max planck institut informatik

CDCL(T)

Automated Reasoning II

Max Planck Institute for Informatics

13th June 2017

Outline

Introduction

- Aim: decide satisfiability of a (quantifier-free) first-order formula with respect to a background theory
  - e.g. linear (real/integer) arithmetic, equality and uninterpreted functions, arrays, bitvectors, etc.

Example (Linear Integer Arithmetic)

$$(2x - 2y \leq 1) \wedge$$
$$((-2x + 2y \leq 1) \vee \bot) \wedge$$
$$(\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)) \wedge$$
$$((2x + 2y \leq 3) \vee (-2x + 2y \geq 2))$$

- Problem: Many decision procedures for theories (e.g. simplex) can only decide consistency of a conjunction of literals

Introduction

- Aim: decide satisfiability of a (quantifier-free) first-order formula with respect to a background theory
  - e.g. linear (real/integer) arithmetic, equality and uninterpreted functions, arrays, bitvectors, etc.

Example (Linear Integer Arithmetic)

$$(2x - 2y \leq 1) \wedge$$
$$((-2x + 2y \leq 1) \vee \bot) \wedge$$
$$(\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)) \wedge$$
$$((2x + 2y \leq 3) \vee (-2x + 2y \geq 2))$$

- Problem: Many decision procedures for theories (e.g. simplex) can only decide consistency of a conjunction of literals

Introduction

- Aim: decide satisfiability of a (quantifier-free) first-order formula with respect to a background theory
  - e.g. linear (real/integer) arithmetic, equality and uninterpreted functions, arrays, bitvectors, etc.

Example (Linear Integer Arithmetic)

$$
\begin{aligned}
&(2x - 2y \leq 1) \wedge \\
&((-2x + 2y \leq 1) \vee \bot) \wedge \\
&(\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)) \wedge \\
&((2x + 2y \leq 3) \vee (-2x + 2y \geq 2))
\end{aligned}
$$

- Problem: Many decision procedures for theories (e.g. simplex) can only decide consistency of a conjunction of literals

Lifting $\mathcal{T}$-Reasoning to Arbitrary Boolean Structures

Definition ($\mathcal{T}$-Solver)

Let $\mathcal{T}$ be a theory. A $\mathcal{T}$-solver is a procedure for deciding $\mathcal{T}$-consistency of a conjunction of $\mathcal{T}$-literals.

- Lifting to arbitrary boolean structure:
  1. Given $\mathcal{T}$-formula ϕ, transform ϕ into equivalent ϕ' in DNF.
  2. ϕ' is $\mathcal{T}$-consistent iff $\phi' = (L_1 \wedge \cdots \wedge L_n) \vee \phi''$ and $L_1 \wedge \cdots \wedge L_n$ is $\mathcal{T}$-consistent
- Drawback: potential expontential explosion during DNF-transformation
- Idea: Use SAT-Solver to enumerate (some/sufficiently many) disjuncts

Lifting $\mathcal{T}$-Reasoning to Arbitrary Boolean Structures

Definition ($\mathcal{T}$-Solver)

Let $\mathcal{T}$ be a theory. A $\mathcal{T}$-solver is a procedure for deciding $\mathcal{T}$-consistency of a conjunction of $\mathcal{T}$-literals.

- Lifting to arbitrary boolean structure:
  1. Given $\mathcal{T}$-formula ϕ, transform ϕ into equivalent ϕ' in DNF.
  2. ϕ' is $\mathcal{T}$-consistent iff $\phi' = (L_1 \wedge \cdots \wedge L_n) \vee \phi''$ and $L_1 \wedge \cdots \wedge L_n$ is $\mathcal{T}$-consistent
- Drawback: potential expontential explosion during DNF-transformation
- Idea: Use SAT-Solver to enumerate (some/sufficiently many) disjuncts

Notation

Associate with each $\mathcal{T}$-atom A a propositional variable $\text{atr}(A) = P_A$ and lift atr to $\mathcal{T}$-formulas.

Example (Propositional Abstractions)

$$\begin{aligned} &\text{atr}\,(\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)) \\ =&\neg P_{(2x-2y\leq 1)} \vee P_{(-2x-2y\leq -1)} \\ =&\neg P_1 \vee P_2 \end{aligned}$$

Definition (Entailments)

Let ϕ and ψ be $\mathcal{T}$-formulas.

- $\phi \models \psi$ iff each (propositional) model of $\text{atr}(\phi)$ is also a model of $\text{atr}(\psi)$;
- $\phi \models_{\mathcal{T}} \psi$ iff each $\mathcal{T}$-model of ϕ is also a $\mathcal{T}$-model of ψ.

Notation

Associate with each $\mathcal{T}$-atom A a propositional variable $\text{atr}(A) = P_A$ and lift atr to $\mathcal{T}$-formulas.

Example (Propositional Abstractions)

$$\begin{aligned} &\text{atr}\,(\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)) \\ =&\neg P_{(2x-2y\leq 1)} \vee P_{(-2x-2y\leq -1)} \\ =&\neg P_1 \vee P_2 \end{aligned}$$

Definition (Entailments)

Let ϕ and ψ be $\mathcal{T}$-formulas.

- $\phi \models \psi$ iff each (propositional) model of $\text{atr}(\phi)$ is also a model of $\text{atr}(\psi)$;
- $\phi \models_{\mathcal{T}} \psi$ iff each $\mathcal{T}$-model of ϕ is also a $\mathcal{T}$-model of ψ.

Notation (Cont.)

Example (Entailments)

$$(\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)) \wedge (2x - 2y \leq 1) \models (-2x - 2y \leq -1)$$

$$(2x - 2y \leq 1) \wedge (2x - 2y \geq 3) \models_{\mathcal{T}} \bot$$

$$(2x - 2y \leq 1) \wedge (2x - 2y \geq 3) \not\models \bot$$

Proposition (Property of Entailments)

Let $\mathcal{T}$ be a theory. Then

$$\models \subseteq \models_{\mathcal{T}}$$

Notation (Cont.)

Example (Entailments)

$$(\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)) \wedge (2x - 2y \leq 1) \models (-2x - 2y \leq -1)$$

$$(2x - 2y \leq 1) \wedge (2x - 2y \geq 3) \models_{\mathcal{T}} \bot$$

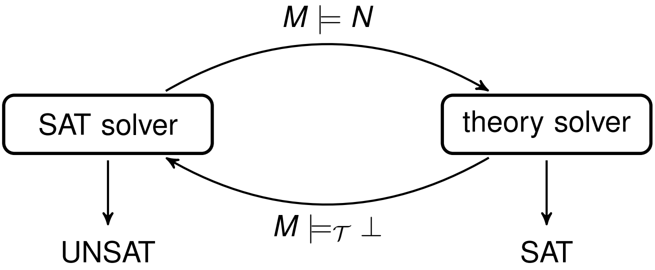
$$(2x - 2y \leq 1) \wedge (2x - 2y \geq 3) \not\models \bot$$

Proposition (Property of Entailments)

Let $\mathcal{T}$ be a theory. Then

$$\models \subseteq \models_{\mathcal{T}}$$

Naive Architecture



Example (Linear Integer Arithmetic)

$C_1 = L_1$ | $(2x - 2y \leq 1)$

$C_2 = L_2$ ↤ $(-2x + 2y \leq 1)$

$C_3 = \neg L_1 \vee L_3$ | $\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)$

$C_4 = L_4 \vee L_5$ | $(2x + 2y \leq 3) \vee (-2x + 2y \geq 2)$

$C_5 = \neg M_1$

$C_6 = \neg M_2$

$C_7 = \neg M_3$

$M_1 = L_1 L_2 L_3 L_4 L_5$

$M_2 = L_1 L_2 L_3 L_4 \neg L_5$

$M_3 = L_1 L_2 L_3 \neg L_4 L_5$

Example (Linear Integer Arithmetic)

$C_1 = L_1$ ↤ $(2x - 2y \leq 1)$

$C_2 = L_2$ ↤ $(-2x + 2y \leq 1)$

$C_3 = \neg L_1 \vee L_3$ ↤ $\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)$

