# The CDSAT Paradigm for Theory Combination in SMT

(Based on joint work with S. Graham-Lengrand and N. Shankar)

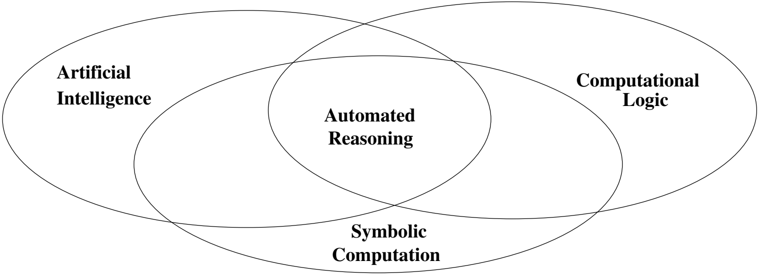
Maria Paola Bonacina

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

Invited Tutorial
21st Int. Conf. on Computability in Europe (CiE)
Lisbon, Portugal, EU

15 and 17 July 2025

# Automated reasoning (AR) in computer science




- AR: make computers reason ... in their own way
- AR $\subset$ AI: logic-based, symbolic AI
- AR $\subset$ SC: logico-deductive, symbolic reasoning
- AR $\subset$ CL: computing to perform logical reasoning

# Applications of automated reasoning

- Embedded in tools for analysis, verification, synthesis, and optimization of software
  - Objectives, e.g.:
    - Correct-by-construction software
    - Provable privacy
    - Verification of distributed systems, distributed protocols, randomized algorithms
  - Complementary to other techniques, e.g.:
    Model checking, static analysis, machine learning (ML)
- Applied in deductive knowledge bases, computer mathematics, mathematical libraries, education
- Integration of AR and ML (e.g., generative AI) towards a better AI ?

## What automated reasoning does

- Design and implementation of computer programs that reason
- To solve problems formulated as
- Validity or satisfiability queries in a logic or a theory
- Using inference and search

In this tutorial:

- Satisfiability is decidable
- Input problems are quantifier-free and in clausal form
- Conflict-driven reasoning procedures used in SMT solvers

# Example problems: clauses involving theory symbols

- Propositional logic (the Boolean theory):
  $\{\overline{A} \vee B,\ \overline{A} \vee C \vee E,\ \overline{B} \vee D,\ \overline{C} \vee \overline{D}, A \vee \overline{B} \vee E,\ B \vee \overline{C},\ F \vee \overline{E}\}$
- Linear integer arithmetic (LIA) and Equality with Uninterpreted Functions (EUF or UF):
  $\{x \leq y,\ y \leq (x + g(x)),\ P(h(x) - h(y)),\ \neg P(0),\ g(x) \simeq 0\}$
- Bool and linear rational arithmetic (LRA):
  $\{x < y,\ x < z,\ (y < w) \vee (z < w),\ w < x\}$
- Bool, LRA, and Arrays (Arr):
  $(i \not\simeq j) \vee (\mathsf{select}(\mathsf{store}(a, i, v), j) < \mathsf{select}(a, j))$
  $(\mathsf{select}(a, j) - \mathsf{select}(a, k)) \simeq 0$
  $(\mathsf{select}(\mathsf{store}(a, i, v), j) \not< \mathsf{select}(a, j)) \vee$
  $(\mathsf{select}(a, j) + \mathsf{select}(a, k) \simeq v)$

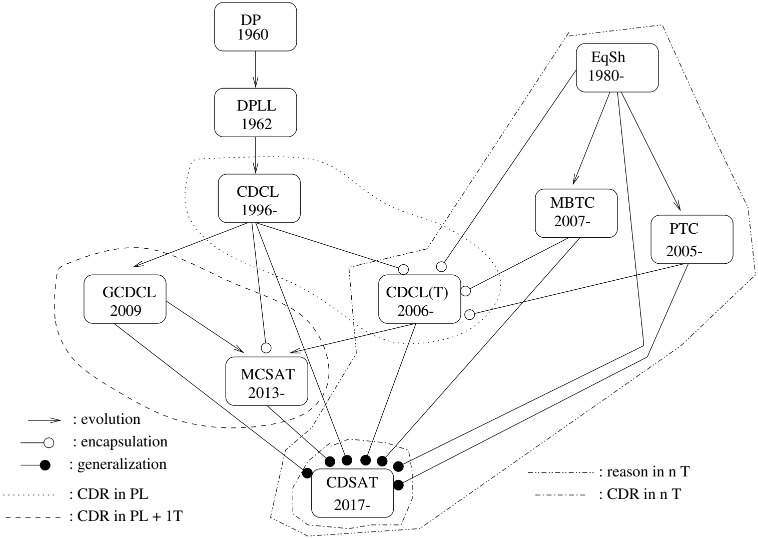
# Conflict-driven reasoning (CDR) paradigm

- Decision procedure for satisfiability of a set of clauses
- Search for a model
- Perform inferences to solve conflicts or prove unsatisfiability
- Search and inferences guide each other:
  - Search focuses inferences on conflicts
  - Inferences allow search to escape dead-end's

# Conflict-driven reasoning (CDR) paradigm

- Search for a model:
  - Decide assignments of values to terms
  - Propagate consequences of assignments (inexpensive inferences)
  - Conflict: contradiction
- Either reach unsatisfiability or solve conflict:
  - Explain conflict by expensive inferences (steps towards a possible refutation)
  - Learn generated lemma which excludes current assignment and avoids hitting same conflict
  - Solve conflict by amending assignment to satisfy lemma

# The big picture



# CDSAT: most general conflict-driven reasoning procedure

- SMT (Satisfiability Modulo Theory): decide satisfiability in theory $\mathcal{T}$
- $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$: predicate-sharing theories
  Disjoint if $\simeq$ is the only shared symbol
- SMA (Satisfiability Modulo Theory and Assignment): input includes initial assignment
  - Boolean assignment: $L \leftarrow$true (Boolean value)
  - First-order assignment: $x \leftarrow 3$ (non-Boolean value)
  - Relevant for parallelization, optimization as satisfiability, quantified satisfiability (QSMA)
- Answer sat if there exists satisfying assignment including initial one, unsat otherwise

# Assignments take center stage

- Assignments of values to terms:
  $(x > 1) \leftarrow \mathrm{false}$, $((x > 1) \vee (y < 0)) \leftarrow \mathrm{true}$,
  $(\mathrm{store}(a, i, v) \simeq b) \leftarrow \mathrm{true}$, $y \leftarrow \sqrt{2}$, $\mathrm{select}(a, j) \leftarrow 3$
- Term and value have the same sort
- Formulas are Boolean terms (sort prop)
- Plausible assignment: does not contain $L \leftarrow \mathrm{true}$ and $L \leftarrow \mathrm{false}$
- Terms and values are kept separate:
  term only on the left, value only on the right of an assignment
- $\mathrm{select}(a, j) \leftarrow 3$ cannot be replaced by $\mathit{select}(a, j) \simeq 3$:
  a value is not a term, is not in the signature
- What are values?

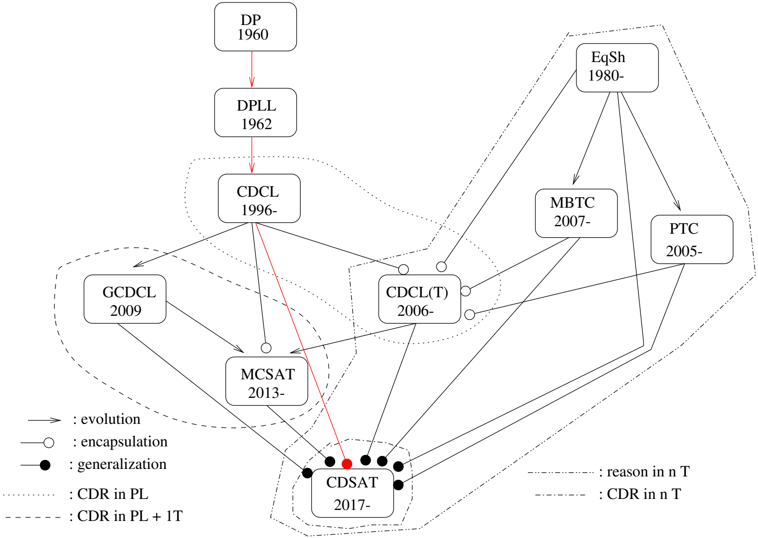
# Theory extensions to define values

- From theory $\mathcal{T}_k$ to theory extension $\mathcal{T}_k^+$:
  - Add new constant symbols (and possibly axioms)
  - E.g.: add a constant symbol for every number (integers, rationals, algebraic reals)
    $\sqrt{2}$ is a constant symbol interpreted as $\sqrt{2}$
  - All $\mathcal{T}_k^+$'s add true and false (all $\mathcal{T}_k$'s have sort prop)
  - Trivial if it adds only true and false
- Values in assignments are these constant symbols: $\mathcal{T}_k$-values
- $\mathcal{T}_k$-assignment: assigns $\mathcal{T}_k$-values
- Conservative theory extension: $\mathcal{T}_k^+$-unsatisfiable implies $\mathcal{T}_k$-unsatisfiable

# CDSAT: most general conflict-driven reasoning procedure

- Transition system: transition rules (e.g., Decide, Deduce)
- Coordinate theory modules ($\mathcal{T}_k$-inference systems)
- Each module offers decisions, deductions (propagations, explanations); with a finite local basis
- Finite global basis from the local ones for termination
- The modules collaborate as peers
  on a shared trail $\Gamma$ containing the current assignment
- Conflict-driven control for all the theories in the union
- Sound, complete, terminating under suitable hypotheses

# The big picture: propositional reasoning



# Propositional satisfiability

- DP [Davis, Putnam: JACM 1960]:
  - Resolution ($C \vee L$ and $D \vee \overline{L}$ resolve to generate $C \vee D$)
  - Subsumption ($L$ subsumes $C \vee L$, and $D$ subsumes $D \vee \overline{L}$)
- DPLL [Davis, Putnam, Logeman, Loveland: CACM 1962]:
  - Resolution replaced by splitting on $L$ and $\overline{L}$
  - Unit propagation: unit subsumption + unit resolution: if $L$ on $\Gamma$, delete $C \vee L$, and replace $D \vee \overline{L}$ with $D$
  - Conflict (e.g., $\{P, \overline{P}\}$): backtrack last guess (e.g., from $L$ to $\overline{L}$)
  - Backtracking search over partial models represented as a trail $\Gamma$ of Boolean assignments (stack)

# Conflict-driven propositional satisfiability

CDCL (Conflict-Driven Clause Learning)
[Marques Silva, Sakallah: ICCAD 1996, IEEE TOC 1999]:

- Decision replaces splitting:
  add $L$ to trail $\Gamma$ provided $L \notin \Gamma$ and $\overline{L} \notin \Gamma$
- Conflict-driven backjumping replaces backtracking
- Every decision opens new level on trail $\Gamma$ (stack)
- Unit propagation detects
  - Implied literal $L$ with justification $C = L_1 \vee \ldots \vee L_k \vee L$ if $\overline{L}_i \in \Gamma$ $(1 \leq i \leq k)$
  - Conflict clause $Q_1 \vee \ldots \vee Q_n$ if $\overline{Q}_i \in \Gamma$ $(1 \leq i \leq n)$

