Proof Generation in CDSAT[1]

Maria Paola Bonacina

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

Invited Keynote Speech
7th Int. Workshop on Proof eXchange for Theorem Proving (PxTP)

11 July 2021

(Subsumes the talk "Proof Reconstruction in Conflict-Driven Satisfiability"

Schloß Dagstuhl Seminar # 19371: "Deduction beyond satisfiability" September 2019)

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

Proofs in Automated Reasoning

- Validity query: `valid` / `invalid` / `don't know`
- Satisfiability query: `sat` / `unsat` / `don't know`
- Beyond ternary answers:
  - Proof of unsatisfiability or validity of the negation
  - Model: evidence of satisfiability or invalidity of the negation
  - Representation: formats, standardization
  - Manipulation: transformation, exchange, verification
  - Qualities: readability, useability, naturalness?

Proofs in Automated Theorem Proving (ATP)

- Derivation: $S_0 \vdash S_1 \vdash \ldots S_i \vdash S_{i+1} \vdash \ldots$
- S_i: set of clauses
- Refutation: $\exists\ k$ such that $\Box \in S_k$
- Proof reconstruction: extract proof from S_k
- Proof: ancestor-graph of $\Box$ (dag or tree)
- Inference rules determine shape of the dag (e.g.: resolution, superposition, hyperresolution, simplification)

Proofs in SAT Solving

- Derivation:
  $(S_0; M_0) \leadsto (S_1; M_1) \leadsto \ldots (S_i; M_i) \leadsto (S_{i+1}; M_{i+1}) \leadsto \ldots$
- M_i: trail of Boolean assignments
- Model found: $\exists\ k$ such that $M_k \models S_k$
- Conflict explanation: resolution btw conflict clause and justification (input or learned)
- Refutation: conflict clause is $\Box$
- Proof reconstruction: return resolution proof of $\Box$
- Encodings, simplification techniques

[Zhang, Malik: DATE 2003] [Cruz-Felipe et al.: CADE 2017]

Proofs in SMT Solving

- Justifications: also learned theory lemmas
- Theory procedures may or may not produce proofs
- Proof reconstruction: return resolution proof of $\Box$ with:
  - Theory lemmas as leaves
    No theory sub-proofs or black-box theory sub-proofs
  - Theory lemmas as roots of open-box theory sub-proofs

[Fontaine et al.: TACAS 2006], [Bjørner, de Moura: IWIL 2008]
[Katz et al.: FMCAD 2016], [Barbosa et al.: JAR 2020]

CDSAT (Conflict-Driven SATisfiability)

- SMT-problem: decide $\mathcal{T}$-satisfiability of a formula (set of clauses) for $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
- Disjoint theories and quantifier-free formulas
- CDSAT is a general framework for:
  - Conflict-Driven reasoning in the union $\mathcal{T}$
  - Orchestrating $\mathcal{T}_k$-inference systems $\mathcal{I}_k$ called theory modules
  - Treating propositional logic as one of the $\mathcal{T}_k$'s
  - Solving also SMA-problems
  - With proof generation assuming that the $\mathcal{I}_k$'s produce proofs

Conflict-driven reasoning

- Procedure to determine satisfiability of a formula
- Search for a model by building candidate models
- 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 it for a generic union $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$

Why CDSAT works with theory inference systems I

- CDCL (Conflict-Driven Clause Learning) procedure for SAT: conflict-driven reasoning for propositional logic
  [Marques Silva, Sakallah: ICCAD 1996, IEEE TOC 1999]
  [Davis, Putnam, Logeman, Loveland: JACM 1960, CACM 1962]
- Conflict-driven satisfiability procedures for other theories (e.g., fragments of arithmetic)