$C_4 = L_4 \vee L_5$ ↤ $(2x + 2y \leq 3) \vee (-2x + 2y \geq 2)$

$C_5 = \neg M_1$

$C_6 = \neg M_2$

$C_7 = \neg M_3$

$M_1 = L_1 L_2 L_3 L_4 L_5$

$M_2 = L_1 L_2 L_3 L_4 \neg L_5$

$M_3 = L_1 L_2 L_3 \neg L_4 L_5$

Example (Linear Integer Arithmetic)

$C_1 = L_1$ ↤ $(2x - 2y \leq 1)$

$C_2 = L_2$ ↤ $(-2x + 2y \leq 1)$

$C_3 = \neg L_1 \vee L_3$ ↤ $\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)$

$C_4 = L_4 \vee L_5$ ↤ $(2x + 2y \leq 3) \vee (-2x + 2y \geq 2)$

$C_5 = \neg M_1$

$C_6 = \neg M_2$

$C_7 = \neg M_3$

$M_1 = L_1 L_2 L_3 L_4 L_5$

$M_2 = L_1 L_2 L_3 L_4 \neg L_5$

$M_3 = L_1 L_2 L_3 \neg L_4 L_5$

Example (Linear Integer Arithmetic)

$C_1 = L_1$ — $(2x - 2y \leq 1)$

$C_2 = L_2$ $\hookleftarrow$ $(-2x + 2y \leq 1)$

$C_3 = \neg L_1 \vee L_3$ — $\neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)$

$C_4 = L_4 \vee L_5$ — $(2x + 2y \leq 3) \vee (-2x + 2y \geq 2)$

$C_5 = \neg M_1$

$C_6 = \neg M_2$

$C_7 = \neg M_3$

$M_1 = L_1 L_2 L_3 L_4 L_5$

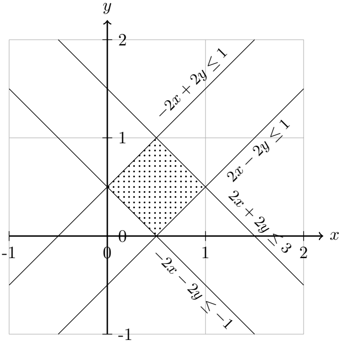
$M_2 = L_1 L_2 L_3 L_4 \neg L_5$

$M_3 = L_1 L_2 L_3 \neg L_4 L_5$

Example (Linear Integer Arithmetic)

$$C_1 = L_1 \qquad (2x - 2y \leq 1)$$
$$C_2 = L_2 \qquad (-2x + 2y \leq 1)$$
$$C_3 = \neg L_1 \vee L_3 \quad \mapsfrom \quad \neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)$$
$$C_4 = L_4 \vee L_5 \qquad (2x + 2y \leq 3) \vee (-2x + 2y \geq 2)$$
$$C_5 = \neg M_1$$
$$C_6 = \neg M_2$$
$$C_7 = \neg M_3$$

$$M_1 = L_1 L_2 L_3 L_4 L_5$$
$$M_2 = L_1 L_2 L_3 L_4 \neg L_5$$
$$M_3 = L_1 L_2 L_3 \neg L_4 L_5$$



Example (Linear Integer Arithmetic)

$$C_1 = L_1 \qquad (2x - 2y \leq 1)$$
$$C_2 = L_2 \qquad (-2x + 2y \leq 1)$$
$$C_3 = \neg L_1 \vee L_3 \qquad \hookleftarrow \qquad \neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)$$
$$C_4 = L_4 \vee L_5 \qquad (2x + 2y \leq 3) \vee (-2x + 2y \geq 2)$$
$$C_5 = \neg M_1$$
$$C_6 = \neg M_2$$
$$C_7 = \neg M_3$$

$$M_1 = L_1 L_2 L_3 L_4 L_5$$
$$M_2 = L_1 L_2 L_3 L_4 \neg L_5$$
$$M_3 = L_1 L_2 L_3 \neg L_4 L_5$$

Example (Linear Integer Arithmetic)

$$
\begin{aligned}
C_1 &= L_1 & & (2x - 2y \leq 1) \\
C_2 &= L_2 & & (-2x + 2y \leq 1) \\
C_3 &= \neg L_1 \vee L_3 & \hookleftarrow & \neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1) \\
C_4 &= L_4 \vee L_5 & & (2x + 2y \leq 3) \vee (-2x + 2y \geq 2) \\
C_5 &= \neg M_1 & & \\
C_6 &= \neg M_2 & & \\
C_7 &= \neg M_3 & &
\end{aligned}
$$

$$
\begin{aligned}
M_1 &= L_1 L_2 L_3 L_4 L_5 \\
M_2 &= L_1 L_2 L_3 L_4 \neg L_5 \\
M_3 &= L_1 L_2 L_3 \neg L_4 L_5
\end{aligned}
$$

Tighter Collaboration between SAT and Theory Solver

- Generate “small” $\mathcal{T}$-conflict clauses
- Incrementality
- Detect $\mathcal{T}$-inconsistencies early
- $\mathcal{T}$-propagations
- Case splits by learning additional clauses

Problem State

$$(\underbrace{M}_{\text{trail}};\ \underbrace{N}_{\text{problem clauses}};\ \underbrace{U}_{\text{learned clauses}};\ \underbrace{T}_{\mathcal{T}\text{-learned clauses}};\ \underbrace{k}_{\text{decision level}};\ \underbrace{D}_{\text{conflict}})$$

$(\epsilon; N; \emptyset; \emptyset; 0; \top)$	is the start state for some clause set N
$(M; N; U; T; k; \top)$	is a final state where N is $\mathcal{T}$-satisfiable if $M \models N$, $M \not\models_{\mathcal{T}} \bot$ and all literals from $N \cup U \cup T$ are defined in M.
$(M; N; U; T; k; \bot)$	is a final state, where N has no $\mathcal{T}$-model
$(M; N; U; T; k; \top)$	is an intermediate model search state if not all literals from $N \cup U \cup T$ are defined in M, $M \not\models N$ or $M \models_{\mathcal{T}} \bot$
$(M; N; U; T; k; D)$	is a backtracking state if $D \notin \{\top, \bot\}$

Problem State

$$(\underbrace{M}_{\text{trail}};\ \underbrace{N}_{\text{problem clauses}};\ \underbrace{U}_{\text{learned clauses}};\ \underbrace{T}_{\mathcal{T}\text{-learned clauses}};\ \underbrace{k}_{\text{decision level}};\ \underbrace{D}_{\text{conflict}})$$

$(\epsilon; N; \emptyset; \emptyset; 0; \top)$	is the start state for some clause set N
$(M; N; U; T; k; \top)$	is a final state where N is $\mathcal{T}$-satisfiable if $M \models N$, $M \not\models_{\mathcal{T}} \bot$ and all literals from $N \cup U \cup T$ are defined in M.
$(M; N; U; T; k; \bot)$	is a final state, where N has no $\mathcal{T}$-model
$(M; N; U; T; k; \top)$	is an intermediate model search state if not all literals from $N \cup U \cup T$ are defined in M, $M \not\models N$ or $M \models_{\mathcal{T}} \bot$
$(M; N; U; T; k; D)$	is a backtracking state if $D \notin \{\top, \bot\}$

CDCL(T) Calculus – Propositional Reasoning

Decide $(M; N; U; T; k; \top) \Rightarrow_{\text{CDCL}(\mathcal{T})} (ML^{k+1}; N; U; T; k+1; \top)$
provided L is undefined in M and $L \in \text{lits}(N \cup U \cup T)$.

Propagate $(M; N; U; T; k; \top) \Rightarrow_{\text{CDCL}(\mathcal{T})} (ML^{C \vee L}; N; U; T; k; \top)$
provided $C \vee L \in (N \cup U \cup T)$, $M \models \neg C$ and L is undefined in M.

Conflict $(M; N; U; T; k; \top) \Rightarrow_{\text{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$
provided $D \in (N \cup U \cup T)$ and $M \models \neg D$.