# Conflict-driven propositional satisfiability

- Apply resolution only to explain conflict
- Learn lemma (resolvent)
- Backjump away from conflict
  to a state that satisfies the lemma
- First assertion clause heuristic:
  - Resolve until $C = L_1 \vee \ldots \vee L_k \vee L$ (first assertion clause) where only $L$ is false on current level
  - Learn $C$
  - Backjump to the smallest level such that $\overline{L}_i \in \Gamma$ $(1 \leq i \leq k)$ and $L$ undefined
  - $L$ is implied with justification $C$

CDSAT reduces to CDCL if Bool is the only theory in the union

# CDSAT generalizes CDCL: basic CDSAT

- Trail $\Gamma$ is a sequence of assignments:
  clause $C$ abbreviates $C{\leftarrow}\mathsf{true}$
- Transition rule Decide: ${}_{?}L$
  acceptable if $L \notin \Gamma$ and $\overline{L} \notin \Gamma$ (more later for first-order decisions)
- Transition rule Deduce adds justified assignment ${}_{J\vdash}L$
  with justification $J$ if $J \vdash_k L$ for some $\mathcal{T}_k$
  $\mathrm{level}_\Gamma({}_{J\vdash}L) = \mathrm{level}_\Gamma(J)$ and $\mathrm{level}_\Gamma(J) = max\{\mathrm{level}_\Gamma(A) | A \in J\}$
  Deduce covers unit propagation: implied literal: ${}_{J\vdash}L$
  $J \vdash_{\mathrm{Bool}} L \qquad J = \{C \vee L, \neg C\}$
- Trail not a stack: ${}_{J\vdash}L$ may be added after assignments of higher level as multiple modules share $\Gamma$: late propagation
- Input assignments on $\Gamma$ at level 0 as justified assignments with empty justification: ${}_{\emptyset\vdash}C$ (two kinds of assignment and not three)

# CDSAT generalizes CDCL: basic CDSAT

- Conflict: $J \subseteq \Gamma$, $J \vdash_k L$ for some $\mathcal{T}_k$, and $\overline{L} \in \Gamma$
  unsatisfiable assignment $E = J \cup \{\overline{L}\}$
- Conflict state: $\langle \Gamma; E \rangle$, $E \subseteq \Gamma$
- Transition rule Resolve explains $E$ by replacing ${}_{J\vdash}L$ in $E$ with $J$
- Given conflict $E = J \uplus H$ where $H = \{\overline{L}_1, \ldots, \overline{L}_k\}$
  transition rule LearnBackjump
  - Learns ${}_{J\vdash}C$ where $C = L_1 \vee \ldots \vee L_k$:
    $J$ entails $C$ since $J \uplus H$ is unsatisfiable
  - Backjumps to a level $m$ such that
    $m < \mathrm{level}_\Gamma(H)$ (quit conflict) and
    $m \geq \mathrm{level}_\Gamma(J)$ so that ${}_{J\vdash}C$ can be added to $\Gamma$

# First assertion clause heuristic in CDSAT

- Apply Resolve until conflict $E$ contains only one literal $\overline{L}$ whose level $m$ is max in $E$
- Generalization: $m$ is not necessarily the current level
- Apply LearnBackjump to conflict $E = J \uplus H$ where $H = \{\overline{L}\} \uplus H'$ and $H' = \{\overline{L}_1, \ldots, \overline{L}_k\}$
- Learn ${}_{J\vdash}C$ where $C = L_1 \vee \ldots \vee L_k \vee L$
- Backjump to level $n = \mathrm{level}_\Gamma(J \uplus H')$:
  $n < \mathrm{level}_\Gamma(H)$ as $\mathrm{level}_\Gamma(H) = \mathrm{level}_\Gamma(\overline{L})$ which is max in $E$
  $n \geq \mathrm{level}_\Gamma(J)$ as $J \uplus H'$ is superset of $J$
- Apply Deduce to add ${}_{\{C\}\uplus H'\vdash}L$ supported by $\{C\} \uplus H' \vdash_{\mathrm{Bool}} L$

LearnBackjump may follow other heuristics (e.g., learn and restart)

# CDSAT module for theory Bool

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

# Example where CDSAT emulates CDCL

1. $S = \{\overline{A} \vee B,\ \overline{A} \vee C \vee E,\ \overline{B} \vee D,\ \overline{C} \vee \overline{D}, A \vee \overline{B} \vee E,\ B \vee \overline{C},\ F \vee \overline{E}\}$ subset of input
2. Decide adds ${}_{?}\overline{F}$ to trail $\Gamma$ opening level $n$
3. Deduce adds ${}_{J\vdash}\overline{E}$ with $J = \{F \vee \overline{E}, {}_{?}\overline{F}\}$ to level $n$ since $\{F \vee \overline{E}, {}_{?}\overline{F}\} \vdash_{\text{Bool}} \overline{E}$
4. Two more Decide create levels $n+1$ and $n+2$
5. Another Decide adds ${}_{?}A$ opening level $n+3$
6. Deduce adds to level $n+3$
   ${}_{H\vdash}B$ with $H = \{\overline{A} \vee B, {}_{?}A\}$
   ${}_{I\vdash}C$ with $I = \{\overline{A} \vee C \vee E, {}_{J\vdash}\overline{E}, {}_{?}A\}$
   ${}_{K\vdash}D$ with $K = \{\overline{B} \vee D, {}_{H\vdash}B\}$

# Example where CDSAT emulates CDCL

7. $\{\overline{C} \vee \overline{D},\ {}_{I\vdash} C\} \vdash_{\text{Bool}} \overline{D}$ but ${}_{K\vdash} D \in \Gamma$
   Conflict: $E_0 = \{\overline{C} \vee \overline{D},\ {}_{I\vdash} C,\ {}_{K\vdash} D\}$
   /* $\overline{C} \vee \overline{D}$ is conflict clause, not assertion clause */
8. $E_0$ contains literals ${}_{I\vdash} C$ and ${}_{K\vdash} D$ of max level $(n+3)$
   Resolve: $E_1 = \{\overline{C} \vee \overline{D},\ {}_{I\vdash} C,\ \overline{B} \vee D,\ {}_{H\vdash} B\}$
   /* $\overline{C} \vee \overline{D}$ and $\overline{B} \vee D$ yield $\overline{B} \vee \overline{C}$ (not assertion clause) */
9. $E_1$ contains literals ${}_{I\vdash} C$ and ${}_{H\vdash} B$ of max level $(n+3)$
   Resolve: $E_2 = \{\overline{C} \vee \overline{D},\ \overline{A} \vee C \vee E,\ {}_{J\vdash} \overline{E},\ {}_{?} A,\ \overline{B} \vee D,\ {}_{H\vdash} B\}$
   /* $\overline{B} \vee \overline{C}$ and $\overline{A} \vee C \vee E$ yield $\overline{B} \vee \overline{A} \vee E$ (not assertion clause) */
10. $E_2$ contains literals ${}_{?} A$ and ${}_{H\vdash} B$ of max level $(n+3)$
    Resolve: $E_3 = \{\overline{C} \vee \overline{D},\ \overline{A} \vee C \vee E,\ {}_{J\vdash} \overline{E},\ {}_{?} A,\ \overline{B} \vee D,\ \overline{A} \vee B\}$
    /* $\overline{B} \vee \overline{A} \vee E$ and $\overline{A} \vee B$ yield $\overline{A} \vee E$ (assertion clause) */

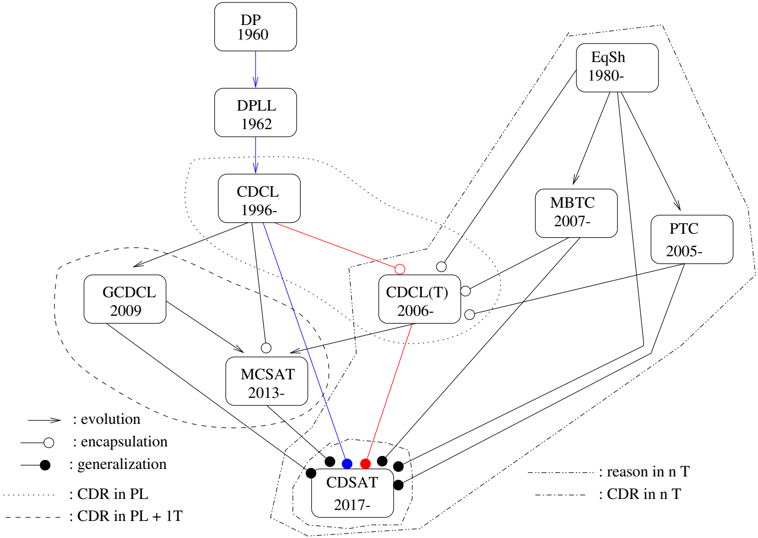
# Example where CDSAT emulates CDCL

$E_3 = \{\overline{C} \vee \overline{D},\ \overline{A} \vee C \vee E,\ {}_{J\vdash}\overline{E},\ {}_{?}A,\ \overline{B} \vee D,\ \overline{A} \vee B\}$
${}_{?}A$ has level $n+3$ (max), ${}_{J\vdash}\overline{E}$ has level $n$, and the rest has level 0

11. LearnBackjump jumps back to level $n$
    adds ${}_{G\vdash}(\overline{A} \vee E)$ with $G = \{\overline{C} \vee \overline{D},\ \overline{A} \vee C \vee E,\ \overline{B} \vee D,\ \overline{A} \vee B\}$
12. Deduce adds ${}_{M\vdash}\overline{A}$ with $M = \{{}_{G\vdash}(\overline{A} \vee E),\ {}_{J\vdash}\overline{E}\}$
    since $\{{}_{G\vdash}(\overline{A} \vee E), {}_{J\vdash}\overline{E}\} \vdash_{\mathrm{Bool}} \overline{A}$
13. Deduce adds ${}_{N\vdash}\overline{B}$ with $N = \{A \vee \overline{B} \vee E,\ {}_{M\vdash}\overline{A},\ {}_{J\vdash}\overline{E}\}$
14. Deduce adds ${}_{P\vdash}\overline{C}$ with $P = \{B \vee \overline{C},\ {}_{N\vdash}\overline{B}\}$

$\Gamma$ contains $\{\overline{E},\ \overline{A},\ \overline{B},\ \overline{C}\}$ model of $S$

# The big picture: from SAT to SMT



# CDCL($\mathcal{T}$): from SAT to SMT

DPLL($\mathcal{T}$) later renamed CDCL($\mathcal{T}$) for $\mathcal{T}$ a single theory
[Nieuwenhuis, Oliveras, Tinelli: JACM 2006]

- CDCL + decision procedure for $\mathcal{T}$-satisfiability of set of $\mathcal{T}$-literals
- CDCL works on propositional abstraction: $\mathcal{T}$-atoms replaced by propositional variables
- Let $\{L_1, \ldots, L_n\} \subseteq \Gamma$ and $C = \overline{L}_1 \vee \ldots \vee \overline{L}_n$
  $\mathcal{T}$-sat procedure contributes only:
  - $\mathcal{T}$-conflict detection: if $\{L_1, \ldots, L_n\}$ is $\mathcal{T}$-unsat
    $C$ is conflict clause
  - $\mathcal{T}$-propagation: if $\{L_1, \ldots, L_n\}$ $\mathcal{T}$-entails $L$
    add $L$ to $\Gamma$ with justification $C \vee L$
    $L$ must be an input literal (i.e., not new)