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 et al.: CP 2009] [McMillan et al.: CAV 2009]
[Cotton: FORMATS 2010] [Jovanović, de Moura: JAR 2013]
[Haller et al.: FMCAD 2012] [Jovanović, de Moura: IJCAR 2012]
[Brauße et al.: 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:
  $\{x{<}-y,\ -y{<}-2\} \vdash_{\mathsf{LRA}} x{<}-2$
  $\{x+y{<}0,\ -y+2{<}0\} \vdash_{\mathsf{LRA}} x+2{<}0$
  It generates new (non-input) atoms
- FM-resolution on demand to respond to conflicts
  [Korovin et al.: CP 2009] [McMillan et al.: CAV 2009]
  [Cotton: FORMATS 2010]

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

Why CDSAT works with theory inference systems II

- How to integrate CDCL and a conflict-driven satisfiability procedure for another theories?
- MCSAT (Model-Constructing SATisfiability)
  [de Moura, Jovanović: VMCAI 2013] [Jovanović et al.: FMCAD 2013]
- More general: CDSAT

Why CDSAT works with theory inference systems III

CDSAT:

- Generalizes MCSAT to generic unions of disjoint theories
- No need for theory procedures to be model-constructing
- Provides a new paradigm for reasoning in unions of theories

Key abstraction in CDSAT:

- From procedure to inference system
- Conflict-driven mechanism provided centrally by CDSAT

Why a new paradigm for theory combination

- Combination of theories by combination of procedures: Equality sharing method [Nelson, Oppen: ACM TOPLAS 1979] several variants
- Separation of the problem
- $\mathcal{T}_k$-sat procedures combined as black-boxes that
  - Build arrangement of shared variables by
  - Exchanging entailed (disjunctions of) equalities
- Combination lemmas with requirement on theories (e.g., stably infinite, polite)
- A $\mathcal{T}_k$-sat procedure can be conflict-driven inside the box
- The combination itself is not conflict-driven

Why treating propositional logic as one of the theories

DPLL($\mathcal{T}$) aka CDCL($\mathcal{T}$) with $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
[Nieuwenhuis et al.: 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)

CDSAT: a new paradigm for theory combination I

- CDCL loses centrality:
  Not the only conflict-driven procedure
- Resolution loses centrality:
  Not the only rule for conflict explanation
- Multiple theory modules access the trail, post assignments, perform inferences, explain $\mathcal{T}_k$-conflicts, deduce lemmas
- Combination of theories by cooperation of theory modules

CDSAT: a new paradigm for theory combination II

- Propositional logic as theory Bool
- No conflict-driven $\mathcal{T}_k$-sat procedure?
  Black-box theory module $L_1, \ldots, L_m \vdash_k \bot$
  invokes the $\mathcal{T}_k$-procedure to detect $\mathcal{T}_k$-unsat
- All theory modules contribute directly to the proof:
  Not necessarily resolution $+$ black-box $\mathcal{T}_k$-subproofs

CDSAT generalizes SMT to SMA

- SMA: Satisfiability Modulo theories and Assignments
- Generalize first-order assignments of conflict-driven theory procedures: from $x \leftarrow 3$ to $t \leftarrow \mathfrak{c}$
- Everything is assignment: $t \leftarrow \mathsf{true}$, $t \leftarrow \mathsf{false}$, $t \leftarrow \mathfrak{b}$
- Formulas as terms of sort prop (from proposition)
- Mixed assignments: $(x > 1) \leftarrow \mathsf{false}$, $x \leftarrow 3$, $select(a, j) \leftarrow 3$
- Difference btw $x \leftarrow 3$ and $(x \simeq 3) \leftarrow \mathsf{true}$
- Theory values made available by theory extensions

Plausible assignment

- An assignment is plausible if
  it does not contain $L \leftarrow \text{true}$ and $L \leftarrow \text{false}$
- 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 (SMT) problem: a plausible Boolean assignment
- Satisfiability Modulo theory and Assignment (SMA) problem: a plausible assignment with both Boolean and first-order assignments
- Relevant to:
  - Optimization problems [de Moura, Passmore: ADDCT 2013]
  - Parallelization (e.g., cube-and-conquer for SMT)

Theory view of an assignment

- The $\mathcal{T}_k$-view H_k of an assignment H:
  - 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
- Example: $\{x \leftarrow 3,\ y \leftarrow 3,\ z \leftarrow 4\} \subseteq H$:
  $\{x \simeq y,\ x \not\simeq z,\ y \not\simeq z\} \subseteq H_k$
  for all $\mathcal{T}_k$ having the sort of x, y, and z

