarrow\mathfrak{b}$
- Negation: $\neg L\vdash_{\text{Bool}}\overline{L}$ and $\overline{\neg L}\vdash_{\text{Bool}} L$
- Conjunction: $\overline{L_1\vee\cdots\vee L_m}\vdash_{\text{Bool}}\overline{L_i}$ and $L_1\wedge\cdots\wedge L_m\vdash_{\text{Bool}} L_i$
- Unit propagation: $L_1\vee\cdots\vee L_m,\{\overline{L_j}\mid j\neq i\}\vdash_{\text{Bool}} L_i$ and $\overline{L_1\wedge\cdots\wedge L_m},\{L_j\mid j\neq i\}\vdash_{\text{Bool}}\overline{L_i}$
- $\text{basis}_{\text{Bool}}(X)$: all subformulas of formulas in X and all their disjunctions (for learning)

Theory module for equality

- $\Sigma_{\mathsf{EUF}}=(S,F)$, $\mathsf{prop}\in S$, $\simeq_S\subseteq F$
- EUF^+ may be trivial or add countably many values for each $s\in S\setminus\{\mathsf{prop}\}$ used as labels of congruence classes
- Congruence:
  - $(t_i\simeq u_i)_{i=1\ldots m},(f(t_1,\ldots,t_m)\not\simeq f(u_1,\ldots,u_m))\vdash_{\mathsf{EUF}}\bot$
  - $(t_i\simeq u_i)_{i=1\ldots m}\vdash_{\mathsf{EUF}} f(t_1,\ldots,t_m)\simeq f(u_1,\ldots,u_m)$
  - $(t_i\simeq u_i)_{i=1\ldots m,i\neq j},f(t_1,\ldots,t_m)\not\simeq f(u_1,\ldots,u_m)\vdash_{\mathsf{EUF}} t_j\not\simeq u_j$
- $\mathsf{basis}_{\mathsf{EUF}}(X)$: all subterms of terms in X and all equalities between them

Theory module for arrays

- $\Sigma_{\mathsf{Arr}}=(S,F)$, $S=\{\mathsf{prop}, I, V, \ldots, I \Rightarrow V, \ldots\}$
  $F= \simeq_S \cup \{\mathsf{select}_{I\Rightarrow V}, \mathsf{store}_{I\Rightarrow V}, \mathsf{diff}_{I\Rightarrow V}\}$
- Arr^+: like for EUF^+
- Inference rules corresponding to congruence axioms, select-over-store axioms, and extensionality axiom:
  - $a \not\simeq b \vdash_{\mathsf{Arr}} a[\mathsf{diff}(a,b)] \not\simeq b[\mathsf{diff}(a,b)]$
- $\mathsf{basis}_{\mathsf{Arr}}(X)$: all subterms of terms in X, equalities btw them, and witness terms $a[\mathsf{diff}(a,b)], b[\mathsf{diff}(a,b)]$

Theory module for linear arithmetic

- Σ_{LRA}: $S{=}\{\mathsf{prop}, \mathsf{Q}\}$, $F{=} \simeq_S \cup \{1, +, <, \leq, c\cdot\}$ for all $c \in \mathbb{Q}$
- LRA^+ adds constants $\tilde{q}$ for all rational numbers $q \in \mathbb{Q}$
- Evaluation: $(t_1 \leftarrow \tilde{q_1}, \ldots, t_m \leftarrow \tilde{q_m}) \vdash_{\mathsf{LRA}} l \leftarrow \mathfrak{b}$
- FM-resolution: $(t_1 \leq_1 x,\ x \leq_2 t_2) \vdash_{\mathsf{LRA}} t_1 \leq_3 t_2$
- Disequality elimination:
  $t_1 \leq x, x \leq t_2, t_1 \simeq_{\mathsf{Q}} t_0, t_2 \simeq_{\mathsf{Q}} t_0, x \not\simeq_{\mathsf{Q}} t_0 \vdash_{\mathsf{LRA}} \bot$
- $\mathsf{basis}_{\mathsf{LRA}}(X)$: subterms, equalities, disequalities restricting FM-resolution to resolve on the $\prec_{\mathsf{LRA}}$-maximum variable