CDCL(T) Calculus – Propositional Reasoning (Cont.)

Skip $(ML; N; U; T; k; D) \Rightarrow_{\mathrm{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$
provided $\mathrm{comp}(L) \notin D$ and $D \notin \{\top, \bot\}$.

Resolve $(ML^{C \vee L}; N; U; T; k; D \vee \mathrm{comp}(L)) \Rightarrow_{\mathrm{CDCL}(\mathcal{T})}$
$(M; N; U; T; k; D \vee C)$
provided D and L are of the same level or $D = \bot$.

Backtrack $(M_1 K^{i+1} M_2; N; U; T; k; D \vee L) \Rightarrow_{\mathrm{CDCL}(\mathcal{T})}$
$(M_1 L^{D \vee L}; N; U \cup \{D \vee L\}; T; i; \top)$
provided L is of level k and D is of level i where $i < k$.

CDCL(T) Calculus – Theory Reasoning

$\mathcal{T}$-Conflict $(M; N; U; T; k; \top) \Rightarrow_{\mathrm{CDCL}(\mathcal{T})} (M; N; U; T; k'; D)$
provided $M \models L_1, \ldots, L_n$ (i.e. $L_1, \ldots, L_n$ occur in M), $L_1 \wedge \cdots \wedge L_n \models_{\mathcal{T}} \bot$ and $D = \mathrm{comp}(L_1) \vee \cdots \vee \mathrm{comp}(L_n)$ and D is of level k'.

$\mathcal{T}$-Propagate $(M; N; U; T; k; \top) \Rightarrow_{\mathrm{CDCL}(\mathcal{T})} (ML^{C \vee L}; N; U; T; k; \top)$
provided $M \models L_1, \ldots, L_n$ (i.e. $L_1, \ldots, L_n$ occur in M), $L_1 \wedge \cdots \wedge L_n \models_{\mathcal{T}} L$ and $L \in \mathrm{lits}(N \cup U \cup T)$, L is undefined in M and $C = \mathrm{comp}(L_1) \vee \cdots \vee \mathrm{comp}(L_n)$.

CDCL(T) Calculus – Theory Reasoning

$\mathcal{T}$-Conflict $(M; N; U; T; k; \top) \Rightarrow_{\text{CDCL}(\mathcal{T})} (M; N; U; T; k'; D)$
provided $M \models L_1, \ldots, L_n$ (i.e. $L_1, \ldots, L_n$ occur in M), $L_1 \wedge \cdots \wedge L_n \models_{\mathcal{T}} \bot$ and $D = \text{comp}(L_1) \vee \cdots \vee \text{comp}(L_n)$ and D is of level k'.

$\mathcal{T}$-Propagate $(M; N; U; T; k; \top) \Rightarrow_{\text{CDCL}(\mathcal{T})} (ML^{C \vee L}; N; U; T; k; \top)$
provided $M \models L_1, \ldots, L_n$ (i.e. $L_1, \ldots, L_n$ occur in M), $L_1 \wedge \cdots \wedge L_n \models_{\mathcal{T}} L$ and $L \in \text{lits}(N \cup U \cup T)$, L is undefined in M and $C = \text{comp}(L_1) \vee \cdots \vee \text{comp}(L_n)$.

Example (Linear Integer Arithmetic)

$$L_1 \vee L_2 = (x \geq 5) \vee (x \leq 3)$$
$$\neg L_1 \vee L_3 \vee L_4 = \neg(x \leq 5) \vee (y \geq 7) \vee (y \leq 4)$$
$$\neg L_1 \vee L_5 \vee L_6 = \neg(x \geq 5) \vee (y \leq 6) \vee (x + y \leq 4)$$

$$(\epsilon; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^1; N; \emptyset; \emptyset; 1; \top)$$
$$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2}; N; \emptyset; \emptyset; 1; \top)$$
$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2; N; \emptyset; \emptyset; 2; \top)$$
$$\Rightarrow^{T\text{-Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2 \neg L_5^{\neg L_3 \vee \neg L_5}; N; \emptyset; \emptyset; 2; \top)$$
$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2 \neg L_5^{\neg L_3 \vee \neg L_5} L_6^{\neg L_1 \vee L_5 \vee L_6}; N; \emptyset; \emptyset; 2; \top)$$

Example (Linear Integer Arithmetic)

$$L_1 \vee L_2 = (x \geq 5) \vee (x \leq 3)$$
$$\neg L_1 \vee L_3 \vee L_4 = \neg(x \leq 5) \vee (y \geq 7) \vee (y \leq 4)$$
$$\neg L_1 \vee L_5 \vee L_6 = \neg(x \geq 5) \vee (y \leq 6) \vee (x + y \leq 4)$$

$$(\epsilon; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^1; N; \emptyset; \emptyset; 1; \top)$$
$$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2}; N; \emptyset; \emptyset; 1; \top)$$
$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2; N; \emptyset; \emptyset; 2; \top)$$
$$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2 \neg L_5^{\neg L_3 \vee \neg L_5}; N; \emptyset; \emptyset; 2; \top)$$
$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2 \neg L_5^{\neg L_3 \vee \neg L_5} L_6^{\neg L_1 \vee L_5 \vee L_6}; N; \emptyset; \emptyset; 2; \top)$$

Example (Linear Integer Arithmetic)

$$L_1 \vee L_2 = (x \geq 5) \vee (x \leq 3)$$
$$\neg L_1 \vee L_3 \vee L_4 = \neg(x \leq 5) \vee (y \geq 7) \vee (y \leq 4)$$
$$\neg L_1 \vee L_5 \vee L_6 = \neg(x \geq 5) \vee (y \leq 6) \vee (x + y \leq 4)$$

$$(\epsilon; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^1; N; \emptyset; \emptyset; 1; \top)$$
$$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2}; N; \emptyset; \emptyset; 1; \top)$$
$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2; N; \emptyset; \emptyset; 2; \top)$$
$$\Rightarrow^{T\text{-Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2 \neg L_5^{\neg L_3 \vee \neg L_5}; N; \emptyset; \emptyset; 2; \top)$$
$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2 \neg L_5^{\neg L_3 \vee \neg L_5} L_6^{\neg L_1 \vee L_5 \vee L_6}; N; \emptyset; \emptyset; 2; \top)$$

Example (Cont.)(Linear Integer Arithmetic)

$$L_1 \vee L_2 = (x \geq 5) \vee (x \leq 3)$$
$$\neg L_1 \vee L_3 \vee L_4 = \neg(x \leq 5) \vee (y \geq 7) \vee (y \leq 4)$$
$$\neg L_1 \vee L_5 \vee L_6 = \neg(x \geq 5) \vee (y \leq 6) \vee (x + y \leq 4)$$

$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2 \neg L_5^{\neg L_3 \vee \neg L_5} L_6^{\neg L_1 \vee L_5 \vee L_6}; N; \emptyset; \emptyset; 2; \neg L_1 \vee \neg L_3 \vee \neg L_6)$

$\Rightarrow^{\text{Resolve}^*}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2; N; \emptyset; \emptyset; 2; \neg L_1 \vee \neg L_3)$

$\Rightarrow^{\text{Backtrack}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3}; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3} L_4^{\neg L_1 \vee L_3 \vee L_4}; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3} L_4^{\neg L_1 \vee L_3 \vee L_4} L_5^{\neg L_4 \vee L_5}; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3} L_4^{\neg L_1 \vee L_3 \vee L_4} L_5^{\neg L_4 \vee L_5} \neg L_6^2; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 2; \top)$

Example (Cont.) (Linear Integer Arithmetic)

$$L_1 \vee L_2 = (x \geq 5) \vee (x \leq 3)$$
$$\neg L_1 \vee L_3 \vee L_4 = \neg(x \leq 5) \vee (y \geq 7) \vee (y \leq 4)$$
$$\neg L_1 \vee L_5 \vee L_6 = \neg(x \geq 5) \vee (y \leq 6) \vee (x + y \leq 4)$$

$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2 \neg L_5^{\neg L_3 \vee \neg L_5} L_6^{\neg L_1 \vee L_5 \vee L_6}; N; \emptyset; \emptyset; 2; \neg L_1 \vee \neg L_3 \vee \neg L_6)$