# CDCL($\mathcal{T}$): from SAT to SMT

- $\mathcal{T}$-sat procedure integrated as a black-box
- That only raises a flag if it detects an inconsistency in the propositional model that CDCL is building ignoring the theory:
  - $\mathcal{T}$-conflict: $\{L_1, \ldots, L_n\}$ is $\mathcal{T}$-unsat hence $\overline{L}_1 \vee \ldots \vee \overline{L}_n$ is $\mathcal{T}$-valid
  - $\mathcal{T}$-propagation: $\{L_1, \ldots, L_n, \overline{L}\}$ is $\mathcal{T}$-unsat hence $\overline{L}_1 \vee \ldots \vee \overline{L}_n \vee L$ is $\mathcal{T}$-valid

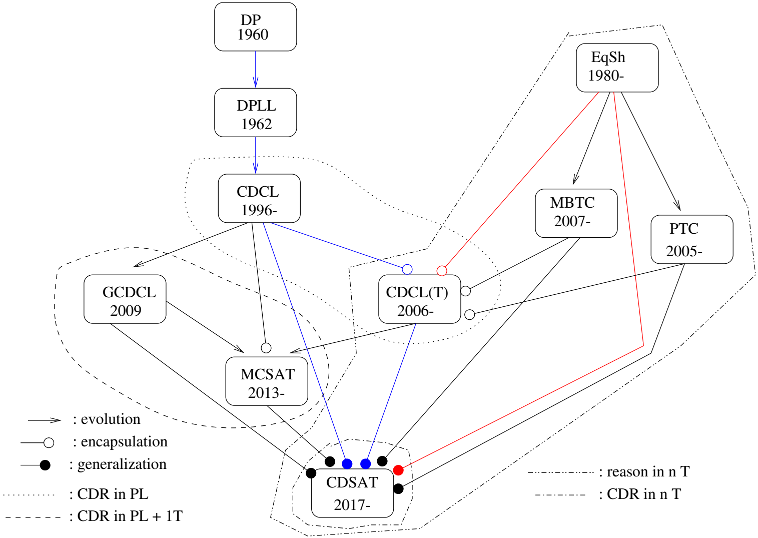
  Never deduce anything that excludes a $\mathcal{T}$-model but is not $\mathcal{T}$-valid
- Model search, trail, conflict explanation, conflict-driven reasoning remain propositional

# CDSAT generalizes CDCL($\mathcal{T}$)

- Consider a theory union whose members are Bool and $\mathcal{T}$
- Theory modules:
  - Bool-module
  - black-box $\mathcal{T}$-module:
    - Only one inference rule: $L_1, \ldots, L_m \vdash \bot$
    - That invokes the $\mathcal{T}$-procedure to detect $\mathcal{T}$-unsat of a set of literals

CDSAT can use a black-box $\mathcal{T}$-module
whenever a theory $\mathcal{T}$ is not involved in conflict-driven reasoning

# The big picture: theory combination



# Classical approach to theory combination: equality sharing

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

- $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$: disjoint theories (share $\simeq$ and sorts)
- Decision procedure for $\mathcal{T}_k$-satisfiability of set of $\mathcal{T}_k$-literals
- Stably infinite: $\mathcal{T}_k$-model with infinite cardinality
- Get decision procedure for $\mathcal{T}$-satisfiability of set of $\mathcal{T}$-literals
- Combination of decision procedures as black-boxes
- By disjointness, agreement is needed on:
  - Cardinalities of shared sorts: by stable infiniteness
  - Equalities between shared terms: needs work

# Equality sharing: separation

- Input set $S$: $\mathcal{T}$-literals mix symbols from the $\mathcal{T}_k$'s signatures
- Separate $S$ into sets $S_k$ of $\mathcal{T}_k$-literals sharing only $\simeq$ and variables

Example: $S$ contains $f(2, y) \simeq f(x, y)$

- EUF ($f \in \Sigma_{\mathrm{EUF}}$) and LIA ($2 \in \Sigma_{\mathrm{LIA}}$)
- Shared sort: Z; $\quad \simeq$ is $\simeq_{\mathrm{Z}}$; $\quad f : \mathrm{Z} \times \mathrm{Z} \rightarrow \mathrm{Z}$
- EUF: 2 is a variable
- LIA: $f(2, y)$ and $f(x, y)$ are variables
- $S_{\mathrm{EUF}} = \{w_1 \simeq f(w_2, y),\ w_3 \simeq f(x, y),\ w_1 \simeq w_3\}$
- $S_{\mathrm{LIA}} = \{w_2 \simeq 2,\ w_1 \simeq w_3\}$
- Shared variables: $\mathcal{V}_{\mathrm{sh}}(S) = \{w_1,\ w_2,\ w_3\}$

# How CDSAT handles separation

- Input set $S$: $\mathcal{T}$-literals mix symbols from the $\mathcal{T}_k$'s signatures
- Each $\mathcal{T}_k$ treats as a variable a term whose top symbol is foreign

Example: $S$ contains $f(2, y) \simeq f(x, y)$
(i.e., $(f(2, y) \simeq f(x, y)) \leftarrow \mathsf{true}$)

- EUF ($f \in \Sigma_{\mathrm{EUF}}$) and LIA ($2 \in \Sigma_{\mathrm{LIA}}$)
- Shared sort: $\mathsf{Z}$; $\simeq$ is $\simeq_{\mathsf{Z}}$; $f\colon \mathsf{Z} \times \mathsf{Z} \rightarrow \mathsf{Z}$
- EUF: 2 is foreign hence a variable
- LIA: $f$ is foreign hence $f(2, y)$ and $f(x, y)$ are variables
- Shared terms:
  $\mathcal{V}_{\mathrm{sh}}(S) = \{f(2, y) \simeq f(x, y),\ f(2, y),\ 2,\ f(x, y)\}$

# Equality sharing: the reduction

- Reduce the $\mathcal{T}$-sat problem to $\mathcal{T}_k$-sat problems
- $S$ is $\mathcal{T}$-sat iff $\bigcup_{k=1}^{n} S_k$ is $\mathcal{T}$-sat
- Arrangement $\alpha$: represents a partition of $\mathcal{V}_{\mathrm{sh}}(S)$
- $\alpha$: conjunction that contains
  - $u \simeq v$ if $u$ and $v$ in the same class of the partition
  - $u \not\simeq v$ otherwise
- Combination theorem:
  $\bigcup_{k=1}^{n} S_k$ is $\mathcal{T}$-sat iff $\exists\ \alpha$ s.t. $S_k \wedge \alpha$ is $\mathcal{T}_k$-sat $(1 \leq k \leq n)$

# Equality sharing: build arrangement (convex theories)

- $\mathcal{E}_0 = \emptyset$
- $\mathcal{E}_i = \mathcal{E}_{i-1} \cup \{u \simeq v\}$ if a $\mathcal{T}_k$-sat procedure deduces $u \simeq v$ from $S_k \cup \mathcal{E}_{i-1}$
- If a $\mathcal{T}_k$-sat procedure deduces $\perp$ from $S_k \cup \mathcal{E}_i$ for some $i$: return unsat ($S$ is $\mathcal{T}$-unsat)
- Otherwise, let $\alpha = \mathcal{E}_q$ such that $\mathcal{E}_q = \mathcal{E}_{q-1}$ (no more equalities) and return sat ($S$ is $\mathcal{T}$-sat)

Complete for convex theories:
$\mathcal{T}_k$ is convex if
$\mathcal{T}_k \models H \supset \bigvee_{i=1}^{n} u_i \simeq v_i$ implies $\exists\, j,\ 1 \leq j \leq n,\ \mathcal{T}_k \models H \supset u_j \simeq v_j$
$H$: a conjunction of $\mathcal{T}_k$-literals

# Equality sharing: build arrangement (non-convex theories)

- $\mathcal{T}_k$ not convex: $\mathcal{T}_k$-procedure deduces $\bigvee_{j=1}^{m} u_j \simeq v_j$
- $\mathcal{T}$-procedure calls itself recursively on each subproblem obtained by adding $u_j \simeq v_j$ to current $\mathcal{E}_i$
- In practice: CDCL($\mathcal{T}$) where $\mathcal{T}$-procedure is equality sharing combination [Barrett, Nieuwenhuis, Oliveras, Tinelli: LPAR 2006] [Krstić, Amit Goel: FroCoS 2007]
  - $\mathcal{T}$-procedure sends (propositional abstraction of) $\bigvee_{j=1}^{m} u_j \simeq v_j$ to CDCL
  - Reasoning about disjunction is entrusted to CDCL
  - Case $u_j \simeq v_j$ is considered when CDCL puts it on the trail
  - Sole new (i.e., non-input) literals in CDCL($\mathcal{T}$): (propositional abstractions of) equalities between shared variables

# Equality sharing is not conflict-driven

- Combining theories by combining procedures
- $\mathcal{T}_k$-procedures combined as black-boxes
- Generation of (disjunctions of) equalities resembles saturation (can be emulated by superposition)
- In CDCL($\mathcal{T}$) where $\mathcal{T}$-procedure is equality sharing combination, model search, trail, conflict explanation, conflict-driven reasoning remain propositional

In order to see how CDSAT emulates Equality Sharing, let's learn more about theory modules in CDSAT

# CDSAT modules

Theory modules $\mathcal{I}_1, \ldots, \mathcal{I}_n$ for theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$

- Theory module $\mathcal{I}_k$ for theory $\mathcal{T}_k$ is a set of inference rules $J \vdash_k L$ where
  - $J$ is a $\mathcal{T}_k$-assignment: may contain first-order assignments
  - $L$ is a singleton Boolean assignment
  - If a first-order assignment to $x$ follows from the trail it can be added as a decision (forced decision)
- Local basis: $\mathrm{basis}_k(X)$ contains all terms that $\mathcal{I}_k$ can generate from set of terms $X$

# CDSAT modules: equality inferences

All CDSAT 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, two ways to make $t \simeq s$ true:
$(t \simeq s){\leftarrow}\mathsf{true}$ and $t{\leftarrow}\mathfrak{c},\ s{\leftarrow}\mathfrak{c}$

# CDSAT generalizes equality sharing

- Each $\mathcal{T}_k$ module can place its inferences $J \vdash_k L$ as justified assignments $_{J\vdash} L$ on the shared trail by Deduce transitions (Deduce covers $\mathcal{T}_k$-propagation)
  - Equality inferences: transitivity steps and equalities from first-order assignments contribute to build an arrangement
  - Theory specific inference rules can deduce (disjunctions of) equalities
- The $\mathcal{T}_k$ modules cooperate to build an arrangement publicly on the shared trail
- Disjunctions are handled by the Bool-module by decision and unit propagation (as in CDCL)

# CDSAT module for equality with uninterpreted functions

- $\Sigma_{\mathsf{EUF}} = \langle S, F \rangle$ prop $\in S$ $\simeq_s \in F$ for all sorts $s \in S$
- $\mathsf{EUF}^+$ may be trivial or add countably many values for each $s \in S \setminus \{\text{prop}\}$ used as labels of congruence classes, e.g.:
  $t_1 \leftarrow \mathfrak{c},\ t_2 \leftarrow \mathfrak{c},\ t_3 \leftarrow \mathfrak{c}_3,\ t_4 \leftarrow \mathfrak{c}_4,\ t_5 \leftarrow \mathfrak{c}_5$
  shorter than
  $t_1 \simeq t_2,\ t_1 \not\simeq t_3,\ t_1 \not\simeq t_4,\ t_1 \not\simeq t_5,\ t_3 \not\simeq t_4,\ t_3 \not\simeq t_5,\ t_4 \not\simeq t_5$
- 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