CDSAT trail

- Sequence of assignments: decision or justified assignment
- Decision: either Boolean or first-order; opens the next level
- Justification of A: set H of assignments that appear before A
  - Due to an inference $H \vdash_k A$
  - Input assignment ($H = \emptyset$)
  - Due to conflict-solving transitions
  - Boolean or input first-order assignment in SMA
- Level of A: max among those of the elements of H
- A justified assignment of level 5 may appear after a decision of level 6: late propagation; a trail is not a stack

The CDSAT transition system

- Trail rules: Decide, Deduce, Fail, ConflictSolve
- Apply to the trail Γ
- Conflict state rules: UndoClear, Resolve, UndoDecide, Learn
- Apply to trail and conflict: $\langle \Gamma, H \rangle$ with $H \subseteq \Gamma$
- Conflict: H is an unsatisfiable assignment
- Parameter: finite global basis $\mathcal{B}$:
  - A set from which CDSAT can draw new terms
  - Used only to prove termination of CDSAT
  - Its existence can be shown from that of local bases

The CDSAT transition system: Decide

Decide: $\Gamma \longrightarrow \Gamma, {}_?(u \leftarrow \mathfrak{c})$
adds decision ${}_?(u \leftarrow \mathfrak{c})$
if $u \leftarrow \mathfrak{c}$ is an acceptable $\mathcal{T}_k$-assignment for $\mathcal{I}_k$ in Γ_k:

- Γ_k does not already assign a $\mathcal{T}_k$-value to u
- $u \leftarrow \mathfrak{c}$ first-order: it does not happen $J \cup \{u \leftarrow \mathfrak{c}\} \vdash_k L$ where $J \subseteq \Gamma_k$ and $\bar{L} \in \Gamma_k$
- u is relevant to $\mathcal{T}_k$:
  either u occurs in Γ_k and $\mathcal{T}_k$ has $\mathcal{T}_k$-values for its sort;
  or u is an equality whose sides occur in Γ_k,
  $\mathcal{T}_k$ has their sort, but not $\mathcal{T}_k$-values

Example: relevance

- $H = \{x \leftarrow 5,\ f(x) \leftarrow 2,\ f(y) \leftarrow 3\}$
- x, y: Q, f: Q $\rightarrow$ Q, LRA and EUF share sort Q
- $H_{\mathsf{LRA}} = H \cup \{x \not\simeq f(x),\ x \not\simeq f(y),\ f(x) \not\simeq f(y)\}$
- $H_{\mathsf{EUF}} = \{x \not\simeq f(x),\ x \not\simeq f(y),\ f(x) \not\simeq f(y)\}$
- x and y are LRA-relevant, not EUF-relevant
- $x \simeq y$ is EUF-relevant, not LRA-relevant
- LRA makes x and y equal/different by assigning them same/different values
- EUF makes x and y equal/different by assigning a truth value to $x \simeq y$

The CDSAT transition system: Deduce

Deduce: $\Gamma \longrightarrow \Gamma, {}_{J}\!\vdash L$

- Adds justified assignment ${}_{J}\!\vdash L$
  - $J \vdash_k L$, for some k, $1 \leq k \leq n$, $J \subseteq \Gamma$, and $L \notin \Gamma$
  - $\overline{L} \notin \Gamma$
  - L is in $\mathcal{B}$ (finite global basis)
- Both $\mathcal{T}_k$-propagation and explanation of $\mathcal{T}_k$-conflicts