$\Rightarrow^{\text{Resolve}^*}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2; N; \emptyset; \emptyset; 2; \neg L_1 \vee \neg L_3)$

$\Rightarrow^{\text{Backtrack}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3}; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3} L_4^{\neg L_1 \vee L_3 \vee L_4}; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3} L_4^{\neg L_1 \vee L_3 \vee L_4} L_5^{\neg L_4 \vee L_5}; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3} L_4^{\neg L_1 \vee L_3 \vee L_4} L_5^{\neg L_4 \vee L_5} \neg L_6^2; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 2; \top)$

Example (Cont.) (Linear Integer Arithmetic)

$$L_1 \lor L_2 = (x \geq 5) \lor (x \leq 3)$$
$$\neg L_1 \lor L_3 \lor L_4 = \neg(x \leq 5) \lor (y \geq 7) \lor (y \leq 4)$$
$$\neg L_1 \lor L_5 \lor L_6 = \neg(x \geq 5) \lor (y \leq 6) \lor (x + y \leq 4)$$

$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \lor \neg L_2} L_3^2 \neg L_5^{\neg L_3 \lor \neg L_5} L_6^{\neg L_1 \lor L_5 \lor L_6}; N; \emptyset; \emptyset; 2; \neg L_1 \lor \neg L_3 \lor \neg L_6)$

$\Rightarrow^{\text{Resolve}^*}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \lor \neg L_2} L_3^2; N; \emptyset; \emptyset; 2; \neg L_1 \lor \neg L_3)$

$\Rightarrow^{\text{Backtrack}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \lor \neg L_2} \neg L_3^{\neg L_1 \lor \neg L_3}; N; \{\neg L_1 \lor \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \lor \neg L_2} \neg L_3^{\neg L_1 \lor \neg L_3} L_4^{\neg L_1 \lor L_3 \lor L_4}; N; \{\neg L_1 \lor \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \lor \neg L_2} \neg L_3^{\neg L_1 \lor \neg L_3} L_4^{\neg L_1 \lor L_3 \lor L_4} L_5^{\neg L_4 \lor L_5}; N; \{\neg L_1 \lor \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \lor \neg L_2} \neg L_3^{\neg L_1 \lor \neg L_3} L_4^{\neg L_1 \lor L_3 \lor L_4} L_5^{\neg L_4 \lor L_5} \neg L_6^2; N; \{\neg L_1 \lor \neg L_3\}; \emptyset; 2; \top)$

Example (Cont.)(Linear Integer Arithmetic)

$$L_1 \vee L_2 = (x \geq 5) \vee (x \leq 3)$$
$$\neg L_1 \vee L_3 \vee L_4 = \neg(x \leq 5) \vee (y \geq 7) \vee (y \leq 4)$$
$$\neg L_1 \vee L_5 \vee L_6 = \neg(x \geq 5) \vee (y \leq 6) \vee (x + y \leq 4)$$

$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2 \neg L_5^{\neg L_3 \vee \neg L_5} L_6^{\neg L_1 \vee L_5 \vee L_6}; N; \emptyset; \emptyset; 2; \neg L_1 \vee \neg L_3 \vee \neg L_6)$

$\Rightarrow^{\text{Resolve}^*}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} L_3^2; N; \emptyset; \emptyset; 2; \neg L_1 \vee \neg L_3)$

$\Rightarrow^{\text{Backtrack}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3}; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3} L_4^{\neg L_1 \vee L_3 \vee L_4}; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3} L_4^{\neg L_1 \vee L_3 \vee L_4} L_5^{\neg L_4 \vee L_5}; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 1; \top)$

$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^1 \neg L_2^{\neg L_1 \vee \neg L_2} \neg L_3^{\neg L_1 \vee \neg L_3} L_4^{\neg L_1 \vee L_3 \vee L_4} L_5^{\neg L_4 \vee L_5} \neg L_6^2; N; \{\neg L_1 \vee \neg L_3\}; \emptyset; 2; \top)$

CDCL(T) Calculus – Splitting on Demand

- Problem: Solvers for many theories need to do case splits
- Idea: Use SAT Solver for case splits
  - Encode splits as clauses,
  - Reuse advanced backtracking techniques of CDCL for free,
  - Avoid re-implementing them in (several) theory solvers.

$\mathcal{T}$-Learn $(M; N; U; T; k; \top) \Rightarrow_{\text{CDCL}(\mathcal{T})} (M; N; U; T \uplus T'; k; \top)$
provided $(N \cup U \cup T) \models_{\mathcal{T}} T'$, $T' \cap (N \cup U \cup T) = \emptyset$, T' is finite.

- Potential disadvantages: may introduce a huge number of clauses that are used infrequently

$\mathcal{T}$-Forget $(M; N; U; T \uplus T'; k; D) \Rightarrow_{\text{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$
provided $D \notin \{\top, \bot\}$, $T' \neq \emptyset$ and
$\text{atoms}(M) \subseteq \text{atoms}(N \cup U \cup T)$.

CDCL(T) Calculus – Splitting on Demand

- Problem: Solvers for many theories need to do case splits
- Idea: Use SAT Solver for case splits
  - Encode splits as clauses,
  - Reuse advanced backtracking techniques of CDCL for free,
  - Avoid re-implementing them in (several) theory solvers.

$\mathcal{T}$-Learn $(M; N; U; T; k; \top) \Rightarrow_{\text{CDCL}(\mathcal{T})} (M; N; U; T \uplus T'; k; \top)$
provided $(N \cup U \cup T) \models_{\mathcal{T}} T'$, $T' \cap (N \cup U \cup T) = \emptyset$, T' is finite.

- Potential disadvantages: may introduce a huge number of clauses that are used infrequently

$\mathcal{T}$-Forget $(M; N; U; T \uplus T'; k; D) \Rightarrow_{\text{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$
provided $D \notin \{\top, \bot\}$, $T' \neq \emptyset$ and
atoms$(M) \subseteq$ atoms$(N \cup U \cup T)$.

Example (Linear Integer Arithmetic)

$$L_1 = (2x - 2y \leq 1)$$
$$L_2 = (-2x + 2y \leq 1)$$
$$\neg L_1 \vee L_3 = \neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)$$
$$L_4 \vee L_5 = (2x + 2y \leq 3) \vee (-2x + 2y \geq 2)$$

$$(\epsilon; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1}; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Propagate*}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2}; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5}; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4}; N; \emptyset; \emptyset; 0; \top)$$

Example (Linear Integer Arithmetic)

$$L_1 = (2x - 2y \leq 1)$$
$$L_2 = (-2x + 2y \leq 1)$$
$$\neg L_1 \vee L_3 = \neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)$$
$$L_4 \vee L_5 = (2x + 2y \leq 3) \vee (-2x + 2y \geq 2)$$

$$(\epsilon; N; \emptyset; \emptyset; 0; \top)$$

$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1}; N; \emptyset; \emptyset; 0; \top)$

$\Rightarrow^{\text{Propagate*}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2}; N; \emptyset; \emptyset; 0; \top)$

$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5}; N; \emptyset; \emptyset; 0; \top)$

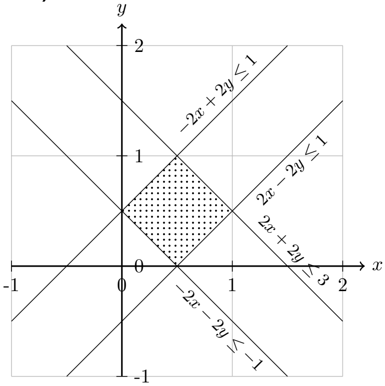
$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4}; N; \emptyset; \emptyset; 0; \top)$

Example (Linear Integer Arithmetic)

$$L_1 = (2x - 2y \leq 1)$$
$$L_2 = (-2x + 2y \leq 1)$$
$$\neg L_1 \vee L_3 = \neg(2x - 2y \leq 1) \vee (-2x - 2y \leq -1)$$
$$L_4 \vee L_5 = (2x + 2y \leq 3) \vee (-2x + 2y \geq 2)$$

$$(\epsilon; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1}; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Propagate*}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2}; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\mathcal{T}\text{-Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5}; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4}; N; \emptyset; \emptyset; 0; \top)$$