# Example where CDSAT emulates equality sharing

1. $\{x \leq y,\ y \leq (x + g(x)),\ P(h(x) - h(y)),\ \neg P(0),\ g(x) \simeq 0\}$
   Theory union: LIA $\cup$ EUF
2. $S{=}\{x{\leq}y, y{\leq}(x{+}g(x)), f(h(x){-}h(y)) \simeq \bullet, f(0) \not\simeq \bullet, g(x) \simeq 0\}$
   $\mathcal{V}_{\mathrm{sh}}(S) = \{x,\ y,\ g(x),\ h(x),\ h(y),\ h(x) - h(y),\ 0\}$
3. LIA-module: $\{y \leq x + g(x),\ g(x) \simeq 0\} \vdash_{\mathsf{LIA}} y \leq x$
   Deduce: ${}_{J\vdash}(y \leq x)$ (level 0)
   with $J = \{y \leq x + g(x),\ g(x) \simeq 0\}$
   /* step hidden in black-box LIA-procedure in equality sharing */
4. LIA-module: $\{x \leq y,\ {}_{J\vdash}(y \leq x)\} \vdash_{\mathsf{LIA}} x \simeq y$
   Deduce: ${}_{H\vdash}(x \simeq y)$ (level 0)
   with $H = \{x \leq y,\ {}_{J\vdash}(y \leq x)\}$

# Example where CDSAT emulates equality sharing

5. EUF-module: ${}_{H\vdash}(x \simeq y) \vdash_{\mathsf{EUF}} h(x) \simeq h(y)$
   Deduce: ${}_{I\vdash}(h(x) \simeq h(y))$ (level 0)
   with $I = \{{}_{H\vdash}(x \simeq y)\}$
6. LIA-module: ${}_{I\vdash}(h(x) \simeq h(y)) \vdash_{\mathsf{LIA}} h(x) - h(y) \simeq 0$
   Deduce: ${}_{K\vdash}(h(x) - h(y) \simeq 0)$ (level 0)
   with $K = \{{}_{I\vdash}(h(x) \simeq h(y))\}$
7. EUF-module:
   $\{f(h(x) - h(y)) \simeq \bullet,\ {}_{K\vdash}(h(x) - h(y) \simeq 0)\} \vdash_{\mathsf{EUF}} f(0) \simeq \bullet$
   but the trail contains $f(0) \not\simeq \bullet$
   EUF-conflict:
   $E = \{f(h(x) - h(y)) \simeq \bullet,\ {}_{K\vdash}(h(x) - h(y) \simeq 0),\ f(0) \not\simeq \bullet\}$
   (level 0)
   Fail returns unsat (nowhere to backjump to)

# CDSAT can emulate equality sharing

- Each $\mathcal{T}_k$ module can also place decisions on the shared trail by Decide transitions
- A $\mathcal{T}_k$-inference $J \vdash_k L$ from $J \subseteq \Gamma$ leads to $\mathcal{T}_k$-conflict $E = J \cup \{\overline{L}\}$ if $\overline{L} \in \Gamma$
- Solved by LearnBackjump

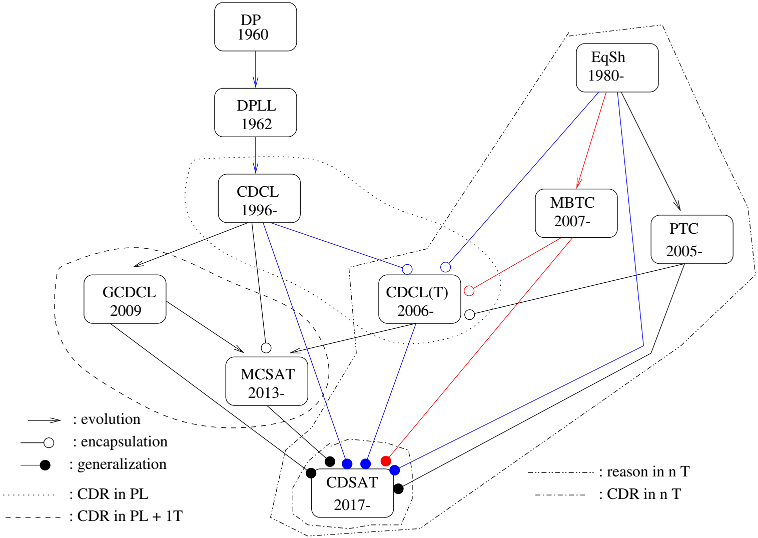
# Example where CDSAT emulates equality sharing: variant

1. $\{x \leq y,\ y \leq (x + g(x)),\ P(h(x) - h(y)),\ \neg P(0),\ g(x) \simeq 0\}$
   theories: LIA $\cup$ EUF
2. $S{=}\{x{\leq}y, y{\leq}(x{+}g(x)), f(h(x){-}h(y)) \simeq \bullet, f(0) \not\simeq \bullet, g(x) \simeq 0\}$
   $\mathcal{V}_{\mathsf{sh}}(S) = \{x,\ y,\ g(x),\ h(x),\ h(y),\ h(x) - h(y),\ 0\}$
3. EUF-module: Decide adds ${}_{?}(x \not\simeq y)$ (level 1)
4. LIA-module: $\{y \leq x + g(x),\ g(x) \simeq 0\} \vdash_{\mathsf{LIA}} y \leq x$
   Deduce: ${}_{J\vdash}(y \leq x)$ (level 0)
   with $J = \{y \leq x + g(x),\ g(x) \simeq 0\}$ /* late propagation */
5. LIA-module: $\{x \leq y,\ {}_{J\vdash}(y \leq x)\} \vdash_{\mathsf{LIA}} x \simeq y$
   but the trail contains ${}_{?}(x \not\simeq y)$
   LIA-conflict: $E_0 = \{{}_{?}(x \not\simeq y),\ x \leq y,\ {}_{J\vdash}(y \leq x)\}$

# Example where CDSAT emulates equality sharing: variant

6. LIA-conflict: $E_0 = \{_?(x \not\simeq y),\ x \leq y,\ _{J\vdash}(y \leq x)\}$
   $_?(x \not\simeq y)$ has level 1, the rest has level 0
7. LearnBackjump: back to level 0 adding $_{H\vdash}(x \simeq y)$
   $H = \{x \leq y,\ _{J\vdash}(y \leq x)\}$
   the derivation continues as before

# The big picture: more theory combination



# Model-based theory combination (MBTC)

[de Moura, Bjørner: SMT 2007]

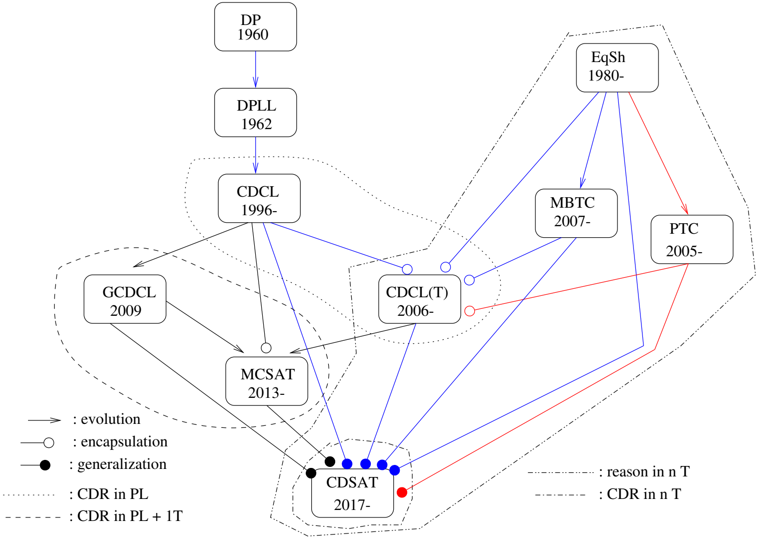
- Variant of equality sharing in CDCL($\mathcal{T}$)
- Assume $\mathcal{T}_k$-sat procedure builds candidate model $\mathcal{M}_k$ (e.g., linear arithmetic)
- Share $u \simeq v$ if true in $\mathcal{M}_k$ not necessarily $\mathcal{T}_k$-entailed by $S_k \cup \mathcal{E}_i$ ($u$ and $v$ $\mathcal{T}_k$-terms occurring in $S_k$)
- (Propositional abstraction of) $u \simeq v$ posted on trail as decision
- If $\mathcal{T}_k$-conflict ensues, undo, and update $\mathcal{M}_k$
- Useful to accelerate reaching sat

$\mathcal{M}_k$ and conflict-driven updates remain inside black-box procedure

# CDSAT generalizes MBTC

- All theory modules cooperate as peers to build a model for the union of the theories on the shared trail
- A model-constructing theory module $\mathcal{I}_k$ can build and update its model $\mathcal{M}_k$ publicly on the shared trail
- Any theory module can place a decision on the trail by a Decide transition
- A model-constructing theory module $\mathcal{I}_k$ can decide an equality $u \simeq v$ that follows from the assignments in $\mathcal{M}_k$
- CDSAT does not require model-constructing $\mathcal{T}_k$-sat procedures in MBTC's strong sense

# The big picture: more theory combination



# Polite theory combination (PTC)

[Ranise, Ringeissen, Zarba: FroCoS 2005] [Jovanović, Barrett: LPAR 2010] [Sheng et al.: CADE 2021] [Toledo, Przybocki, Zohar: CADE 2025]

- Variant of equality sharing in CDCL($\mathcal{T}$)
- Equality sharing requires the theories to be stably infinite
- PTC allows $\mathcal{T}_1$ not stably infinite, but $\mathcal{T}_2$ satisfies stronger cardinality requirements: strongly polite
- PTC combines theories by combining procedures
- Procedures combined as black-boxes
- Completeness approach like equality sharing: hypotheses on theories + combination theorem

CDSAT requires neither stable infiniteness nor strong politeness

# CDSAT and agreement on cardinalities of sorts

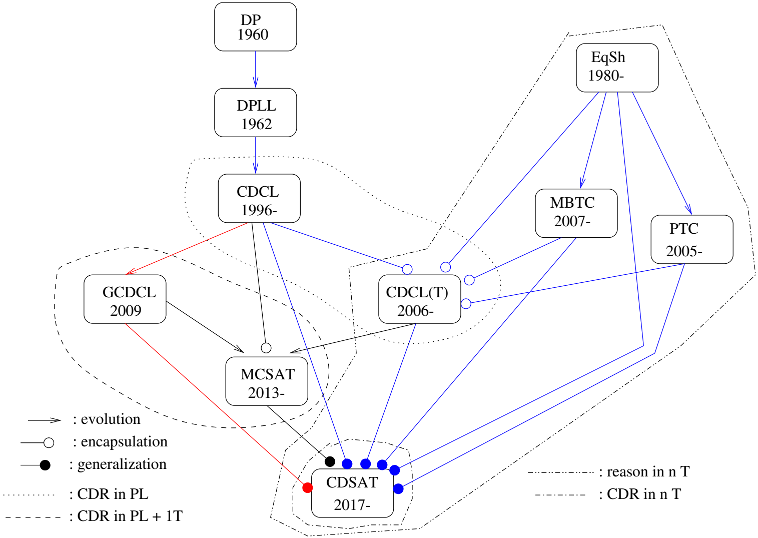
- CDSAT requires that there exists leading theory, say $\mathcal{T}_1$, that
  - Has all sorts in the theory union
  - Has all cardinality constraints aggregated and enforced by $\mathcal{T}_1$-module inferences
- Every $\mathcal{T}_k$ $(k \neq 1)$ has to agree with $\mathcal{T}_1$ on what's shared: any two $\mathcal{T}_k$ and $\mathcal{T}_j$ $(k \neq j)$ agree
- Agreement guaranteed by theory modules completeness requirements
- Different approach to completeness:
  - $\mathcal{T}_1$-module complete
  - $\mathcal{T}_k$-module $(k \neq 1)$ leading-theory-complete

