CDCL SAT Solvers & SAT-Based Problem Solving

Joao Marques-Silva\textsuperscript{1,2} & Mikolas Janota\textsuperscript{2}

\textsuperscript{1}University College Dublin, Ireland
\textsuperscript{2}IST/INESC-ID, Lisbon, Portugal

SAT/SMT Summer School 2013
Aalto University, Espoo, Finland
The Success of SAT

• Well-known NP-complete decision problem
• In practice, SAT is a success story of Computer Science
  – Hundreds (even more?) of practical applications
Part I

CDCL SAT Solvers
Outline

Basic Definitions

DPLL Solvers

CDCL Solvers

What Next in CDCL Solvers?
Outline

Basic Definitions

DPLL Solvers

CDCL Solvers

What Next in CDCL Solvers?
Preliminaries

- **Variables**: $w, x, y, z, a, b, c, \ldots$
- **Literals**: $w, \bar{x}, \bar{y}, a, \ldots$, but also $\neg w, \neg y, \ldots$
- **Clauses**: disjunction of literals or set of literals
- **Formula**: conjunction of clauses or set of clauses
- **Model** (satisfying assignment): partial/total mapping from variables to $\{0, 1\}$
- Formula can be **SAT/UNSAT**
Preliminaries

- **Variables**: \( w, x, y, z, a, b, c, \ldots \)
- **Literals**: \( w, \bar{x}, \bar{y}, a, \ldots \), but also \( \neg w, \neg y, \ldots \)
- **Clauses**: disjunction of literals or set of literals
- **Formula**: conjunction of clauses or set of clauses
- **Model** (satisfying assignment): partial/total mapping from variables to \( \{0, 1\} \)
- **Formula can be** SAT/UNSAT
- **Example**:

\[
F \triangleq (r) \land (\bar{r} \lor s) \land (\bar{w} \lor a) \land (\bar{x} \lor b) \land (\bar{y} \lor \bar{z} \lor c) \land (\bar{b} \lor \bar{c} \lor d)
\]

- Example models:
  - \( \{r, s, a, b, c, d\} \)
  - \( \{r, s, \bar{x}, y, \bar{w}, z, \bar{a}, b, c, d\} \)
Resolution

- Resolution rule:

\[
\frac{(\alpha \lor x) \quad (\beta \lor \bar{x})}{(\alpha \lor \beta)}
\]

- Complete proof system for propositional logic
Resolution

- Resolution rule:

\[
\frac{(\alpha \lor x) \quad (\beta \lor \bar{x})}{\alpha \lor \beta}
\]

- Complete proof system for propositional logic

- Extensively used with (CDCL) SAT solvers
Resolution

- Resolution rule: \[
\frac{(\alpha \lor x) \quad (\beta \lor \bar{x})}{(\alpha \lor \beta)}
\]

- Complete proof system for propositional logic

- Extensively used with (CDCL) SAT solvers

- Self-subsuming resolution (with \(\alpha' \subseteq \alpha\)):

\[
\frac{(\alpha \lor x) \quad (\alpha' \lor \bar{x})}{(\alpha)}
\]

- \((\alpha)\) subsumes \((\alpha \lor x)\)
Unit Propagation

\[ \mathcal{F} = (r) \land (\bar{r} \lor s) \land \\
(\bar{w} \lor a) \land (\bar{x} \lor \bar{a} \lor b) \\
(\bar{y} \lor \bar{z} \lor c) \land (\bar{b} \lor \bar{c} \lor d) \]
Unit Propagation

\[ \mathcal{F} = (r) \land (\bar{r} \lor s) \land \\
(\bar{w} \lor a) \land (\bar{x} \lor \bar{a} \lor b) \\
(\bar{y} \lor \bar{z} \lor c) \land (\bar{b} \lor \bar{c} \lor d) \]

- Decisions / Variable Branchings:
  \[ w = 1, x = 1, y = 1, z = 1 \]
Unit Propagation

\[ F = (r) \land (\bar{r} \lor s) \land \\
(\bar{w} \lor a) \land (\bar{x} \lor \bar{a} \lor b) \\
(\bar{y} \lor \bar{z} \lor c) \land (\bar{b} \lor \bar{c} \lor d) \]

- Decisions / Variable Branchings:
  \[ w = 1, x = 1, y = 1, z = 1 \]

