# Decision Procedures

Jochen Hoenicke

Software Engineering
Albert-Ludwigs-University Freiburg

Winter Term 2019/2020

# DPLL(T)

# Satisfiability and Conjunctive Theories

Suppose we have a $T_{\mathbb{Q}}$-formulae that is not conjunctive:

$$(x \geq 0 \rightarrow y > z) \wedge (x + y \geq z \rightarrow y \leq z) \wedge (y \geq 0 \rightarrow x \geq 0) \wedge x + y \geq z$$

Our approach so far: Converting to DNF.
Yields in 8 conjuncts that have to be checked separately.

Is there a more efficient way to prove unsatisfiability?

# CNF and Propositional Core

Suppose we have the following $T_{\mathbb{Q}}$-formulae:

$$(x \geq 0 \rightarrow y > z) \land (x + y \geq z \rightarrow y \leq z) \land (y \geq 0 \rightarrow x \geq 0) \land x + y \geq z$$

Converting to CNF and restricting to $\leq$:

$$\begin{aligned}(\neg(0 \leq x) \lor \neg(y \leq z)) \land (\neg(z \leq x + y) \lor (y \leq z))\\ \land(\neg(0 \leq y) \lor (0 \leq x)) \land (z \leq x + y)\end{aligned}$$

Now, introduce boolean variables for each atom:

$$\begin{array}{llll} P_1 : 0 \leq x & \qquad & P_2 : y \leq z \\ P_3 : z \leq x + y & & P_4 : 0 \leq y \end{array}$$

Gives a propositional formula:

$$(\neg P_1 \lor \neg P_2) \land (\neg P_3 \lor P_2) \land (\neg P_4 \lor P_1) \land P_3$$

# DPLL-Algorithm

The core feature of the DPLL-algorithm is Unit Propagation.

$$(\neg P_1 \lor \neg P_2) \land (\neg P_3 \lor P_2) \land (\neg P_4 \lor P_1) \land P_3$$

The clause $P_3$ is a unit clause; set $P_3$ to $\top$.
Then $\neg P_3 \lor P_2$ is a unit clause; set $P_2$ to $\top$.
Then $\neg P_1 \lor \neg P_2$ is a unit clause; set $P_1$ to $\bot$.
Then $\neg P_4 \lor P_1$ is a unit clause; set $P_4$ to $\bot$.

Only solution is $P_3 \land P_2 \land \neg P_1 \land \neg P_4$.

# DPLL-Algorithm

Only solution is $P_3 \wedge P_2 \wedge \neg P_1 \wedge \neg P_4$.

$$P_1 : 0 \leq x \qquad\qquad P_2 : y \leq z$$
$$P_3 : z \leq x + y \qquad\qquad P_4 : 0 \leq y$$

This gives the conjunctive $T_{\mathbb{Q}}$-formula

$$z \leq x + y \wedge y \leq z \wedge x < 0 \wedge y < 0.$$

# DPLL(T) with Learning (CDCL)

We will extend the DPLL algorithm to support theory reasoning.
Reminder DPLL was described by a set of rules modifying a configuration.
A configuration is a triple

$$\langle M, F, C \rangle ,$$

where

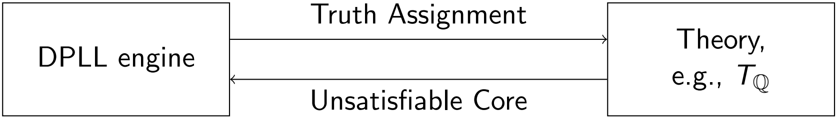
- $M$ (model) is a sequence of literals (that are currently set to true) interspersed with backtracking points denoted by $\square$.
- $F$ (formula) is a formula in CNF, i. e., a set of clauses where each clause is a set of literals.
- $C$ (conflict) is either $\top$ or a conflict clause (a set of literals). A conflict clause $C$ is a clause with $F \Rightarrow C$ and $M \not\models C$. Thus, a conflict clause shows $M \not\models F$.

# Rules for CDCL (Conflict Driven Clause Learning)

| | | |
|---|---|---|
| Decide | $\dfrac{\langle M, F, \top\rangle}{\langle M \cdot \ell^{\Box}, F, \top\rangle}$ | where $\ell \in \mathit{lit}(F)$, $\ell, \overline{\ell}$ ~~in~~ $M$ |
| Propagate | $\dfrac{\langle M, F, \top\rangle}{\langle M \cdot \ell^{C_\ell}, F, \top\rangle}$ | where $C_\ell = \{\ell_1, \dots, \ell_k, \ell\} \in F$ with $\overline{\ell_1}, \dots, \overline{\ell_k}$ in $M$, $\ell, \overline{\ell}$ ~~in~~ $M$. |
| Conflict | $\dfrac{\langle M, F, \top\rangle}{\langle M, F, \{\ell_1, \dots, \ell_k\}\rangle}$ | where $\{\ell_1, \dots, \ell_k\} \in F$ and $\overline{\ell_1}, \dots, \overline{\ell_k}$ in $M$. |
| Explain | $\dfrac{\langle M, F, C \cup \{\overline{\ell}\}\rangle}{\langle M, F, C \cup \{\ell_1, \dots, \ell_k\}\rangle}$ | where $\overline{\ell} \notin C$, $\ell^{C_\ell}$ in $M$, and $C_\ell = \{\ell_1, \dots, \ell_k, \ell\}$. |
| Learn | $\dfrac{\langle M, F, C\rangle}{\langle M, F \cup \{C\}, C\rangle}$ | where $C \neq \top$, $C \notin F$. |
| Back | $\dfrac{\langle M, F, C_\ell\rangle}{\langle M' \cdot \ell^{C_\ell}, F, \top\rangle}$ | where $C_\ell = \{\ell_1, \dots, \ell_k, \ell\} \in F$, $M = M' \cdot \ell'^{\Box} \cdots$, and $\overline{\ell_1}, \dots, \overline{\ell_k}$ in $M'$, $\overline{\ell}$ ~~in~~ $M'$. |

# DPLL(T): DPLL Modulo Theory

The DPLL/CDCL algorithm is combined with a Decision Procedures for a Theory




DPLL takes the propositional core of a formula,
assigns truth-values to atoms.