# Examples

1. All theories stably infinite: $\mathcal{T}_1$ is fictional $\mathcal{T}_{\mathbb{N}}$ that interprets all sorts (except prop) as having the cardinality of $\mathbb{N}$
2. At-most-$m$ cardinality constraint on sort $s$:
   $\forall x_0, \ldots, \forall x_m.\ \bigvee_{0 \leq i \neq k \leq m} x_i \simeq_s x_k$
   $x_0, \ldots, x_m$: $m+1$ distinct variables of sort $s$

   Inference rule in the $\mathcal{T}_1$-module:
   $\bigwedge_{0 \leq i \neq k \leq m} u_i \not\simeq_s u_k \vdash_{\mathcal{T}_1} \bot$
   $u_0, \ldots u_m$: any $m+1$ distinct terms of sort $s$
3. Aggregation: if $\mathcal{T}_2$ says at-most-$m$ and $\mathcal{T}_2$ says at-most-$p$, $\mathcal{T}_1$ says at-most-$min(m, p)$

# The big picture: conflict-driven theory reasoning



# Conflict-driven satisfiability procedures in arithmetic

Generalize the CDCL pattern:

- Candidate model: theory model (e.g., LRA, LIA, NRA)
- Assignment: also to first-order terms
  (e.g., $x \leftarrow 3$, $x < y \leftarrow$true, $z \leftarrow y + 3$)
- Propagation: also evaluation of arithmetic expressions
  (e.g., $y \leftarrow 0 \vdash_{\mathsf{LRA}} (y > 2) \leftarrow$false)
- Explanation: also theory-conflicts by theory inferences
- Learn lemmas that may contain new (non-input) atoms and may exclude first-order assignments
- Expensive theory inferences only on demand to respond to conflicts

# Outline of GCDCL procedure for generic single theory $\mathcal{T}$

[McMillan, Kuehlmann, Sagiv: CAV 2009]

- Embed reasoning about disjunction into theory reasoning by generalizing to $\mathcal{T}$-clauses a theory reasoning inference rule for $\mathcal{T}$-literals
- Apply the generalized rule only to explain conflicts
- Devise restrictions to ensure termination

Achieved in GCDCL: linear rational arithmetic (LRA)

# Linear rational arithmetic (LRA)

- Input: set $S$ of LRA-clauses
- LRA-term: rational constant $c$, sum $c_1 \cdot x_1 + \ldots + c_n \cdot x_n$
- LRA-clause: disjunction of $t_1 \lessdot t_2$ literals, $\lessdot \in \{<, \leq\}$
- $\overline{(t_1 < t_2)}$ and $\overline{(t_1 \leq t_2)}$ replaced by $t_2 \leq t_1$ and $t_2 < t_1$
- $t_1 \simeq t_2$ rewritten as $t_1 \leq t_2$ and $t_2 \leq t_1$
- Variable $x$ with positive coefficient:
  rearrange literal into upper bound $x \lessdot t$
- Variable $x$ with negative coefficient:
  rearrange literal into lower bound $t \lessdot x$

# Linear rational arithmetic (LRA)

- Fourier-Motzkin (FM) resolution:
  $\{t_1 \lessdot_1 x,\ x \lessdot_2 t_2\} \vdash_{\mathsf{LRA}} t_1 \lessdot_3 t_2$
  $\lessdot_1, \lessdot_2, \lessdot_3 \in \{<, \leq\}$
  $\lessdot_3$ is $<$ if either $\lessdot_1$ or $\lessdot_2$ is $<$ and $\leq$ otherwise
  - Transitive closure: $\{x < -y,\ -y < -2\} \vdash_{\mathsf{LRA}} x < -2$
  - Linear combination of constraints:
    $\{x + y < 0,\ -y + 2 < 0\} \vdash_{\mathsf{LRA}} x + 2 < 0$
- Fourier-Motzkin algorithm:
  termination guaranteed
  (elim one var at each round, finitely many variables)
  but generates a doubly exponential number of constraints

[Lassez, Maher: JAR 1992]

# Generalized CDCL (GCDCL) for LRA

[McMillan, Kuehlmann, Sagiv: CAV 2009]

- Generalize FM-resolution to LRA-clauses: shadow rule e.g.:
  $\{(b < d) \vee (c < d),\ d < a\} \vdash_{\mathrm{LRA}} (b < a) \vee (c < a)$
- Generates new (non-input) atoms
- Applied only to explain LRA-conflicts
  generating lemmas excluding LRA-assignments
- Add restrictions to recover termination:
  assume fixed total ordering $\prec_{\mathrm{LRA}}$ on rational variables
  apply inference only if the variable resolved upon is
  $\prec_{\mathrm{LRA}}$-maximum in both premises

Independently:

[Korovin, Tsiskaridze, Voronkov: CP 2009] [Cotton: FORMATS 2010]

# CDSAT module for linear rational arithmetic (LRA)

- Signature $\Sigma_{\mathsf{LRA}}$:
  - Sorts: $S = \{\mathsf{prop}, \mathsf{Q}\}$
  - Symbols: $\simeq_s$ for all $s \in S$
    $1, +, <, \leq, q\cdot$ for all rational numbers $q \in \mathbb{Q}$
- Theory extension $\mathsf{LRA}^+$ adds constants $\tilde{q}$ for all $q \in \mathbb{Q}$
- Inference rules:
  - Evaluation: $(t_1 \leftarrow \tilde{q}_1, \ldots, t_m \leftarrow \tilde{q}_m) \vdash_{\mathsf{LRA}} l \leftarrow \mathfrak{b}$
  - Disequality elimination:
    $t_1 \leq x,\ x \leq t_2,\ t_1 \simeq_{\mathsf{Q}} t_0,\ t_2 \simeq_{\mathsf{Q}} t_0,\ x \not\simeq_{\mathsf{Q}} t_0 \vdash_{\mathsf{LRA}} \bot$
    detects LRA-conflict: no value for variable $x$

# CDSAT module for linear rational arithmetic (LRA)

- FM-resolution: $\{t_1 \lessdot_1 x,\ x \lessdot_2 t_2\} \vdash_{\mathrm{LRA}} t_1 \lessdot_3 t_2$
  $\lessdot_1, \lessdot_2, \lessdot_3 \in \{<, \leq\}$
  $\lessdot_3$ is $<$ if either $\lessdot_1$ or $\lessdot_2$ is $<$ and $\leq$ otherwise
- $\mathrm{basis}_{\mathrm{LRA}}(X)$: subterms, equalities, disequalities restricting FM-resolution to resolve on the $\prec_{\mathrm{LRA}}$-maximum variable
- Detection of empty solution space:
  $\{y_1 \leftarrow \tilde{q}_1, \ldots, y_m \leftarrow \tilde{q}_m\} \uplus E \vdash_{\mathrm{LRA}} \bot$
  for all $x$ in $E$, $x \prec_{\mathrm{LRA}} y_i$ or $x = y_i$ for some $i$ $(1 \leq i \leq m)$
- Alternatively and in practice: sensible search plan that selects rational variables for decision in $\prec_{\mathrm{LRA}}$-increasing order

# More on CDSAT

For CDSAT at work on conflict-driven theory reasoning, we need:

- Acceptability of first-order decisions
- Transition rule Deduce beyond unit propagation and deduction of equalities between shared variables
- Transition rule to solve conflicts due to first-order decisions: UndoClear

Let's also have a more formal look at the CDSAT trail

# 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 $H \vdash_k A$
  - Input assignment ($H = \emptyset$)
  - Due to conflict-solving transitions
  - Boolean or input first-order assignment
- 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

# Acceptability of a decision

- Boolean decision ${}_{?}L$: it suffices $L \notin \Gamma$ and $\overline{L} \notin \Gamma$
- First-order decision ${}_{?}(u \leftarrow \mathfrak{c})$
  where $\mathfrak{c}$ is a $\mathcal{T}_k$-value:
  - Trail $\Gamma$ does not assign a $\mathcal{T}_k$-value to term $u$
  - $u \leftarrow \mathfrak{c}$ does not trigger a $\mathcal{T}_k$-inference $J \cup \{u \leftarrow \mathfrak{c}\} \vdash_k \overline{L}$
    with $J \subseteq \Gamma$ and $L \in \Gamma$
  - Excluding a first-order decision that triggers an immediate conflict from which nothing can be learned

# CDSAT transition rule Deduce

- Propagation:
  - Boolean propagation: e.g., unit propagation
  - $\mathcal{T}_k$-propagation: e.g., propagation of equalities when emulating equality sharing
- $\mathcal{T}_k$-inferences that explain a $\mathcal{T}_k$-conflict
  generating lemmas excluding $\mathcal{T}_k$-assignments
  until the $\mathcal{T}_k$-conflict can be detected
  as a Boolean conflict on the trail:
  $J \vdash_k L$ and $\overline{L} \in \Gamma$
  unsatisfiable assignment $E = J \cup \{\overline{L}\}$

# CDSAT transition rule UndoClear

- The assignment of max level in the conflict is a first-order decision
- A first-order assignment does not have a complement that can be learned
- UndoClear incorporates backtracking from the level of the bad decision to the previous one
- The state has changed due to a late propagation
- UndoClear fires after a late propagation:
  bad decision was acceptable prior to the late propagation;
  causes a conflict afterwards

# Example with UndoClear

$\{l_0\colon 2x+y\simeq 1,\ l_1\colon 2x+2y\simeq 1\}$ subset of the input (level 0)

1. Decide: ${}_{?}(x\leftarrow 0)$ (level 1) /* acceptable */
2. Deduce: ${}_{J\vdash}(y\simeq 0)$ with $J=\{2x+y\simeq 1,\ 2x+2y\simeq 1\}$ (level 0)
   FM-resolution: $\{2x+y\simeq 1,\ 2x+2y\simeq 1\}\vdash_{\mathsf{LRA}} y\simeq 0\ (l_1-l_0)$
   /* late propagation */
3. $\{{}_{?}(x\leftarrow 0),\ {}_{J\vdash}(y\simeq 0)\}\vdash_{\mathsf{LRA}} 2x+y\not\simeq 1$ detects
   LRA-conflict $E=\{{}_{?}(x\leftarrow 0),\ {}_{J\vdash}(y\simeq 0),\ 2x+y\simeq 1\}$
   UndoClear: undo ${}_{?}(x\leftarrow 0)$ (max level in $E$) back to level 0
4. Decide: ${}_{?}(x\leftarrow 1/2)$ (level 1)
   /* forced decision: only acceptable value for $x$ */