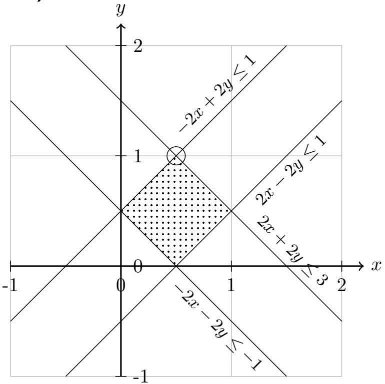
Example (Cont.)



$$L_6 \vee L_7 = (x \leq 0) \vee (x \geq 1)$$

$$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} \quad (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4}; N; \emptyset; \{L_6 \vee L_7\}; 0; \top)$$

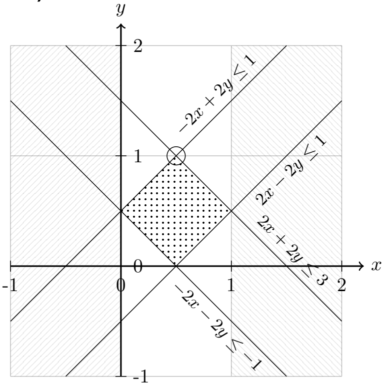
Example (Cont.)



$$L_6 \vee L_7 = (x \leq 0) \vee (x \geq 1)$$

$$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} \quad (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4}; N; \emptyset; \{L_6 \vee L_7\}; 0; \top)$$

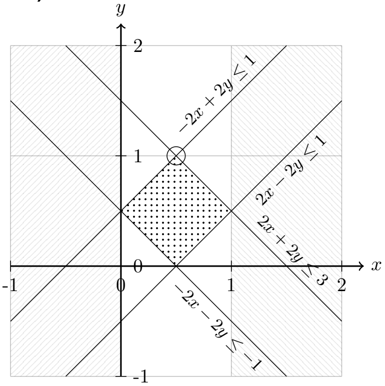
Example (Cont.)



$$L_6 \vee L_7 = (x \leq 0) \vee (x \geq 1)$$

$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} \quad (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4}; N; \emptyset; \{L_6 \vee L_7\}; 0; \top)$

Example (Cont.)



$$L_6 \vee L_7 = (x \leq 0) \vee (x \geq 1)$$

$$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} \quad (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4}; N; \emptyset; \{L_6 \vee L_7\}; 0; \top)$$

Example (Cont.)

$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} L_6^{1}; N; \emptyset; \{L_6 \vee L_7\}; 1; \top)$

$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} L_6^{1};$
$N; \emptyset; \{L_6 \vee L_7\}; 1; \neg L_2 \vee \neg L_3 \vee \neg L_6)$

$\Rightarrow^{\text{Backtrack}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} (\neg L_6)^{\neg L_2 \vee \neg L_3 \vee \neg L_6};$
$N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \top)$

$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} (\neg L_6)^{\neg L_2 \vee \neg L_3 \vee \neg L_6} L_7^{L_6 \vee L_7}$
$N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \top)$

$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} (\neg L_6)^{\neg L_2 \vee \neg L_3 \vee \neg L_6} L_7^{L_6 \vee L_7}$
$N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \neg L_1 \vee \neg L_4 \vee \neg L_7)$

$\Rightarrow^{\text{Resolve*}}_{\text{CDCL}(\mathcal{T})}$ $(\epsilon; N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \bot)$

Example (Cont.)

$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} \; (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} L_6^{1}; N; \emptyset; \{L_6 \vee L_7\}; 1; \top)$$

$$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})} \; (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} L_6^{1};$$
$$N; \emptyset; \{L_6 \vee L_7\}; 1; \neg L_2 \vee \neg L_3 \vee \neg L_6)$$

$$\Rightarrow^{\text{Backtrack}}_{\text{CDCL}(\mathcal{T})} \; (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} (\neg L_6)^{\neg L_2 \vee \neg L_3 \vee \neg L_6};$$
$$N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \top)$$

$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} \; (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} (\neg L_6)^{\neg L_2 \vee \neg L_3 \vee \neg L_6} L_7^{L_6 \vee L_7}$$
$$N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \top)$$

$$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})} \; (L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} (\neg L_6)^{\neg L_2 \vee \neg L_3 \vee \neg L_6} L_7^{L_6 \vee L_7}$$
$$N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \neg L_1 \vee \neg L_4 \vee \neg L_7)$$

$$\Rightarrow^{\text{Resolve}^*}_{\text{CDCL}(\mathcal{T})} \; (\epsilon; N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \bot)$$

Example (Cont.)

$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} L_6^1; N; \emptyset; \{L_6 \vee L_7\}; 1; \top)$

$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} L_6^1;$
$N; \emptyset; \{L_6 \vee L_7\}; 1; \neg L_2 \vee \neg L_3 \vee \neg L_6)$

$\Rightarrow^{\text{Backtrack}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} (\neg L_6)^{\neg L_2 \vee \neg L_3 \vee \neg L_6};$
$N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \top)$

$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} (\neg L_6)^{\neg L_2 \vee \neg L_3 \vee \neg L_6} L_7^{L_6 \vee L_7}$
$N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \top)$

$\Rightarrow^{\mathcal{T}\text{-Conflict}}_{\text{CDCL}(\mathcal{T})}$ $(L_1^{L_1} L_3^{\neg L_1 \vee L_3} L_2^{L_2} (\neg L_5)^{\neg L_2 \vee \neg L_5} L_4^{L_5 \vee L_4} (\neg L_6)^{\neg L_2 \vee \neg L_3 \vee \neg L_6} L_7^{L_6 \vee L_7}$
$N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \neg L_1 \vee \neg L_4 \vee \neg L_7)$

$\Rightarrow^{\text{Resolve*}}_{\text{CDCL}(\mathcal{T})}$ $(\epsilon; N; \{\neg L_2 \vee \neg L_3 \vee \neg L_6\}; \{L_6 \vee L_7\}; 0; \bot)$

Lemma (Invariants I)

Let $(\epsilon; N_0; \emptyset; \emptyset; 0; \top) \Rightarrow^*_{\text{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$. Then:

1. $N = N_0$;
2. M is (propositionally) consistent, i.e. it does not contain a literal L as well as comp(L);
3. M does not contain the same literal twice;
4. Decision literal annotations are ordered in a strictly increasing manner on the trail and k is equal to the maximal annotation unless $D \notin \{\top, \bot\}$ in which case k is greater or equal to the maximal level on the trail and equal to the level of D;

Proof.

Induction on the length of the derivation. □

Lemma (Invariants I)

Let $(\epsilon; N_0; \emptyset; \emptyset; 0; \top) \Rightarrow^*_{\text{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$. Then:

1. $N = N_0$;
2. M is (propositionally) consistent, i.e. it does not contain a literal L as well as comp(L);
3. M does not contain the same literal twice;
4. Decision literal annotations are ordered in a strictly increasing manner on the trail and k is equal to the maximal annotation unless $D \notin \{\top, \bot\}$ in which case k is greater or equal to the maximal level on the trail and equal to the level of D;

Proof.

Induction on the length of the derivation. □

Lemma (Invariants II)

Let $(\epsilon; N_0; \emptyset; \emptyset; 0; \top) \Rightarrow^*_{\text{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$. Then:

1. both if $M = M_1 L^{C \vee L} M_2$ then $M_1 \models \neg C$, and if $D \notin \{\bot, \top\}$ then $M \models \neg D$;
2. $N \models_{\mathcal{T}} (U \cup T)$, $N \models_{\mathcal{T}} D$ and if $M = M_1 L^{C \vee L} M_2$ then $N \models_{\mathcal{T}} C \vee L$.
3. $\text{lits}(D) \subseteq \text{lits}(N \cup U \cup T)$, $\text{lits}(M) \subseteq \text{lits}(N \cup U \cup T)$ and if $M = M_1 L^{C \vee L} M_2$ then $\text{lits}(C \vee L) \subseteq \text{lits}(N \cup U \cup T)$.
4. U and T are finite if N_0 is finite;

Proof.