Theory takes a conjunctive formula (conjunction of literals),
returns a minimal unsatisfiable core.

# Minimal Unsatisfiable Core

Suppose we have a decision procedure for a conjunctive theory, e.g., Simplex Algorithm for $T_{\mathbb{Q}}$.

Given an unsatisfiable conjunction of literals $\ell_1 \wedge \cdots \wedge \ell_n$.
Find a subset $\text{UnsatCore} = \{\ell_{i_1}, \ldots, \ell_{i_m}\}$, such that

- $\ell_{i_1} \wedge \ldots \wedge \ell_{i_m}$ is unsatisfiable.
- For each subset of UnsatCore the conjunction is satisfiable.

Possible approach: check for each literal whether it can be omitted.
$\longrightarrow$ $n$ calls to decision procedure.

Most decision procedures can give small unsatisfiable cores for free.

# Unsatisfiable Core and Conflict Clause

Theory returns an unsatisfiable core:

- a conjunction of literals from current truth assignment
- that is unsatisfiable.

DPLL learns conflict clauses, a disjunction of literals

- that are implied by the formula
- and in conflict to current truth assignment.

Thus the negation of an unsatisfiable core is a conflict clause.

The DPLL part only needs one new rule:

TConflict $\dfrac{\langle M, F, \top\rangle}{\langle M, F, C\rangle}$ where $M$ is unsatisfiable in the theory and $\neg C$ an unsatisfiable core of $M$.

# Example: DPLL(T)

$$F : \; y \geq 1 \land (x \geq 0 \rightarrow y \leq 0) \land (x \leq 1 \rightarrow y \leq 0)$$

Atomic propositions:

$$P_1 : y \geq 1 \qquad\qquad P_2 : x \geq 0$$
$$P_3 : y \leq 0 \qquad\qquad P_4 : x \leq 1$$

Propositional core of $F$ in CNF:

$$F_0 : (P_1) \land (\neg P_2 \lor P_3) \land (\neg P_4 \lor P_3)$$

# Running DPLL(T)

$$F_0 : \{\{P_1\}, \{\overline{P_2}, P_3\}, \{\overline{P_4}, P_3\}\}$$
$$P_1 : y \geq 1 \quad P_2 : x \geq 0 \quad P_3 : y \leq 0 \quad P_4 : x \leq 1$$

$$\langle \epsilon, F_0, \top \rangle \xrightarrow{\text{Propagate}} \langle P_1, F_0, \top \rangle \xrightarrow{\text{Decide}} \langle P_1 \Box P_3, F_0, \top \rangle \xrightarrow{\text{TConflict}}$$
$$\langle P_1 \Box P_3, F_0, \{\overline{P_1}, \overline{P_3}\} \rangle \xrightarrow{\text{Learn}} \langle P_1 \Box P_3, F_1, \{\overline{P_1}, \overline{P_3}\} \rangle \xrightarrow{\text{Back}}$$
$$\langle P_1 \overline{P_3}, F_1, \top \rangle \xrightarrow{\text{Propagate}} \langle P_1 \overline{P_3 P_2}, F_1, \top \rangle \xrightarrow{\text{Propagate}}$$
$$\langle P_1 \overline{P_3 P_2 P_4}, F_1, \top \rangle \xrightarrow{\text{TConflict}} \langle P_1 \overline{P_3 P_2 P_4}, F_1, \{P_2, P_4\} \rangle \xrightarrow{\text{Explain}}$$
$$\langle P_1 \overline{P_3 P_2 P_4}, F_1, \{P_2, P_3\} \rangle \xrightarrow{\text{Explain}} \langle P_1 \overline{P_3 P_2 P_4}, F_1, \{P_3\} \rangle \xrightarrow{\text{Explain}}$$
$$\langle P_1 \overline{P_3 P_2 P_4}, F_1, \{\overline{P_1}\} \rangle \xrightarrow{\text{Explain}} \langle P_1 \overline{P_3 P_2 P_4}, F_1, \emptyset \rangle \xrightarrow{\text{Learn}}$$
$$\langle P_1 \overline{P_3 P_2 P_4}, F_1 \cup \{\emptyset\}, \emptyset \rangle$$

where $F_1 := F_0 \cup \{\{\overline{P_1}, \overline{P_3}\}\}$

No further step is possible; the formula $F$ is unsatisfiable.

# Correctness of DPLL(T)

UNI FREIBURG

## Theorem (Correctness of DPLL(T))

*Let $F$ be a $\Sigma$-formula and $F'$ its propositional core. Let*

$$\langle \epsilon, F', \top \rangle = \langle M_0, F_0, C_0 \rangle \longrightarrow \ldots \longrightarrow \langle M_n, F_n, C_n \rangle$$

*be a maximal sequence of rule application of DPLL(T).*

*Then $F$ is $T$-satisfiable iff $C_n$ is $\top$.*

## Theorem (Termination of DPLL)

*Let $F$ be a propositional formula. Then every sequence*

$$\langle \epsilon, F, \top \rangle = \langle M_0, F_0, C_0 \rangle \longrightarrow \langle M_1, F_1, C_1 \rangle \longrightarrow \ldots$$

*terminates.*