<table>
<thead>
<tr>
<th>Level</th>
<th>Dec.</th>
<th>Unit Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>\emptyset</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>( w )</td>
<td>( a )</td>
</tr>
<tr>
<td>2</td>
<td>( x )</td>
<td>( b )</td>
</tr>
<tr>
<td>3</td>
<td>( y )</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>( z )</td>
<td>( c ) ( d )</td>
</tr>
</tbody>
</table>
Unit Propagation

\[ F = (r) \land (\overline{r} \lor s) \land \\
(\overline{\overline{w}} \lor a) \land (\overline{x} \lor \overline{\overline{a}} \lor b) \\
(\overline{y} \lor \overline{\overline{z}} \lor c) \land (\overline{b} \lor \overline{\overline{c}} \lor d) \]

- Decisions / Variable Branchings:
  \( w = 1, x = 1, y = 1, z = 1 \)

- Additional definitions:
  - Antecedent (or reason) of an implied assignment
    - \((\overline{b} \lor \overline{\overline{c}} \lor d)\) for \(d\)
  - Associate assignment with decision levels
    - \(w = 1 \oplus 1, x = 1 \oplus 2, y = 1 \oplus 3, z = 1 \oplus 4\)
    - \(r = 1 \oplus 0, d = 1 \oplus 4, \ldots\)
Outline

Basic Definitions

DPLL Solvers

CDCL Solvers

What Next in CDCL Solvers?
The DPLL Algorithm

- Optional: pure literal rule
The DPLL Algorithm

\[ \mathcal{F} = (x \lor y) \land (a \lor b) \land (\bar{a} \lor b) \land (a \lor \bar{b}) \land (\bar{a} \lor \bar{b}) \]

- Optional: pure literal rule
The DPLL Algorithm

- **Optional:** pure literal rule

\[ \mathcal{F} = (x \lor y) \land (a \lor b) \land (\overline{a} \lor b) \land (a \lor \overline{b}) \land (\overline{a} \lor \overline{b}) \]

<table>
<thead>
<tr>
<th>Level</th>
<th>Dec.</th>
<th>Unit Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>\emptyset</td>
<td>\null</td>
</tr>
<tr>
<td>1</td>
<td>(x)</td>
<td>\null</td>
</tr>
<tr>
<td>2</td>
<td>(y)</td>
<td>\null</td>
</tr>
<tr>
<td>3</td>
<td>(a) \rightarrow (b) \rightarrow \bot</td>
<td>\null</td>
</tr>
</tbody>
</table>
The DPLL Algorithm

Unassigned variables?

- Yes: Satisfiable
- No: Assign value to variable

Assign value to variable

- Yes: Unit propagation
- No: Conflict?

Conflict?

- Yes: Can undo decision?
  - Yes: Backtrack & flip variable
  - No: Unassigned variables
- No: Conflict?

Optional: pure literal rule

\[ F = (x \lor y) \land (a \lor b) \land (\bar{a} \lor b) \land (a \lor \bar{b}) \land (\bar{a} \lor \bar{b}) \]

Level Dec. Unit Prop.
0 \( \emptyset \)
1 \( x \)
2 \( y \)
3 \( \bar{a} \rightarrow \bar{b} \rightarrow \bot \)
The DPLL Algorithm

- Optional: pure literal rule

\[ F = (x \lor y) \land (a \lor b) \land (\bar{a} \lor b) \land (a \lor \bar{b}) \land (\bar{a} \lor \bar{b}) \]

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
<th>Unit Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(\emptyset)</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>(x)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(\bar{y})</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>(a) \rightarrow (b) \rightarrow \bot</td>
<td></td>
</tr>
</tbody>
</table>

Diagram showing the decision tree with assignments and unit propagation.
The DPLL Algorithm

\[ \mathcal{F} = (x \lor y) \land (a \lor b) \land (\bar{a} \lor b) \land (a \lor \bar{b}) \land (\bar{a} \lor \bar{b}) \]

- Level 0: \( \emptyset \)
- Level 1: \( x \)
- Level 2: \( \bar{y} \)
- Level 3: \( \bar{a} \rightarrow \bar{b} \rightarrow \bot \)

- Optional: pure literal rule

Diagram:

- Unassigned variables?
  - Yes: Satisfiable
  - No: Unit propagation
- Conflict?
  - Yes: Can undo decision?
  - No: Unsatisfiable
  - Yes: Backtrack & flip variable

Optional: pure literal rule
The DPLL Algorithm

- **Optional**: pure literal rule

\[ F = (x \lor y) \land (a \lor b) \land (\overline{a} \lor b) \land (a \lor \overline{b}) \land (\overline{a} \lor \overline{b}) \]