Assignments and models: endorsement

- Model $\mathcal{M}$ endorses ($\models$) $u \leftarrow \mathfrak{c}$:
  $\mathcal{M}$ interprets u and $\mathfrak{c}$ as the same element
- $u \leftarrow \mathfrak{c}$, $t \leftarrow \mathfrak{c}$: $\mathcal{M}$ 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
- Sound: 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: $\mathrm{basis}_k(X)$ contains all terms that $\mathcal{I}_k$ can generate from set of terms X
- Complete: can expand any plausible $\mathcal{T}_k$-assignment not endorsed by a $\mathcal{T}_k$-model

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: $(t \simeq s){\leftarrow}\mathsf{true}$ and $\{t{\leftarrow}\mathfrak{c},\ s{\leftarrow}\mathfrak{c}\}$

Sample theory modules

- Theory module for Bool: abstraction of CDCL
- Theory module for EUF: abstraction of congruence closure
- Theory module for Arrays: inference rules building-in the axioms
- Theory module for LRA: abstraction of LRA-procedure with FM-resolution applied only to explain conflicts

Soundness, termination, and completeness of CDSAT

- Soundness: the theory modules are sound
- Termination:
  - Finite global basis $\mathcal{B}$ from which all new terms are drawn
  - It can be built from the local bases of the theory modules
- Completeness:
  - There is a leading theory: $\mathcal{T}_1$ has all the sorts in $\mathcal{T}$
  - Module $\mathcal{I}_1$ is complete for $\mathcal{T}_1$
  - Every other module $\mathcal{I}_k$ is leading-theory-complete: can expand any plausible $\mathcal{T}_k$-assignment not endorsed by a $\mathcal{T}_k$-model agreeing with a $\mathcal{T}_1$-model on cardinalities of shared sorts and equality of shared terms

Proofs in CDSAT

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

CDSAT trail: a sequence of assignments

- Each assignment is a decision $_{?}A$ or a justified assignment $_{H\vdash}A$
- 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 $H \vdash_k A$: proof term from $\mathcal{I}_k$
  - Input assignment ($H = \emptyset$): proof term $in(A)$
  - Due to conflict solving: proof term for the learned lemma
  - 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, LearnBackjump
- Apply to trail and conflict: $\langle \Gamma; H; c \rangle$
  - Conflict: $H \subseteq \Gamma$ is an unsatisfiable assignment
  - Conflict proof term c for $H \vdash \bot$
- Parameter: finite global basis $\mathcal{B}$

The CDSAT trail rules: 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 $\overline{L} \in \Gamma_k$
- 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

Examples: acceptability and relevance

- $L \in \Gamma$: both L and $\overline{L}$ are unacceptable for all modules
- $\{x \leftarrow 1,\ \overline{x < y}\} \subseteq \Gamma$:
  $y \leftarrow 2$ is unacceptable for LRA
  as $\{x \leftarrow 1,\ y \leftarrow 2\} \vdash_{\mathrm{LRA}} x < y$ by LRA-evaluation
- $\{f(u_1) \leftarrow \mathsf{red},\ u_2 \leftarrow \mathsf{yellow}\} \subseteq \Gamma$
  where f is a function from colors to colors:
  $u_1 \leftarrow \mathsf{yellow}$ is relevant to a theory of colors
  $u_1 \simeq u_2$ is relevant to EUF
  if EUF has the sort of colors

Forced decisions

- $u \leftarrow \mathfrak{c}$ is a forced decision if $\mathfrak{c}$ is the only acceptable value for u
- Examples:
  - $u \leftarrow \mathfrak{c}$ is forced for EUF if $\{u \simeq t,\ t \leftarrow \mathfrak{c}\} \subseteq \Gamma$
  - $u \leftarrow \mathfrak{c}$ is forced for LRA if $\{u \leq t,\ t \leq u,\ t \leftarrow \mathfrak{c}\} \subseteq \Gamma$
  - $y \leftarrow 2$ is forced for LRA if $\{x \leftarrow 1,\ (x + y) \leftarrow 3\} \subseteq \Gamma$