Induction on the length of the derivation. □

Lemma (Invariants II)

Let $(\epsilon; N_0; \emptyset; \emptyset; 0; \top) \Rightarrow^*_{\mathrm{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$. Then:

1. both if $M = M_1 L^{C\vee L} M_2$ then $M_1 \models \neg C$, and if $D \notin \{\bot, \top\}$ then $M \models \neg D$;
2. $N \models_{\mathcal{T}} (U \cup T)$, $N \models_{\mathcal{T}} D$ and if $M = M_1 L^{C\vee L} M_2$ then $N \models_{\mathcal{T}} C \vee L$.
3. $\mathrm{lits}(D) \subseteq \mathrm{lits}(N \cup U \cup T)$, $\mathrm{lits}(M) \subseteq \mathrm{lits}(N \cup U \cup T)$ and if $M = M_1 L^{C\vee L} M_2$ then $\mathrm{lits}(C \vee L) \subseteq \mathrm{lits}(N \cup U \cup T)$.
4. U and T are finite if N_0 is finite;

Proof.

Induction on the length of the derivation. □

Soundness

Proposition (Soundness)

Let $(\epsilon; N_0; \emptyset; \emptyset; 0; \top) \Rightarrow^*_{\text{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$ be terminal. Then exactly one of the following holds:

1. $D = \bot$ and N_0 is $\mathcal{T}$-unsatisfiable;
2. $D = \top$ and N_0 is $\mathcal{T}$-satisfiable.

What about termination?

Soundness

Proposition (Soundness)

Let $(\epsilon; N_0; \emptyset; \emptyset; 0; \top) \Rightarrow^*_{\mathrm{CDCL}(\mathcal{T})} (M; N; U; T; k; D)$ be terminal. Then exactly one of the following holds:

1. $D = \bot$ and N_0 is $\mathcal{T}$-unsatisfiable;
2. $D = \top$ and N_0 is $\mathcal{T}$-satisfiable.

What about termination?

Strategy and Learning Clauses Twice

Definition (Weakly Reasonable Strategy)

A strategy is called *weakly reasonable* if Propagate is preferred over Decide.

Lemma (Learning Twice)

CDCL($\mathcal{T}$) never learns the same clause twice with Backtrack when using a weakly reasonable strategy.

$\mathcal{T}$-Learn can introduce an infinite number of new literals.

Strategy and Learning Clauses Twice

Definition (Weakly Reasonable Strategy)

A strategy is called *weakly reasonable* if Propagate is preferred over Decide.

Lemma (Learning Twice)

CDCL($\mathcal{T}$) never learns the same clause twice with Backtrack when using a weakly reasonable strategy.

$\mathcal{T}$-Learn can introduce an infinite number of new literals.

Strategy and Learning Clauses Twice

Definition (Weakly Reasonable Strategy)

A strategy is called *weakly reasonable* if Propagate is preferred over Decide.

Lemma (Learning Twice)

CDCL($\mathcal{T}$) never learns the same clause twice with Backtrack when using a weakly reasonable strategy.

$\mathcal{T}$-Learn can introduce an infinite number of new literals.

Consider the clause set given by

$$L_1 = (0 \leq x - 1) \qquad L_2 = (x \leq 0)$$

Let $K_i = (x \leq i)$ for $i \in \mathbb{N}$.

$$(\epsilon; N; \emptyset; \emptyset; 0; \top)$$

$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1}; N; \emptyset; \emptyset; 0; \top)$$

$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2}; N; \emptyset; \emptyset; 0; \top)$$

$$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2}; N; \emptyset; \{K_1 \vee \neg K_1\}; 0; \top)$$

$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2} K_1^1; N; \emptyset; \{K_1 \vee \neg K_1\}; 1; \top)$$

$$\Rightarrow^{*}_{\text{CDCL}(\mathcal{T})} \cdots$$

$$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2} K_1^1 \cdots K_{i-1}^{i-1}; N; \emptyset; \{K_1 \vee \neg K_1, \ldots, K_i \vee \neg K_i\}; i-1; \top)$$

$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2} K_1^1 \cdots K_{i-1}^{i-1} K_i^i; N; \emptyset; \{K_1 \vee \neg K_1, \ldots, K_i \vee \neg K_i\}; i; \top)$$

$$\Rightarrow^{*}_{\text{CDCL}(\mathcal{T})} \cdots$$

Consider the clause set given by

$$L_1 = (0 \leq x - 1) \qquad L_2 = (x \leq 0)$$

Let $K_i = (x \leq i)$ for $i \in \mathbb{N}$.

$$(\epsilon; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1}; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2}; N; \emptyset; \emptyset; 0; \top)$$
$$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2}; N; \emptyset; \{K_1 \vee \neg K_1\}; 0; \top)$$
$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2} K_1^1; N; \emptyset; \{K_1 \vee \neg K_1\}; 1; \top)$$
$$\Rightarrow^{*}_{\text{CDCL}(\mathcal{T})} \cdots$$
$$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2} K_1^1 \cdots K_{i-1}^{i-1}; N; \emptyset; \{K_1 \vee \neg K_1, \ldots, K_i \vee \neg K_i\}; i-1; \top)$$
$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2} K_1^1 \cdots K_{i-1}^{i-1} K_i^i; N; \emptyset; \{K_1 \vee \neg K_1, \ldots, K_i \vee \neg K_i\}; i; \top)$$
$$\Rightarrow^{*}_{\text{CDCL}(\mathcal{T})} \cdots$$

Consider the clause set given by

$$L_1 = (0 \le x - 1) \qquad L_2 = (x \le 0)$$

Let $K_i = (x \le i)$ for $i \in \mathbb{N}$.

$$(\epsilon; N; \emptyset; \emptyset; 0; \top)$$

$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1}; N; \emptyset; \emptyset; 0; \top)$$

$$\Rightarrow^{\text{Propagate}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2}; N; \emptyset; \emptyset; 0; \top)$$

$$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2}; N; \emptyset; \{K_1 \lor \neg K_1\}; 0; \top)$$

$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2} K_1^1; N; \emptyset; \{K_1 \lor \neg K_1\}; 1; \top)$$

$$\Rightarrow^{*}_{\text{CDCL}(\mathcal{T})} \cdots$$

$$\Rightarrow^{\mathcal{T}\text{-Learn}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2} K_1^1 \cdots K_{i-1}^{i-1}; N; \emptyset; \{K_1 \lor \neg K_1, \dots, K_i \lor \neg K_i\}; i - 1; \top)$$

$$\Rightarrow^{\text{Decide}}_{\text{CDCL}(\mathcal{T})} (L_1^{L_1} L_2^{L_2} K_1^1 \cdots K_{i-1}^{i-1} K_i^i; N; \emptyset; \{K_1 \lor \neg K_1, \dots, K_i \lor \neg K_i\}; i; \top)$$

$$\Rightarrow^{*}_{\text{CDCL}(\mathcal{T})} \cdots$$

Termination of CDCL(T)

Straight-forward fix by [Barrett et al., 2006]:

Theorem (Termination)

Let $\mathcal{L}(N)$ be a finite set.
Then CDCL($\mathcal{T}$) terminates when using a weakly reasonable strategy such that whenever $\mathcal{T}$-learning the clauses in T', atoms(T') $\subseteq \mathcal{L}$(atoms(N)) holds.

Proof.

The well-founded measure

$$\mu'(M; N; U; T; D) = \begin{cases} (3^n - |U|, 1, n - |M|, 3^n - |T|) & \text{if } D = \top \\ (3^n - |U|, 0, |M|, |T|) & \text{otherwise} \end{cases}$$

for $n = |\mathcal{L}(\text{atoms}(N))|$ is decreased by each rule. □

Termination of CDCL(T)

Straight-forward fix by [Barrett et al., 2006]:

Theorem (Termination)

Let $\mathcal{L}(N)$ be a finite set.
Then CDCL($\mathcal{T}$) terminates when using a weakly reasonable strategy such that whenever $\mathcal{T}$-learning the clauses in T', $\text{atoms}(T') \subseteq \mathcal{L}(\text{atoms}(N))$ holds.