The CDSAT transition system: Fail and ConflictSolve

- $J \vdash_k L$, for some k, $1 \leq k \leq n$, $J \subseteq \Gamma$, $L \notin \Gamma$
- $\overline{L} \in \Gamma$: $J \cup \{\overline{L}\}$ is a conflict
- If $\text{level}_\Gamma(J \cup \{\overline{L}\}) = 0$
  Fail: $\Gamma \longrightarrow$ unsat declares unsatisfiability
- If $\text{level}_\Gamma(J \cup \{\overline{L}\}) > 0$
  ConflictSolve: $\Gamma \longrightarrow \Gamma'$
  solves the conflict by calling the conflict-state rules
  $\langle \Gamma; J \cup \{\overline{L}\} \rangle \Longrightarrow^* \Gamma'$

The CDSAT transition system: UndoClear

The conflict contains a first-order assignment that stands out as its level is maximum in the conflict:

UndoClear: $\langle \Gamma; E \uplus \{A\} \rangle \Longrightarrow \Gamma^{\leq m-1}$

- A is a first-order decision of level $m > \text{level}_\Gamma(E)$
- Removes A and all assignments of level $\geq m$
- $\Gamma^{\leq m-1}$: the restriction of trail Γ to its elements of level at most $m-1$

Example: Deduce as explanation + UndoClear

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

1. Decide $y \leftarrow 0$ (level 1)
2. LRA-conflict: $\{-2{\cdot}x - y < 0,\ x < -1,\ y \leftarrow 0\}$
3. Explanation by FM-resolution:
   $\{-y < 2{\cdot}x,\ 2{\cdot}x < -2\} \vdash_{\mathsf{LRA}} -y < -2$
4. Deduce places $-y < -2$ on the trail (late propagation: level 0)
5. Evaluation: $y \leftarrow 0 \vdash_{\mathsf{LRA}} \overline{-y < -2}$
6. LRA-conflict: $\{y \leftarrow 0,\ -y < -2\}$
7. UndoClear removes $y \leftarrow 0$ resulting in
   $\Gamma \;=\; -2x - y < 0, \quad x + y < 0, \quad x < -1, \quad -y < -2$

Explanation of conflicts in CDSAT

- Explanation of a $\mathcal{T}_k$-conflict by $\mathcal{I}_k$-inferences encapsulated as Deduce steps: CDSAT not in conflict state
- Until the conflict surfaces as a Boolean conflict:
  $J \vdash_k L$ and $\overline{L} \in \Gamma$
  $J \cup \{\overline{L}\}$ is a conflict
- CDSAT switches to conflict state $\langle \Gamma; H \rangle$
- Explanation of conflict H by replacing justified assignments in H with their justifications: Resolve transition rule

The CDSAT transition system: Resolve

Resolve: $\langle \Gamma; E \uplus \{A\} \rangle \Longrightarrow \langle \Gamma; E \cup H \rangle$

- A is a justified assignment ${}_{H}\vdash A$
- Replace A by its justification H
- A can be a Boolean or a first-order assignment
- If A is first-order, it comes from the input ($H = \emptyset$):
  Resolve removes it from the conflict (not from the trail)

Example of Resolve

Γ_0 includes: $(\neg L_4 \vee L_5)$, $(\neg L_2 \vee \neg L_4 \vee \neg L_5)$ (level 0)

1. Decide: A_1 (level 1)
2. Decide: L_2 (level 2)
3. Decide: A_3 (level 3)
4. Decide: L_4 (level 4)
5. Deduce: L_5 with justification $\{\neg L_4 \vee L_5,\ L_4\}$ (level 4)
6. Conflict: $\{\neg L_2 \vee \neg L_4 \vee \neg L_5,\ L_2,\ L_4,\ L_5\}$
   $\neg L_2 \vee \neg L_4 \vee \neg L_5$ is the CDCL conflict clause
7. Resolve: $\{\neg L_2 \vee \neg L_4 \vee \neg L_5,\ L_2,\ L_4,\ \neg L_4 \vee L_5\}$
   $\neg L_2 \vee \neg L_4$ is the CDCL conflict clause, resolvent from the previous one and $\neg L_4 \vee L_5$