The CDSAT trail rules: 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)
- Covers $\mathcal{T}_k$-propagation and $\mathcal{T}_k$-conflict explanation
- $\mathcal{T}_k$-module produces $\mathcal{T}_k$-proof
  coerced into CDSAT deduction proof term

Example: Deduce as propagation

1. Decide: $u_2 \leftarrow \text{yellow}$ (level 1)
2. Decide: $f(u_1) \leftarrow \text{red}$ (level 2)
3. Decide: $u_1 \leftarrow \text{yellow}$ (level 3)
4. Decide: $f(u_2) \leftarrow \text{blue}$ (level 4)
5. Deduce: $u_1 \simeq u_2$ (level 3) /* equality inference */
6. Deduce: $f(u_1) \simeq f(u_2)$ (level 3) /* EUF-inference */

The Deduce steps are late propagations

Example: a conflict emerges

1. Decide: $u_2 \leftarrow$yellow (level 1)
2. Decide: $f(u_1) \leftarrow$red (level 2)
3. Decide: $u_1 \leftarrow$yellow (level 3)
4. Decide: $f(u_2) \leftarrow$blue (level 4)
5. Deduce: $u_1 \simeq u_2$ (level 3) /* late propagation */
6. Deduce: $f(u_1) \simeq f(u_2)$ (level 3) /* late propagation */
7. $\{f(u_1) \leftarrow \text{red},\ f(u_2) \leftarrow \text{blue}\} \vdash f(u_1) \not\simeq f(u_2)$: conflict
   by any theory module since it is an equality inference

The CDSAT trail rules: Fail

- $J \vdash_k L$, 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 d is a deduction proof term for $J \vdash L$
  $\mathit{cfl}(d, \overline{L})$ is a conflict proof term for $J \cup \{\overline{L}\} \vdash \bot$
- Conflict state: $\langle \Gamma; J \cup \{\overline{L}\}; \mathit{cfl}(d, \overline{L}) \rangle$
- If the conflict-state rules transform it into $\langle \Gamma; \emptyset; c \rangle$
  where empty conflict $\emptyset$ yields empty clause $\Box$:
  Fail: $\Gamma \longrightarrow \mathrm{unsat}(c)$ declares unsatisfiability returning the proof term for $\Box$

The CDSAT trail rules: 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 d is a deduction proof term for $J \vdash L$
  $cfl(d, \overline{L})$ is a conflict proof term for $J \cup \{\overline{L}\} \vdash \bot$
- Conflict state: $\langle \Gamma; J \cup \{\overline{L}\}; cfl(d, \overline{L}) \rangle$
- If the conflict-state rules transform it into Γ':
  ConflictSolve: $\Gamma \longrightarrow \Gamma'$ as the conflict is solved

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 $\langle \Gamma; E; c \rangle$
- Explanation of conflict E by replacing justified assignments in E with their justifications: Resolve transition rule

The CDSAT conflict state rules: Resolve

Resolve: $\langle \Gamma; E \uplus \{A\}; c\rangle \Longrightarrow \langle \Gamma; E \cup H; res(d, A.c)\rangle$

- A is a justified assignment ${}_{H\vdash} A$
- Replace A by its justification H
- d: deduction proof term for $H \vdash A$
  c: conflict proof term for $E \uplus \{A\} \vdash \bot$
  $res(d, A.c)$: conflict proof term for $E \cup H \vdash \bot$
  - A can be a Boolean or a first-order assignment
  - If A is first-order, it comes from the input ($H = \emptyset$ and $d = in(A)$): Resolve removes it from the conflict (not from the trail)

The CDSAT conflict state rules: UndoClear

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

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

- A is a first-order decision of level $m > \text{level}_\Gamma(E)$
- Removes A and all assignments of level $\geq m$
- $\Gamma^{\leq m-1}$: Γ restricted to its elements of level at most $m{-}1$
- $\Gamma^{\leq m-1}$ is new because it must contain a late propagation
- No role in proof generation: first-order decisions are for models, not proofs
- Only input first-order assignments may appear in proofs