- **Level**
  - 0 : \( \emptyset \)
  - 1 : \( \overline{x} \) \( \rightarrow \) \( y \)
  - 2 : \( a \) \( \rightarrow \) \( b \) \( \rightarrow \) \( \bot \)

Diagram:

- Decision tree for \( F \)
The DPLL Algorithm

\[
\mathcal{F} = (x \lor y) \land (a \lor b) \land (\bar{a} \lor b) \land (a \lor \bar{b}) \land (\bar{a} \lor \bar{b})
\]

- Level 0: \(\emptyset\)
- Level 1: \(\bar{x} \rightarrow y\)
- Level 2: \(\bar{a} \rightarrow \bar{b} \rightarrow \bot\)

- Optional: pure literal rule
Outline

Basic Definitions

DPLL Solvers

CDCL Solvers

What Next in CDCL Solvers?
What is a CDCL SAT Solver?

- Extend **DPLL SAT** solver with:
  - **Clause learning & non-chronological backtracking**
    - Exploit UIPs
    - Minimize learned clauses
    - Opportunistically delete clauses
  - **Search restarts**
  - **Lazy data structures**
    - Watched literals
  - **Conflict-guided branching**
    - Lightweight branching heuristics
    - Phase saving
  - ...
How Significant are CDCL SAT Solvers?

Results of the SAT competition/race winners on the SAT 2009 application benchmarks, 20mn timeout

- Limmat (2002)
- Zchaff (2002)
- Berkmin (2002)
- Forklift (2003)
- Siege (2003)
- SatELite (2005)
- Minisat 2 (2006)
- Picosat (2007)
- Rsat (2007)
- Minisat 2.1 (2008)
- Precosat (2009)
- Glucose (2009)
- Clasp (2009)
- Cryptominisat (2010)
- Lingeling (2010)
- Minisat 2.2 (2010)
- Glucose 2 (2011)
- Glueminisat (2011)
- Contrasat (2011)
- Lingeling 587f (2011)

GRASP
DPLL
Outline

Basic Definitions

DPLL Solvers

CDCL Solvers

Clause Learning, UIPs & Minimization

Search Restarts & Lazy Data Structures

What Next in CDCL Solvers?
## Clause Learning

<table>
<thead>
<tr>
<th>Level</th>
<th>Dec.</th>
<th>Unit Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\emptyset$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$x$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$y$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$z$</td>
<td>$a \rightarrow \bot$</td>
</tr>
</tbody>
</table>

- **Analyze conflict**
  - **Reasons:**
    - Decision variable & literals assigned at lower decision levels
  - Create new clause: $(\overline{x}, \overline{z})$

- Can relate clause learning with resolution
  - Learned clauses result from (selected) resolution operations
Clause Learning

<table>
<thead>
<tr>
<th>Level</th>
<th>Dec.</th>
<th>Unit Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>∅</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>z</td>
<td>a</td>
</tr>
</tbody>
</table>

- Analyze conflict
  - Reasons: x and z
    - Decision variable & literals assigned at lower decision levels
  - Create new clause: \((\bar{x} \lor \bar{z})\)
- Can relate clause learning with resolution
Clause Learning – After Bracktracking

Level | Dec. | Unit Prop.
-----|------|--------
0    | ∅    |        
1    | x    |        
2    | y    |        
3    | z    | a      | ⊥

 clause \( \overline{x}_\overline{z} \) is asserting at decision level 1

Learned clauses are always asserting \([MSS96,MSS99]\)

Backtracking differs from plain DPLL:

– Always backtrack after a conflict \([MMZZM01]\)
Clause Learning – After Bracktracking

<table>
<thead>
<tr>
<th>Level</th>
<th>Dec.</th>
<th>Unit Prop.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\emptyset$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$x$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>$y$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>$z$</td>
<td></td>
</tr>
</tbody>
</table>

- Clause $(\bar{x} \lor \bar{z})$ is asserting at decision level 1
Clause Learning – After Backtracking

- Learned clauses are always asserting ([MSS96, MSS99])
- Backtracking differs from plain DPLL:
  - Always backtrack after a conflict ([MMZZM01])

- Clause \((\bar{x} \lor \bar{z})\) is asserting at decision level 1
Clause Learning – After Backtracking

- Clause \((\overline{x} \lor \overline{z})\) is asserting at decision level 1
- Learned clauses are always asserting
- Backtracking differs from plain DPLL:
  - Always backtrack after a conflict

[MSS96,MSS99]

[MMZZM01]