The CDSAT transition system: Resolve again

Resolve: $\langle \Gamma; E \uplus \{A\} \rangle \Longrightarrow \langle \Gamma; E \cup H \rangle$

- A is a justified assignment ${}_{H}\!\vdash A$
- Replace A by its justification H
- Provided H does not contain a first-order decision A' that stands out as its level is maximum in the conflict $(\mathsf{level}_\Gamma(A') = \mathsf{level}_\Gamma(E \uplus \{A\}))$
- Avoiding a Resolve–UndoClear–Decide loop
- And what if there is such an A'? UndoDecide rule

The CDSAT transition system: UndoDecide

UndoDecide: $\langle \Gamma; E \uplus \{L\} \rangle \Longrightarrow \Gamma^{\leq m-1}, {}_{?}\overline{L}$

- L is a Boolean justified assignment ${}_{H\vdash}L$ such that
  - H contains a first-order decision A'
  - $\mathsf{level}_\Gamma(A') = \mathsf{level}_\Gamma(L) = \mathsf{level}_\Gamma(E) = m$
- UndoDecide removes A' and decides $\overline{L}$
- A' is first-order and cannot be flipped
  (first-order decisions do not have complement)
- The Boolean L that depends on A' can be flipped

Example of UndoDecide

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

1. Decide: $x \leftarrow 0$ (level 1)
2. Deduce: $(x > 1) \leftarrow \mathit{false}$ (level 1)
   $(x < -1) \leftarrow \mathit{false}$ (level 1)
   $y < 0$ (level 1)
   $y > 0$ (level 1)
3. LRA-conflict: $\{y<0,\ y>0\}$
4. Resolve: $\{x > 1 \vee y < 0,\ x < -1 \vee y > 0,\ (x > 1) \leftarrow \mathit{false},$
   $(x < -1) \leftarrow \mathit{false}\}$
5. UndoDecide: $x > 1$ (level 1)

The CDSAT transition system: Learn

Learn: $\langle \Gamma; E \uplus H \rangle \Longrightarrow \Gamma^{\leq m}, {}_{E\vdash} F$

- H contains only Boolean assignments: H as $L_1 \wedge \ldots \wedge L_k$
- Since $E \uplus H \models \bot$, it is $E \models \overline{L_1} \vee \ldots \vee \overline{L_k}$
- Learned lemma: $F = \overline{L_1} \vee \ldots \vee \overline{L_k}$ (clausal form of H)
- Provided $F \notin \Gamma$, $\overline{F} \notin \Gamma$, $F \in \mathcal{B}$
- Choice of level where to backjump to:
  $\text{level}_\Gamma(E) \leq m < \text{level}_\Gamma(H)$

Recall the example

Γ_0 includes: $(\neg L_4 \vee L_5)$, $(\neg L_2 \vee \neg L_4 \vee \neg L_5)$ (level 0)

1. Decide: A_1 (level 1)
2. Decide: L_2 (level 2)
3. Decide: A_3 (level 3)
4. Decide: L_4 (level 4)
5. Deduce: L_5 with justification $\{\neg L_4 \vee L_5,\ L_4\}$ (level 4)
6. Conflict: $\{\neg L_2 \vee \neg L_4 \vee \neg L_5,\ L_2,\ L_4,\ L_5\}$
   $\neg L_2 \vee \neg L_4 \vee \neg L_5$ is the CDCL conflict clause
7. Resolve: $\{\neg L_2 \vee \neg L_4 \vee \neg L_5,\ L_2,\ L_4,\ \neg L_4 \vee L_5\}$
   $\neg L_2 \vee \neg L_4$ is the CDCL conflict clause, resolvent from the previous one and $\neg L_4 \vee L_5$