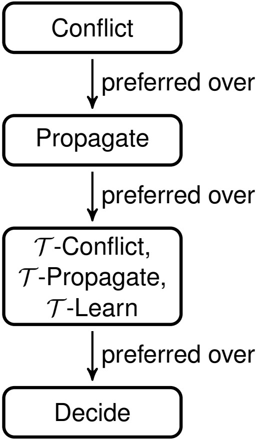
Proof.

The well-founded measure

$$\mu'(M; N; U; T; D) = \begin{cases} (3^n - |U|, 1, n - |M|, 3^n - |T|) & \text{if } D = \top \\ (3^n - |U|, 0, |M|, |T|) & \text{otherwise} \end{cases}$$

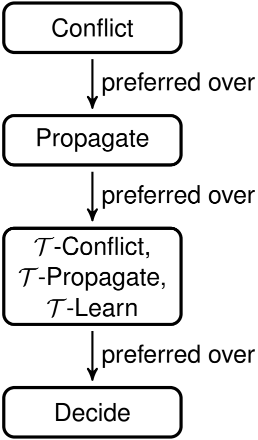
for $n = |\mathcal{L}(\text{atoms}(N))|$ is decreased by each rule. □

Strategy for CDCL(LRA)



In general: trade-off between pruning of propositional search and computational cost of theory solver calls

Strategy for CDCL(LRA)



In general: trade-off between pruning of propositional search and computational cost of theory solver calls

Improvements

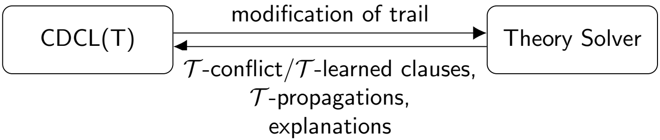
- Layered theory solvers and incomplete checks (e.g. for LIA: relaxation over the reals)
- Lazy computation of $\mathcal{T}$-explanations
- Restart, Forget
- Preprocessing
  - Normalization of $\mathcal{T}$-atoms
  - Static learning
- Redundancy
  - SAT-level redundancy
  - LIA/LRA-specific redundancy

Improvements

- Layered theory solvers and incomplete checks (e.g. for LIA: relaxation over the reals)
- Lazy computation of $\mathcal{T}$-explanations
- Restart, Forget
- Preprocessing
  - Normalization of $\mathcal{T}$-atoms
  - Static learning
- Redundancy
  - SAT-level redundancy
  - LIA/LRA-specific redundancy

Interface

- Frequent $\mathcal{T}$-solver calls with similar trails
- Support efficient addition and removal of $\mathcal{T}$-literals (*incremental* and *backtrackable* $\mathcal{T}$-solver)



Conclusion

- CDCL($\mathcal{T}$): by far most widely used calculus to decide satisfiability of (quantifier-free) formulas w.r.t. a background theory
- CDCL($\mathcal{T}$) lifts theory solvers for conjunctions of literals to (quantifier-free) formulas of an arbitrary structure
- CDCL($\mathcal{T}$) extends propositional CDCL with rules for theory reasoning based on the current trail
- Splitting on demand can be used to avoid case splits in theory solvers
- In practice: trade-off between pruning of propositional search and computational cost of theory solver calls
- Motivates incremental theory solvers

Conclusion

- CDCL($\mathcal{T}$): by far most widely used calculus to decide satisfiability of (quantifier-free) formulas w.r.t. a background theory
- CDCL($\mathcal{T}$) lifts theory solvers for conjunctions of literals to (quantifier-free) formulas of an arbitrary structure
- CDCL($\mathcal{T}$) extends propositional CDCL with rules for theory reasoning based on the current trail
- Splitting on demand can be used to avoid case splits in theory solvers
- In practice: trade-off between pruning of propositional search and computational cost of theory solver calls
- Motivates incremental theory solvers

References I

- Barrett, C., Conway, C. L., Deters, M., Hadarean, L., Jovanović, D., King, T., Reynolds, A., and Tinelli, C. (2011).
  CVC4, pages 171–177.
  Springer Berlin Heidelberg, Berlin, Heidelberg.

- Barrett, C., Fontaine, P., and Tinelli, C. (2016).
  The Satisfiability Modulo Theories Library (SMT-LIB).
  `www.SMT-LIB.org`.

- Barrett, C., Nieuwenhuis, R., Oliveras, A., and Tinelli, C. (2006).
  Splitting on Demand in SAT Modulo Theories, pages 512–526.
  Springer Berlin Heidelberg, Berlin, Heidelberg.

References II

Barrett, C. W., Sebastiani, R., Seshia, S. A., and Tinelli, C. (2009).
Satisfiability modulo theories.
In Biere, A., Heule, M., van Maaren, H., and Walsh, T., editors, *Handbook of Satisfiability*, pages 825–885. IOS Press.

Bozzano, M., Bruttomesso, R., Cimatti, A., Junttila, T. A., van Rossum, P., Schulz, S., and Sebastiani, R. (2005).
MathSAT: Tight integration of SAT and mathematical decision procedures.
J. Autom. Reasoning, 35(1-3):265–293.

References III

Cimatti, A., Griggio, A., Schaafsma, B. J., and Sebastiani, R. (2013).
The MathSAT5 SMT Solver, pages 93–107.
Springer Berlin Heidelberg, Berlin, Heidelberg.

Dutertre, B. and de Moura, L. (2006).
A Fast Linear-Arithmetic Solver for DPLL(T), pages 81–94.
Springer Berlin Heidelberg, Berlin, Heidelberg.

Kroening, D. and Strichman, O. (2016).
Decision Procedures - An Algorithmic Point of View, Second Edition.
Texts in Theoretical Computer Science. An EATCS Series.
Springer.

References IV

Nieuwenhuis, R., Oliveras, A., and Tinelli, C. (2006).
Solving SAT and SAT modulo theories: From an abstract davis–putnam–logemann–loveland procedure to dpll(T).
J. ACM, 53(6):937–977.

Sebastiani, R. (2007).
Lazy satisability modulo theories.
JSAT, 3(3-4):141–224.

Termination of CDCL(T) with Weaker Assumptions

Definition (Strongly Superset-Terminating Relations)

A strict ordering $\prec$ on $\mathcal{P}(A(\Sigma))$ is called *strongly superset-terminating* if $\prec \cap \supset$ is well-founded and for all $A, A', B \subseteq A(\Sigma)$,

1. if $A \preceq B$ and $B \subseteq A' \subseteq A$ then $A' \preceq B$;
2. if $A \preceq B$, $A' \preceq B$ and $A, A' \supseteq B$ then $(A \cup A') \preceq A$.

Theorem (Termination II)

Let $\prec$ be a strongly superset-terminating relation. Then CDCL($\mathcal{T}$) terminates when using a weakly reasonable strategy such that whenever $\mathcal{T}$-learning the clauses in T', $\text{atoms}(T' \cup N \cup U) \preceq \text{atoms}(N \cup U)$ holds.

Termination of CDCL(T) with Weaker Assumptions

Definition (Strongly Superset-Terminating Relations)

A strict ordering $\prec$ on $\mathcal{P}(A(\Sigma))$ is called *strongly superset-terminating* if $\prec \cap \supset$ is well-founded and for all $A, A', B \subseteq A(\Sigma)$,

1. if $A \preceq B$ and $B \subseteq A' \subseteq A$ then $A' \preceq B$;
2. if $A \preceq B$, $A' \preceq B$ and $A, A' \supseteq B$ then $(A \cup A') \preceq A$.

Theorem (Termination II)

Let $\prec$ be a strongly superset-terminating relation.
Then CDCL($\mathcal{T}$) terminates when using a weakly reasonable strategy such that whenever $\mathcal{T}$-learning the clauses in T', $\text{atoms}(T' \cup N \cup U) \preceq \text{atoms}(N \cup U)$ holds.