# Example of non-termination of FM-resolution

Infinite sequence of FM-resolutions alternating on distinct variables:

$$
\begin{array}{llll}
l_0: & -2{\cdot}x - y < 0 & & \\
l_1: & x + y < 0 & & \\
l_2: & x < -1 & & \\
l_3: & -y < -2 & (l_0 + 2l_2) & \mathit{elim}\ x \\
l_4: & x < -2 & (l_1 + l_3) & \mathit{elim}\ y \\
l_5: & -y < -4 & (l_0 + 2l_4) & \mathit{elim}\ x \\
l_6: & x < -4 & (l_1 + l_5) & \mathit{elim}\ y \\
l_7: & -y < -8 & (l_0 + 2l_6) & \mathit{elim}\ x \\
\ldots & \ldots & \ldots & \ldots
\end{array}
$$

It may arise even if FM-resolution is applied
only to respond to LRA-conflicts

# Example where CDSAT emulates GCDCL

$l_0: \; -2{\cdot}x - y < 0,\;\; l_1: x + y < 0,\;\; l_2: x < -1$ (level 0)

1. Decide: $_?(y{\leftarrow}0)$ (level 1) /* acceptable */
   LRA-conflict: $\{-2{\cdot}x - y < 0,\; x < -1,\; y{\leftarrow}0\}$
2. Explained by $l_0 + 2l_2$: $\{-y < 2{\cdot}x,\; 2{\cdot}x < -2\} \vdash_{\mathsf{LRA}} -y < -2$
   Deduce: $l_3: -y < -2$ (level 0) /* late propagation */
3. $y{\leftarrow}0 \vdash_{\mathsf{LRA}} \overline{-y < -2}$ detects LRA-conflict $\{y{\leftarrow}0,\; -y < -2\}$
   UndoClear: undo $_?(y{\leftarrow}0)$ and back to level 0
4. Decide: $_?(x{\leftarrow}-2)$ (level 1) /* acceptable */
   LRA-conflict: $\{x + y < 0,\; -y < -2,\; x{\leftarrow}-2\}$
5. Explained by $l_1 + l_3$: $\{x < -y,\; -y < -2\} \vdash_{\mathsf{LRA}} x < -2$
   Deduce: $l_4: x < -2$ (level 0) /* late propagation */

# Example where CDSAT emulates GCDCL

6. $x \leftarrow -2 \vdash_{\mathsf{LRA}} \overline{x < -2}$ detects LRA-conflict $\{x \leftarrow -2,\ x < -2\}$
   UndoClear: undo $_?(x \leftarrow -2)$ and back to level 0
7. Decide: $_?(y \leftarrow 3)$ (level 1) /* acceptable */
   LRA-conflict: $\{-2 \cdot x - y < 0,\ x < -2,\ y \leftarrow 3\}$
8. Explained by $l_0 + 2l_4$: $\{-y < 2 \cdot x,\ 2 \cdot x < -4\} \vdash_{\mathsf{LRA}} -y < -4$
   Deduce: $l_5$: $-y < -4$ (level 0) /* late propagation */
9. $y \leftarrow 3 \vdash_{\mathsf{LRA}} \overline{-y < -4}$ detects LRA-conflict $\{y \leftarrow 3,\ -y < -4\}$
   UndoClear: undo $_?(y \leftarrow 3)$ and back to level 0
10. Decide: $_?(x \leftarrow -3)$ (level 1) /* acceptable */
    LRA-conflict: $\{x + y < 0,\ -y < -4,\ x \leftarrow -3\}$

# Example where CDSAT emulates GCDCL

11. Explained by $l_1 + l_5$: $\{x < -y,\ -y < -4\} \vdash_{\mathsf{LRA}} x < -4$
    Deduce: $l_6 : x < -4$ (level 0) /* late propagation */
12. $x \leftarrow -3 \vdash_{\mathsf{LRA}} \overline{x < -4}$ detects LRA-conflict $\{x \leftarrow -3,\ x < -4\}$
    UndoClear: undo $_?(x \leftarrow -3)$ and back to level 0
13. Decide: $_?(y \leftarrow 5)$ (level 1) /* acceptable */
    LRA-conflict: $\{-2{\cdot}x - y < 0,\ x < -4,\ y \leftarrow 5\}$
14. Explained by $l_0 + 2l_6$: $\{-y < 2{\cdot}x,\ 2{\cdot}x < -8\} \vdash_{\mathsf{LRA}} -y < -8$
    Deduce: $l_7 : -y < -8$ (level 0) /* late propagation */
15. $y \leftarrow 5 \vdash_{\mathsf{LRA}} \overline{-y < -8}$ detects LRA-conflict $\{y \leftarrow 5,\ -y < -8\}$
    UndoClear: undo $_?(y \leftarrow 5)$ and back to level 0

$\ldots \quad \ldots \quad \ldots$

# Example where CDSAT emulates GCDCL

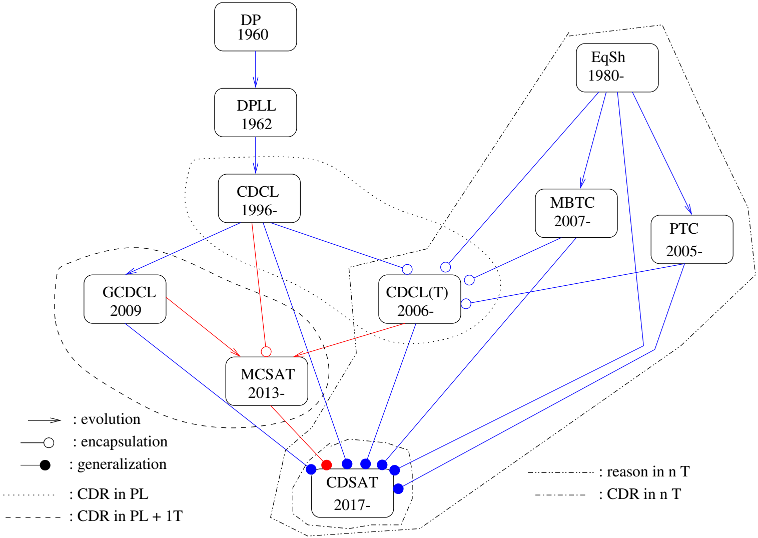
- Assume $y \prec_{\text{LRA}} x$
- 2nd FM-resolution inference in the non-halting sequence:
  $\{x < -y,\ -y < -2\} \vdash_{\text{LRA}} x < -2$
  is barred: it resolves on $y$ when $x$ occurs in the premises
- All GCDCL or CDSAT derivations embedding that diverging series of FM-resolution inferences are barred
- Multiple CDSAT-derivations discover that
  $l_0\colon\ -2{\cdot}x - y < 0,\ \ l_1\colon x + y < 0,\ \ l_2\colon x < -1$
  is LRA-unsatisfiable
- A simple one does it by mere LRA-propagations at level 0

# Example where CDSAT emulates GCDCL

$l_0\colon\ -2{\cdot}x - y < 0,\ l_1\colon x + y < 0,\ l_2\colon x < -1$ (level 0)
Assume $y \prec_{\mathrm{LRA}} x$

1. Deduce: $l_3\colon -y < -2$ (level 0)
   $l_0 + 2l_2\colon \quad \{-y < 2{\cdot}x,\ 2{\cdot}x < -2\} \vdash_{\mathrm{LRA}} -y < -2$
   /* $x$ is $\prec_{\mathrm{LRA}}$-max variable in both premises */
2. Deduce: $l_4\colon y < 0$ (level 0) /*normal form of $-y < -2{\cdot}y$ */
   $l_0 + 2l_1\colon \quad \{-y < 2{\cdot}x,\ 2{\cdot}x < -2{\cdot}y\} \vdash_{\mathrm{LRA}} -y < -2{\cdot}y$
   /* $x$ is $\prec_{\mathrm{LRA}}$-max variable in both premises */
3. Deduce: $l_5\colon 2 < 0$ (level 0)
   $-l_3 + l_4\colon \quad \{2 < y,\ y < 0\} \vdash_{\mathrm{LRA}} 2 < 0$
   /* $y$ is $\prec_{\mathrm{LRA}}$-max variable in both premises as there is no $x$*/
4. $\emptyset \vdash_{\mathrm{LRA}} \overline{2 < 0}$ reveals LRA-conflict at level 0
   so that Fail returns unsat

# The big picture: better conflict-driven theory reasoning



# From GCDCL to MCSAT

Conflict-driven satisfiability procedures for sets of $\mathcal{T}$-literals:

- LIA: Cutting-to-the-chase procedure
  [Jovanović, de Moura: CADE 2011, JAR 2013]
  [Bromberger et al.: CADE 2015]
- NRA: NLSAT
  [Jovanović, de Moura: IJCAR 2012]
- Use first-order assignments
- Explain conflicts by inferences that generate new atoms and exclude first-order assignments

Conflict-driven satisfiability procedures for sets of $\mathcal{T}$-clauses?

# From GCDCL to MCSAT

- No need to generalize to $\mathcal{T}$-clauses an inference rule for $\mathcal{T}$-literals
- Entrust the reasoning about disjunction to CDCL
- Integrate in CDCL a conflict-driven $\mathcal{T}$-satisfiability procedure for sets of $\mathcal{T}$-literals
- CDCL($\mathcal{T}$)?
  No, it allows neither first-order assignment
  nor new atoms on the trail
  nor $\mathcal{T}$-inferences generating lemmas excluding $\mathcal{T}$-assignments
- MCSAT (Model-Constructing SATisfiability)
  [de Moura, Jovanović: VMCAI 2013]
  [Jovanović, Barrett, de Moura: FMCAD 2013]

# MCSAT (Model-Constructing SATisfiability)

- Integrate CDCL and one model-constructing conflict-driven $\mathcal{T}$-sat procedure for sets of $\mathcal{T}$-literals (called $\mathcal{T}$-plugin) that
  - Has access to the trail
  - Proposes assignments to first-order terms: $\mathcal{T}$-assignment
  - Computes $\mathcal{T}$-propagations
  - Explains $\mathcal{T}$-conflicts by $\mathcal{T}$-inferences generating lemmas excluding $\mathcal{T}$-assignments
  - Lemma may contain new (i.e., non-input) atoms coming from a finite basis for termination
- CDCL and the $\mathcal{T}$-plugin cooperate in model construction
- Both propositional and $\mathcal{T}$-reasoning are conflict-driven

# CDSAT generalizes MCSAT

- CDSAT generalizes MCSAT to generic union $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
- MCSAT is not a combination calculus
  hence does not cover, e.g.:
  - Interaction of multiple first-order theories on the trail
  - Conflict-drivenness for more than one first-order theory
  - Combination of conflict-driven and black-box procedures
  - Soundness, completeness, termination for theory combination
  - Construction of finite global basis from local ones
- CDSAT does not require model-constructing $\mathcal{T}_k$-sat procedures in MCSAT's strong sense

# CDSAT generalizes MCSAT

- CDSAT and MCSAT have different transition systems:
  - MCSAT evaluation mechanism $\rightsquigarrow$ $\mathcal{T}_k$-inferences in CDSAT
  - Explanation function in MCSAT $\rightsquigarrow$ $\mathcal{T}_k$-inferences in CDSAT
- CDSAT provides foundations for instances of theory combination in MCSAT implementations, e.g.: Bool $\cup$ EUF $\cup$ LRA [Jovanović, Barrett, de Moura: FMCAD 2013]
- CDSAT allows predicate-sharing theories, MCSAT assumes disjoint theories

CDSAT reduces to MCSAT if theory union contains only Bool and one theory $\mathcal{T}$ equipped with a conflict-driven model-constructing $\mathcal{T}$-sat procedure for sets of $\mathcal{T}$-literals