Example: UndoClear

1. Decide: $u_2 \leftarrow \mathsf{yellow}$ (level 1)
2. Decide: $f(u_1) \leftarrow \mathsf{red}$ (level 2)
3. Decide: $u_1 \leftarrow \mathsf{yellow}$ (level 3)
4. Decide: $f(u_2) \leftarrow \mathsf{blue}$ (level 4)
5. Deduce: $u_1 \simeq u_2$ (level 3) /* late propagation */
6. Deduce: $f(u_1) \simeq f(u_2)$ (level 3) /* late propagation */
7. Conflict: $\{f(u_1) \simeq f(u_2),\ f(u_1) \leftarrow \mathsf{red},\ f(u_2) \leftarrow \mathsf{blue}\}$
8. UndoClear: undoes $f(u_2) \leftarrow \mathsf{blue}$
9. Decide: $f(u_2) \leftarrow \mathsf{red}$ (level 4) /* only acceptable value */

The CDSAT conflict state rules: Resolve again

Resolve: $\langle \Gamma; E \uplus \{A\}; c\rangle \Longrightarrow \langle \Gamma; E \cup H; res(d, A.c)\rangle$

- 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 ($\mathrm{level}_\Gamma(A') = \mathrm{level}_\Gamma(E \uplus \{A\})$)
- Avoiding a Resolve–UndoClear–Decide loop
- And what if there is such an A'? UndoDecide rule

The CDSAT conflict state rules: UndoDecide

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

- L is a Boolean justified assignment ${}_{H\vdash}L$ such that
  - H contains a first-order decision A'
  - $\mathrm{level}_\Gamma(A') = \mathrm{level}_\Gamma(L) = \mathrm{level}_\Gamma(E) = m$
- 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
- No role in proof generation like for UndoClear

Example of UndoDecide

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

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

The CDSAT conflict state rules: LearnBackjump

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

- H contains only Boolean assignments: H as $L_1 \wedge \ldots \wedge L_k$
- Since $H_0 \cup (E \uplus H) \models \bot$, it is $H_0 \cup E \models \overline{L_1} \vee \ldots \vee \overline{L_k}$ for H_0 the input
- Learned lemma: $F = \overline{L_1} \vee \ldots \vee \overline{L_k}$ $\quad (F \notin \Gamma, \overline{F} \notin \Gamma, F \in \mathcal{B})$
- Choice of level where to backjump to: $\mathrm{level}_\Gamma(E) \leq m < \mathrm{level}_\Gamma(H)$
- If it picks $\mathrm{level}_\Gamma(E) = 0$: learn and restart
- If c is a conflict proof term for $E \uplus H \vdash \bot$ $\mathit{lem}(H.c)$ is a deduction proof term for $E \vdash F$

Example of Resolve towards LearnBackjump

Γ 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 next CDCL conflict clause (resolvent of previous one and CDCL justification $\neg L_4 \vee L_5$) and first assertion clause

Examples of learning and backjumping by LearnBackjump

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

- LearnBackjump 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 $\{\neg L_2 \vee \neg L_4,\ L_2\}$

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 (e.g., bridging functions)
- Formulas with quantifiers: CDSAT(SGGS)

References

- Satisfiability modulo theories and assignments.
  Proc. of CADE-26, LNAI 10395, 42–59, Springer, August 2017.
- Proofs in conflict-driven theory combination.
  Proc. of the 7th ACM SIGPLAN Int. Conf. on Certified Programs and Proofs (CPP), ACM Press, 186–200, January 2018.
- Conflict-driven satisfiability for theory combination: transition system and completeness.
  Journal of Automated Reasoning, 64(3):579–609, March 2020.
- Conflict-driven satisfiability for theory combination: modules, lemmas, and proofs.
  Journal article, 54 pages, submitted February 2020.

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

Thanks

Thank you!