Discussion

- Our criterion is equivalent to the one of [Barrett et al., 2006] for deterministic theory solvers.
- Consider a procedure that first guesses a bound for an integer a variable and then refines it.
  - No *a priori* finite set of atoms of for $\mathcal{T}$-learning
  - However, there is an appropriate strongly superset-terminating relation

Discussion

- Our criterion is equivalent to the one of [Barrett et al., 2006] for deterministic theory solvers.
- Consider a procedure that first guesses a bound for an integer a variable and then refines it.
  - No *a priori* finite set of atoms of for $\mathcal{T}$-learning
  - However, there is an appropriate strongly superset-terminating relation

Algorithm 1: CDCL($\mathcal{T}$)(S)

Input : An initial state ($\epsilon; N; \emptyset; 0; \top$).
Output: A final state $S = (M; N; U; k; D)$, $D \in \{\top, \bot\}$

```
for (L ∈ atoms(N)) do
    T-Solver_Inform(L);
while (any rule applicable) do
    ifrule (Conflict(S)) then
        S = Analyze(S);
    else ifrule (Propagate(S)) then
        T-Solver_Assert(L);
    else
        T-Solver_IncompleteCheck(M);
        S = ReactToT-Solver(S);
        if (T-Solver failed or found model for M)
         then
            if (M |= N or complete check heuristic)
             then
                T-Solver_CompleteCheck(M);
                S = ReactToT-Solver(S);
                if (T-Solver found model for M) then
                    return(S);
            else
                Decide(S);
                T-Solver_AddDecision(L);
                T-Solver_Assert(L);
return(S);
```

Algorithm 2: ReactTo$\mathcal{T}$-Solver

Input : A state ($M; N; U; k; \top$).
Output: A state $S = (M'; N; U'; k'; D)$, $D \in \{\top, \bot\}$

```
if (detected T-Conflict) then
    C = T-Solver_GetConflict(S);
    T-Conflict'(S, C);
    S = Analyze(S);
else if (detected T-Propagations) then
    L_1, ..., L_n = T-Solver_GetPropagations(S);
    T-Propagate(S, L);
    T-Solver_Assert(L);
else if (decided to learn clauses in T') then
    T' = T-Solver_GetLearnedClauses(S);
    T-Learn(S, T');
return(S);
```

Algorithm 3: Analyze(S)

Input : A state ($M; N; U; k; D$) with $D \notin \{\top, \bot\}$.
Output: A state $S = (M'; N; U'; k'; D)$, $D \in \{\top, \bot\}$

```
whilerule (Skip(S) or Resolve(S)) do
    ;
if (T-forget heuristic) then
    T-Forget(S, N');
ifrule (Backtrack(S)) then
    T-Solver.backtrack(k);
return(S);
```

Algorithm 1: CDCL($\mathcal{T}$)(S)

Input : An initial state $(\epsilon; N; \emptyset; 0; \top)$.
Output: A final state $S = (M; N; U; k; D)$, $D \in \{\top, \bot\}$

for ($L \in$ atoms(N)) **do**
  $\mathcal{T}$-Solver_Inform(L);
while (*any rule applicable*) **do**
  ifrule (**Conflict**(S)) **then**
    S = Analyze(S);
  else ifrule (**Propagate**(S)) **then**
    $\mathcal{T}$-Solver_Assert(L);
  else
    $\mathcal{T}$-Solver_IncompleteCheck(M);
    S = ReactTo$\mathcal{T}$-Solver(S);
    if ($\mathcal{T}$*-Solver failed* **or** *found model for* M) **then**
      if ($M \models N$ **or** *complete check heuristic*) **then**
        $\mathcal{T}$-Solver_CompleteCheck(M);
        S = ReactTo$\mathcal{T}$-Solver(S);
        if ($\mathcal{T}$*-Solver found model for* M) **then**
          return(S);
      else
        Decide(S);
        $\mathcal{T}$-Solver_AddDecision(L);
        $\mathcal{T}$-Solver_Assert(L);
return(S);

Algorithm 2: ReactTo$\mathcal{T}$-Solver

Input : A state $(M; N; U; k; \top)$.
Output: A state $S = (M'; N; U'; k'; D)$, $D \in \{\top, \bot\}$

if (*detected* $\mathcal{T}$*-Conflict*) **then**
  $C = \mathcal{T}$-Solver_GetConflict(S);
  $\mathcal{T}$**-Conflict'**(S, C);
  S = Analyze(S);
else if (*detected* $\mathcal{T}$*-Propagations*) **then**
  $L_1, \ldots, L_n = \mathcal{T}$-Solver_GetPropagations($S$);
  $\mathcal{T}$**-Propagate**(S, L);
  $\mathcal{T}$-Solver_Assert(L);
else if (*decided to learn clauses in* T') **then**
  $T' = \mathcal{T}$-Solver_GetLearnedClauses(S);
  $\mathcal{T}$**-Learn**(S, T');
return(S);

Algorithm 3: Analyze(S)

Input : A state $(M; N; U; k; D)$ with $D \notin \{\top, \bot\}$.
Output: A state $S = (M'; N; U'; k'; D)$, $D \in \{\top, \bot\}$

whilerule (**Skip**(S) **or Resolve**(S)) **do**
  ;
if ($\mathcal{T}$*-forget heuristic*) **then**
  $\mathcal{T}$**-Forget**(S, N');
ifrule (**Backtrack**(S)) **then**
  $\mathcal{T}$-Solver.backtrack(k);
return(S);

Algorithm 1: CDCL($\mathcal{T}$)(S)

Input : An initial state $(\epsilon; N; \emptyset; 0; \top)$.
Output: A final state $S = (M; N; U; k; D)$, $D \in \{\top, \bot\}$

```
for (L ∈ atoms(N)) do
    T-Solver_Inform(L);
while (any rule applicable) do
    ifrule (Conflict(S)) then
        S = Analyze(S);
    else ifrule (Propagate(S)) then
        T-Solver_Assert(L);
    else
        T-Solver_IncompleteCheck(M);
        S = ReactToT-Solver(S);
        if (T-Solver failed or found model for M) then
            if (M ⊨ N or complete check heuristic) then
                T-Solver_CompleteCheck(M);
                S = ReactToT-Solver(S);
                if (T-Solver found model for M) then
                    return(S);
            else
                Decide(S);
                T-Solver_AddDecision(L);
                T-Solver_Assert(L);
return(S);
```

Algorithm 2: ReactTo$\mathcal{T}$-Solver

Input : A state $(M; N; U; k; \top)$.
Output: A state $S = (M'; N; U'; k'; D)$, $D \in \{\top, \bot\}$

```
if (detected T-Conflict) then
    C = T-Solver_GetConflict(S);
    T-Conflict'(S, C);
    S = Analyze(S);
else if (detected T-Propagations) then
    L_1, ..., L_n = T-Solver_GetPropagations(S);
    T-Propagate(S, L);
    T-Solver_Assert(L);
else if (decided to learn clauses in T') then
    T' = T-Solver_GetLearnedClauses(S);
    T-Learn(S, T');
return(S);
```

Algorithm 3: Analyze(S)

Input : A state $(M; N; U; k; D)$ with $D \notin \{\top, \bot\}$.
Output: A state $S = (M'; N; U'; k'; D)$, $D \in \{\top, \bot\}$

```
whilerule (Skip(S) or Resolve(S)) do
    ;
if (T-forget heuristic) then
    T-Forget(S, N');
ifrule (Backtrack(S)) then
    T-Solver.backtrack(k);
return(S);
```

Interface

- Incremental, backtrackable
  - Inform
  - AddDecision
  - Assert
  - Backtrack
  - CompleteCheck
  - IncompleteCheck
  - GetPropagations
  - GetReason
  - GetConflict
  - GetLearnedClauses

Interface

- Incremental, backtrackable
  - Inform
  - AddDecision
  - Assert
  - Backtrack
  - CompleteCheck
  - IncompleteCheck
  - GetPropagations
  - GetReason
  - GetConflict
  - GetLearnedClauses

Implementation – Architecture