# Example where CDSAT emulates MCSAT

$x < y,\ x < z,\ (y < w) \vee (z < w),\ w < x$ (level 0)
Assume $x \prec_{\mathrm{LRA}} y \prec_{\mathrm{LRA}} z \prec_{\mathrm{LRA}} w$ and a sensible search plan

1. Decide: ${}_?(x \leftarrow 0)$ (level 1) /* acceptable */
2. Decide: ${}_?(y \leftarrow 1)$ (level 2) /* acceptable */
   /* ${}_?(y \leftarrow 0)$ not acceptable: $\{x \leftarrow 0,\ y \leftarrow 0\} \vdash_{\mathrm{LRA}} \overline{(x < y)}$*/
3. Decide: ${}_?(z \leftarrow 1)$ (level 3) /* acceptable */
   /* ${}_?(z \leftarrow 0)$ not acceptable: $\{x \leftarrow 0,\ z \leftarrow 0\} \vdash_{\mathrm{LRA}} \overline{(x < z)}$*/
   LRA-conflict:
   $\{x \leftarrow 0,\ y \leftarrow 1,\ z \leftarrow 1,\ w < x,\ (y < w) \vee (z < w)\}$
   Equivalently: no acceptable value for $w$
   Disjunction: case analysis by Bool-module

# Example where CDSAT emulates MCSAT

4. Decide: ${}_{?}(y < w)$ (level 4)
5. Deduce: ${}_{J\vdash}(y < x)$ (level 4)
   $J = \{{}_{?}(y < w),\ {}_{\emptyset\vdash}(w < x)\}$ (level 4)
   $\{{}_{?}(y < w),\ {}_{\emptyset\vdash}(w < x)\} \vdash_{\mathsf{LRA}} y < x$
   /* $w$ is $\prec_{\mathsf{LRA}}$-max variable in both $y < w$ and $w < x$ */
6. Deduce: ${}_{I\vdash}(x < x)$ (level 4)
   $I = \{{}_{\emptyset\vdash}(x < y),\ {}_{J\vdash}(y < x)\}$ (level 4)
   $\{{}_{\emptyset\vdash}(x < y),\ {}_{J\vdash}(y < x)\} \vdash_{\mathsf{LRA}} x < x$
   /* $y$ is $\prec_{\mathsf{LRA}}$-max variable in both $x < y$ and $y < x$ */
   LRA-conflict: $E_0 = \{{}_{I\vdash}(x < x)\}$
7. Resolve: $E_1 = \{{}_{\emptyset\vdash}(x < y),\ {}_{J\vdash}(y < x)\}$
8. Resolve: $E_2 = \{{}_{\emptyset\vdash}(x < y),\ {}_{?}(y < w),\ {}_{\emptyset\vdash}(w < x)\}$

# Example where CDSAT emulates MCSAT

9. LearnBackjump: back to level 0 adding ${}_{H\vdash}(\overline{y < w})$
   $H = \{{}_{\emptyset\vdash}(x < y),\ {}_{\emptyset\vdash}(w < x)\}$
   /* 0 is smallest level where $\overline{y < w}$ is undefined */
10. Deduce: ${}_{G\vdash}(z < w)$ (level 0)
    $G = \{{}_{H\vdash}(\overline{y < w}),\ {}_{\emptyset\vdash}((y < w) \vee (z < w))\}$ (level 0)
    $\{{}_{H\vdash}(\overline{y < w}),\ {}_{\emptyset\vdash}((y < w) \vee (z < w))\} \vdash_{\mathsf{Bool}} z < w$
    /* shadow rule unnecessary: Bool-module handles $\vee$ by decision and unit propagation; LRA-module reasons about LRA-literals */
11. Deduce: ${}_{K\vdash}(z < x)$ (level 0)
    $K = \{{}_{G\vdash}(z < w),\ {}_{\emptyset\vdash}(w < x)\}$ (level 0)
    $\{{}_{G\vdash}(z < w),\ {}_{\emptyset\vdash}(w < x)\} \vdash_{\mathsf{LRA}} z < x$
    /* $w$ is $\prec_{\mathsf{LRA}}$-max variable in both $z < w$ and $w < x$ */

# Example where CDSAT emulates MCSAT

12. Deduce: ${}_{M\vdash}(x < x)$ (level 0)
    $M = \{{}_{\emptyset\vdash}(x < z),\ {}_{\kappa\vdash}(z < x)\}$ (level 0)
    $\{{}_{\emptyset\vdash}(x < z),\ {}_{\kappa\vdash}(z < x)\} \vdash_{\mathsf{LRA}} x < x$
    /* $z$ is $\prec_{\mathsf{LRA}}$-max variable in both $x < z$ and $z < x$ */
13. LRA-conflict: $E_3 = \{{}_{M\vdash}(x < x)\}$ (level 0)
    Fail returns unsat

- Deduce covers both conflict explanation and propagation
- CDSAT can apply inferences (e.g., FM-resolution) more liberally than MCSAT

# 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
- Key abstraction in CDSAT: open the black-boxes
  pull out the $\mathcal{T}_k$-inference systems
  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

# Theory view of an assignment

It defines what a theory sees of an assignment:

- $\mathcal{T}_k$-view of assignment $H$, written $H_k$:
  - $\mathcal{T}_k$-assignments in $H$: those that assign $\mathcal{T}_k$-values
  - $u \simeq t$ if $H$ contains $u \leftarrow \mathfrak{c}$ and $t \leftarrow \mathfrak{c}$ of a $\mathcal{T}_k$ sort $s$ ($s \neq \mathrm{prop}$)
  - $u \not\simeq t$ if $H$ contains $u \leftarrow \mathfrak{c}$ and $t \leftarrow \mathfrak{q}$ with $\mathfrak{c} \neq \mathfrak{q}$

  including $u \leftarrow \mathfrak{c}$ and $t \leftarrow \mathfrak{c}$ made by $\mathcal{T}_j$ ($k \neq j$) for $s$ shared
- Global view:
  - The $\mathcal{T}$-view of $H$ for $\mathcal{T} = \bigcup_{k=1}^{n} \mathcal{T}_k$
  - $H_{\mathcal{T}}$ has everything

Key notion for theory combination (MCSAT does not have it)

# Theory view: example

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

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

# Relevance

Term $u$ is relevant to $\mathcal{T}_k$ in assignment $J$ if

- Either $u$ occurs in $J$ (also as subterm) and $\mathcal{T}_k$ has the sort $s$ of $u$ and has values for $s$
- Term $u$ is an equality $u_1 \simeq_s u_2$ s. t. $u_1$ and $u_2$ occur in $J$, and $\mathcal{T}_k$ has sort $s$, but does not have values for $s$
- Term $u$ is a Boolean term $p(u_1, \ldots, u_m)$ s. t. $p$ is a predicate symbol that $\mathcal{T}_k$ shares with at least another theory, the $u_i$'s occur in $J$, and $\mathcal{T}_k$ has their sorts

Key notion for theory combination (MCSAT does not have it)

# Relevance: example

- $H = \{x \leftarrow 5,\ f(x) \leftarrow 2,\ f(y) \leftarrow 3\}$
- $x, y\colon \mathrm{Q}$, $f\colon \mathrm{Q} \rightarrow \mathrm{Q}$, LRA and EUF share sort Q
- $H_{\mathrm{LRA}} = H \cup \{x \not\simeq f(x),\ x \not\simeq f(y),\ f(x) \not\simeq f(y)\}$
- $H_{\mathrm{EUF}} = \{x \not\simeq f(x),\ x \not\simeq f(y),\ f(x) \not\simeq f(y)\}$
- $x$ 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 $x \simeq y$

# Acceptability revisited

$\Gamma_{\mathcal{T}_k}$: the $\mathcal{T}_k$-view of trail $\Gamma$

A $\mathcal{T}_k$-assignment $u\leftarrow\mathfrak{c}$ is an acceptable decision $_?(u\leftarrow\mathfrak{c})$ for the $\mathcal{T}_k$-module if

1. Term $u$ is relevant to $\mathcal{T}_k$ in $\Gamma_{\mathcal{T}_k}$
2. $\Gamma_{\mathcal{T}_k}$ does not assign a $\mathcal{T}_k$-value to term $u$
3. If $u\leftarrow\mathfrak{c}$ is a first-order assignment: $t\leftarrow\mathfrak{c}$ does not trigger a $\mathcal{T}_k$-inference $J \cup \{u\leftarrow\mathfrak{c}\} \vdash_k \overline{L}$ with $J \subseteq \Gamma_{\mathcal{T}_k}$ and $L \in \Gamma_{\mathcal{T}_k}$

# CDSAT transition rule UndoDecide

- The assignment of max level in conflict $E$ is a justified assignment ${}_{J}{\vdash}L$ where $J$ contains a first-order decision ${}_{?}A$ such that $\mathsf{level}_\Gamma({}_{?}A) = \mathsf{level}_\Gamma(J) = \mathsf{level}_\Gamma(E)$
- UndoDecide undoes ${}_{?}A$, backtracks, and puts $\overline{L}$ on the trail
- A first-order assignment does not have a complement, but its Boolean consequence does
- Resolve is forbidden: replacing ${}_{J}{\vdash}L$ with $J$ in $E$ and undoing ${}_{?}A$ by UndoClear can cause a loop if Decide reiterates ${}_{?}A$

# CDSAT module for arrays

- Signature $\Sigma_{\mathsf{Arr}}$:
  - Sorts: $S = \{\mathsf{prop}, \mathsf{I}, \mathsf{V}, \mathsf{A}\}$, I: indices, V: (array) values, A: arrays with indices of sort I and values of sort V
  - Symbols: $\simeq_s$ for all $s \in S$, select (read), store (write)
- Theory extension $\mathsf{Arr}^+$ may be trivial or add countably many values for each $s \in S \setminus \{\mathsf{prop}\}$
- Inference rules corresponding to the select-over-store axioms:
  1. $i \simeq j \longrightarrow \mathsf{select}(\mathsf{store}(a, i, v), j) \simeq v$
     $\{i \simeq j,\ b \simeq \mathsf{store}(a, i, v),\ \mathsf{select}(b, j) \not\simeq v\} \vdash_{\mathsf{Arr}} \bot$
  2. $i \not\simeq j \longrightarrow \mathsf{select}(\mathsf{store}(a, i, v), j) \simeq \mathsf{select}(a, j)$
     $\{i \not\simeq j,\ b \simeq \mathsf{store}(a, i, v),\ \mathsf{select}(b, j) \not\simeq \mathsf{select}(a, j)\} \vdash_{\mathsf{Arr}} \bot$

# CDSAT module for arrays with extensionality

- Extensionality axiom:
  $(\forall i.\ \mathsf{select}(a,i) \simeq \mathsf{select}(b,i)) \longrightarrow a \simeq b$
- Clausal form:
  $\mathsf{select}(a, \mathsf{diff}(a,b)) \not\simeq \mathsf{select}(b, \mathsf{diff}(a,b)) \vee a \simeq b$
  Skolem function $\mathsf{diff} : \mathsf{A} \times \mathsf{A} \rightarrow \mathsf{I}$ captures the witness index
- Inference rule:
  $a \not\simeq b \vdash_{\mathsf{Arr}} \mathsf{select}(a, \mathsf{diff}(a,b)) \not\simeq \mathsf{select}(b, \mathsf{diff}(a,b))$
- $\mathsf{basis}_{\mathsf{Arr}}(X)$: all subterms of terms in $X$, equalities btw them, and witness terms $\mathsf{select}(a, \mathsf{diff}(a,b))$, $\mathsf{select}(b, \mathsf{diff}(a,b))$