Examples of learning and backjumping by Learn

Conflict: $\{\neg L_2 \vee \neg L_4 \vee \neg L_5,\ L_2,\ L_4,\ \neg L_4 \vee L_5\}$

- Learn with $H = \{L_2, L_4\}$:
  learns the first assertion clause $\neg L_2 \vee \neg L_4$ with justification $\{\neg L_2 \vee \neg L_4 \vee \neg L_5,\ \neg L_4 \vee L_5\}$ (level 0)
- With destination level $m = 0$: restart from
  $(\neg L_4 \vee L_5),\ (\neg L_2 \vee \neg L_4 \vee \neg L_5),\ (\neg L_2 \vee \neg L_4)$
- With destination level $m = 2$:
  - Backjump to
    $(\neg L_4 \vee L_5),\ (\neg L_2 \vee \neg L_4 \vee \neg L_5),\ A_1,\ L_2,\ (\neg L_2 \vee \neg L_4)$
  - Deduce: $\neg L_4$ with justification $\{\neg L_2 \vee \neg L_4,\ L_2\}$

An example in a union of theories

$\Gamma_0 \;=\; f(select(store(a,i,v),j)) \simeq w,\;\; f(u) \simeq w-2,\;\; i \simeq j,\;\; u \simeq v$

- Decide: $u \leftarrow \mathfrak{c}$ (level 1)
- Decide: $v \leftarrow \mathfrak{c}$ (level 2)
- Decide: $select(store(a,i,v),j) \leftarrow \mathfrak{c}$ (level 3)
- Decide: $w \leftarrow 0$ (level 4)
- Decide: $f(select(store(a,i,v),j)) \leftarrow 0$ (level 5)
- Decide: $f(u) \leftarrow -2$ (level 6)
- Deduce: $u \simeq select(store(a,i,v),j)$ (level 3)
- Deduce: $f(u) \not\simeq f(select(store(a,i,v),j))$ (level 6)

  Both supported by equality inferences in EUF

An example in a union of theories (continued)

$\Gamma_0 \;=\; f(select(store(a,i,v),j)) \simeq w,\;\; f(u) \simeq w-2,\;\; i \simeq j,\;\; u \simeq v$

- Deduce: $u \simeq select(store(a,i,v),j)$ (level 3)
- Deduce: $f(u) \not\simeq f(select(store(a,i,v),j))$ (level 6)
- Conflict: the last two yield $\bot$ in $\mathcal{I}_{EUF}$
- Conflict:
  $\{u \simeq select(store(a,i,v),j), f(u) \not\simeq f(select(store(a,i,v),j))\}$
- Learn with destination level 3 backjumps and adds
  $f(u) \simeq f(select(store(a,i,v),j))$ with
  $u \simeq select(store(a,i,v),j)$ as justification

Proofs in CDSAT

- Proof objects in memory (checkable by proof checker)
  - The theory modules produce proofs
  - Proof-carrying CDSAT transition system
  - Proof reconstruction: from proof terms to proofs (e.g., resolution proofs)
- LCF style as in ITP (correct by construction)
  - Trusted kernel of primitives

Implementation

- MCSAT as add-on in DPLL(T)-based solvers Z3, CVC4, Yices
- MCSAT/CDSAT with the E-graph at the center
  [Bobot, Graham-Lengrand, Marre, Bury: SMT 2018]
- CDSAT in C++: prototype SMT/SMA solver Eos
  (by Giulio Mazzi at U. Verona)
  first solver built from the start based on CDSAT
  [MPB, Mazzi: SMT 2019]

Current and future work

- CDSAT search plans: both global and local issues
  - Heuristic strategies to make decisions, prioritize theory inferences, control lemma learning
  - Efficient techniques to detect the applicability of theory inference rules and the acceptability of assignments
- More theory modules (e.g., real arithmetic)
- Unions of non-disjoint theories
- Formulas with quantifiers

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: MPB, S. Graham-Lengrand, and N. Shankar

Thanks

Thank you!