# Example with theory clauses and UndoDecide

- Input set $S$ contains clauses:
  - $C_1$: $(i \not\simeq j) \vee (\mathsf{select}(\mathsf{store}(a, i, v), j) < \mathsf{select}(a, j))$
  - $C_2$: $(\mathsf{select}(a, j) - \mathsf{select}(a, k)) \simeq 0$
  - $C_3$: $(\mathsf{select}(\mathsf{store}(a, i, v), j) \not< \mathsf{select}(a, j)) \vee$ $(\mathsf{select}(a, j) + \mathsf{select}(a, k) \simeq v)$
- Theory union: $\mathsf{Bool} \cup \mathsf{LRA} \cup \mathsf{Arr}$
- Suppose Arr interprets indices as integers: $\mathsf{I} = \mathsf{Z}$ and $\mathsf{Arr}^+$ adds integer constants as Arr-values

# Example with theory clauses and UndoDecide

1. Arr-module: Decide $_?(i \leftarrow 0)$ (level 1)
   /* acceptable as $i$ is relevant to Arr */
2. Arr-module: Decide $_?(j \leftarrow 0)$ (level 2)
3. Arr-module: equality inference $\{i \leftarrow 0, j \leftarrow 0\} \vdash_{\mathsf{Arr}} i \simeq j$
   Deduce: $A_1 : {}_J\vdash(i \simeq j)$ with $J = \{_?(i \leftarrow 0), {}_?(j \leftarrow 0)\}$ (level 2)
4. Bool-module: unit propagation
   $\{A_1,\ C_1\} \vdash_{\mathsf{Bool}} \mathsf{select}(\mathsf{store}(a, i, v), j) < \mathsf{select}(a, j)$
   Deduce: $A_2 : {}_I\vdash(\mathsf{select}(\mathsf{store}(a, i, v), j) < \mathsf{select}(a, j))$
   with $I = \{A_1,\ C_1\}$ (level 2)

# Example with theory clauses and UndoDecide

5. Bool-module: unit propagation
   $\{A_2,\ C_3\} \vdash_{\mathsf{Bool}} \mathsf{select}(a,j) + \mathsf{select}(a,k) \simeq v$
   Deduce: $A_3 \colon {}_{H\vdash}(\mathsf{select}(a,j) + \mathsf{select}(a,k) \simeq v)$
   with $H = \{A_2,\ C_3\}$ (level 2)
6. Arr-module: first select-over-store rule
   $\{A_1,\ A_2\} \vdash_{\mathsf{Arr}} v < \mathsf{select}(a,j)$
   Deduce: $A_4 \colon {}_{G\vdash}(v < \mathsf{select}(a,j))$
   with $G = \{A_1,\ A_2\}$ (level 2)
7. LRA-module: FM-resolution $A_3 + C_2$
   $\{A_3,\ C_2\} \vdash_{\mathsf{LRA}} \mathsf{select}(a,j) \simeq v/2$
   Deduce: $A_5 \colon {}_{M\vdash}(\mathsf{select}(a,j) \simeq v/2)$
   with $M = \{A_3,\ C_2\}$ (level 2)

# Example with theory clauses and UndoDecide

LRA-conflict: $E_0 = \{A_4,\ A_5\}$
as $A_4 : {}_{G}{\vdash}(v < \text{select}(a, j))$ and $A_5 : {}_{M}{\vdash}(\text{select}(a, j) \simeq v/2)$

8. $E_0$ contains literals $A_4$ and $A_5$ of max level (2)
   Resolve: $E_1 = \{A_4,\ A_3,\ C_2\}$
9. $E_1$ contains literals $A_3$ and $A_4$ of max level (2)
   Resolve: $E_2 = \{A_1,\ A_2,\ A_3,\ C_2\}$
10. $E_2$ contains literals $A_1$, $A_2$ and $A_3$ of max level (2)
    Resolve: $E_3 = \{A_1,\ A_2,\ C_3,\ C_2\}$
11. $E_3$ contains literals $A_1$, and $A_2$ of max level (2)
    Resolve: $E_4 = \{A_1,\ C_1,\ C_3,\ C_2\}$

# Example with theory clauses and UndoDecide

$E_4 = \{A_1,\ C_1,\ C_3,\ C_2\}$

$E_4$ contains one literal of max level: $\text{level}_\Gamma(A_1) = 2 = \text{level}_\Gamma(E_4)$

$A_1$ is ${}_{J\vdash}(i \simeq j)$ and $J = \{{}_{?}(i\leftarrow 0), {}_{?}(j\leftarrow 0)\}$
where ${}_{?}(j\leftarrow 0)$ also has level 2

Apply Resolve to replace $A_1$ with $J$
and UndoClear to undo ${}_{?}(j\leftarrow 0)$ ?

No, the system could loop by repeating ${}_{?}(j\leftarrow 0)$
(still acceptable)

# Example with theory clauses and UndoDecide

12. UndoDecide: undo ${}_?(j\leftarrow 0)$, backtrack to level 1, and add decision ${}_?(i \not\simeq j)$ (level 2)
    /* clause $C_1$ is satisfied */
13. LRA-module: Decide ${}_?(\mathrm{select}(a, j)\leftarrow 1)$ (level 3)
14. LRA-module: Decide ${}_?(\mathrm{select}(a, k)\leftarrow 1)$ (level 4)
    /* clause $C_2$ is satisfied */
15. LRA-module: Decide ${}_?(v\leftarrow 2)$ (level 5)
    /* clause $C_3$ is satisfied */

# Example with theory clauses and UndoDecide: variant

Suppose theory Arr does not have values for array indices:
$i$ and $j$ not relevant, Arr-module cannot decide their values

1. Arr-module: Decide ${}_{?}(i \simeq j)$ (level 1)
   /* acceptable as $i \simeq j$ is relevant to Arr */
2. The same transitions as before lead to conflict
   $\{{}_{?}(i \simeq j),\ C_1,\ C_3,\ C_2\}$ (level 1)
3. LearnBackjump backtracks to level 0 and places ${}_{N\vdash}(i \not\simeq j)$ on the trail with $N = \{C_1,\ C_3,\ C_2\}$
4. The satisfiability of the clauses can be detected as before

# The CDSAT transition system

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

# 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 $\Gamma_k$:

- $\Gamma_k$ does not assign a $\mathcal{T}_k$-value to $u$
- $u{\leftarrow}\mathfrak{c}$ first-order: no inference $J \cup \{u{\leftarrow}\mathfrak{c}\} \vdash_k L$ where $J \subseteq \Gamma_k$ and $\bar{L} \in \Gamma_k$
- $u$ is relevant to $\mathcal{T}_k$:
  either $u$ occurs in $\Gamma_k$ and $\mathcal{T}_k$ has $\mathcal{T}_k$-values for its sort;
  or $u$ is an equality whose sides occur in $\Gamma_k$,
  $\mathcal{T}_k$ has their sort, but not $\mathcal{T}_k$-values;
  or $u$ is Boolean term with $\mathcal{T}_k$-shared predicate whose arguments occur in $\Gamma_k$, and $\mathcal{T}_k$ has their sorts

# 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 $\mathcal{B}$ (global basis)
- Both $\mathcal{T}_k$-propagation and explanation of $\mathcal{T}_k$-conflicts

# The CDSAT trail rules: 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 conflict state rules: UndoClear

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

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

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

# Explanation of conflicts in CDSAT

- Explanation of a $\mathcal{T}_k$-conflict by $\mathcal{T}_k$-inferences encapsulated as Deduce steps: 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
- Switch 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 conflict state rules: 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)

# The CDSAT conflict state rules: 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 $H$
- Provided $H$ does not contain a first-order decision $A'$ that stands out as its level is maximum in the conflict $(\text{level}_\Gamma(A') = \text{level}_\Gamma(E \uplus \{A\}))$
- Avoiding a Resolve–UndoClear–Decide loop
- And what if there is such an $A'$? UndoDecide rule

# The CDSAT conflict state rules: UndoDecide

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

- $L$ 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

# The CDSAT conflict state rules: LearnBackjump

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

- $H$ contains only Boolean assignments: $H$ as $L_1 \wedge \ldots \wedge L_k$
- Since $E \uplus H \models \perp$, 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:
  $\mathrm{level}_\Gamma(E) \leq m < \mathrm{level}_\Gamma(H)$

# Assignments and models: endorsement

- Model $\mathcal{M}$ endorses ($\models$) $u \leftarrow \mathfrak{c}$:
  $\mathcal{M}$ 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 $u \not\simeq t$
  that is, $\mathcal{M}$ 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)
- $J \models L$: if $\mathcal{M} \models J_k$ then $\mathcal{M} \models L$
- Sound inference: if $J \vdash_k L$ then $J \models L$

# Three main theorems

Input assignment: $H$, all terms occurring in $H$ are in global basis $\mathcal{B}$

- CDSAT is
  - Sound: if all theory modules are sound,
    if CDSAT returns unsat, $H$ is unsatisfiable
  - Terminating: if $\mathcal{B}$ is finite,
    CDSAT is guaranteed to terminate
  - Complete: if the leading theory module is complete
    and the others are leading-theory-complete,
    if CDSAT terminates without returning unsat,
    there exists a $\mathcal{T}^+$-model of $\Gamma$ and hence of $H$

# 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 interactive theorem proving (correct by construction)
  - Trusted kernel of primitives

# Current and future work

- More theory modules: maps, vectors (aka dynamic arrays), vectors with concatenation (subsuming sequences and hence strings)
- Formulas with quantifiers: CDSAT(QSMA)
- CDSAT search plans: both global and local issues
  - Heuristic strategies to make decisions, prioritize theory inferences, control lemma learning
  - Efficient techniques to detect applicability of theory inference rules and acceptability of decisions
- Architecture of a CDSAT solver
- Baby verified implementation written in Rust by Xavier Denis: `https://github.com/xldenis/cdsat`

# References


- Satisfiability modulo theories and assignments.
  Proc. CADE-26, LNAI 10395, 42–59, Springer, Aug. 2017.
- Proofs in conflict-driven theory combination.
  Proc. CPP-7, ACM Press, 186–200, Jan. 2018.
- Conflict-driven satisfiability for theory combination: transition system and completeness.
  Journal of Automated Reasoning, 64(3):579–609, Mar. 2020.
- Conflict-driven satisfiability for theory combination: modules, lemmas, and proofs.
  Journal of Automated Reasoning, 66(1):43–91, Feb. 2022.

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

# References

- CDSAT for nondisjoint theories with shared predicates: arrays with abstract length.
  Proc. SMT-20, CEUR 3185, 18–37, CEUR WS-org, Aug. 2022.
- CDSAT for predicate-sharing theories: arrays, maps, and vectors with abstract domain.
  In preparation, 46 pages.

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

# References


- On conflict-driven reasoning.
  Proc. AFM-7, Kalpa Publications 5, 31–49, EasyChair, Apr. 2018.
- Conflict-driven reasoning in unions of theories. (Abstract)
  Proc. FroCoS-12, LNAI 11715, xi–xiii, Springer, Sept. 2019.
- Proof generation in CDSAT. (Abstract)
  Proc. PxTP-7, EPTCS 366, 1–4, Open Publishing Association, July 2021.
- The CDSAT method for satisfiability modulo theories and assignments: an exposition.
  Proc. CiE-21, LNAI 15764, 1–16, Springer, July 2025.

Author: MPB

# Thanks

Thank you!