SAT solver essentials, SAT modeling
Incremental SAT

Gilles Audemard

VTSA School - Liege - 2021

Thanks to N. Szczepanski and L. Simon
A surprising effect of solvers’ efficiency: used as NP-Complete oracles

- IC3: thousands of calls on simple formulas [Bradley 2012]
- MUS extraction [Belov et al. 2012]
- MaxSAT

Many calls on similar instances

CDCL solvers learn form the PAST!!

Keep the solver alive
The formula is inconsistent: Why?

Minimal unsatisfiable subset of clauses
The formula is inconsistent: Why?

Minimal unsatisfiable subset of clauses
The formula is inconsistent: Why?

Minimal unsatisfiable subset of clauses

Different approaches

- Local search [Piette et al, ECAI 2006]
- Resolution based [Nadel, FMCAD 2010]
- Constructive or destructive [Belov et al, AI Com 2012]. The tool MUSER
The formula is inconsistent: Why?

Minimal unsatisfiable subset of clauses

Different approaches

- Local search [Piette et al, ECAI 2006]
- Resolution based [Nadel, FMCAD 2010]
- Constructive or destructive [Belov et al, AI Com 2012]. The tool MUSER
The formula is inconsistent: Why?

Minimal unsatisfiable subset of clauses

Different approaches

- Local search [Piette et al, ECAI 2006]
- Resolution based [Nadel, FMCAD 2010]
- Constructive or destructive [Belov et al, AI Com 2012]. The tool MUSER
The formula is inconsistent: Why?

Minimal unsatisfiable subset of clauses

Different approaches

- Local search [Piette et al, ECAI 2006]
- Resolution based [Nadel, FMCAD 2010]
- Constructive or destructive [Belov et al, AI Com 2012]. The tool MUSER
Minimum Unsatisfiable Subformula

The formula is inconsistent: Why?

Minimal unsatisfiable subset of clauses

Different approaches

- Local search [Piette et al, ECAI 2006]
- Resolution based [Nadel, FMCAD 2010]
- Constructive or destructive [Belov et al, AI Com 2012]. The tool MUSER
The formula is inconsistent: Why?

- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov et al, AI Com 2012]. The tool MUSER
The formula is inconsistent: Why?

- Minimal unsatisfiable subset of clauses
- Different approaches
  - Local search [Piette et al, ECAI 2006]
  - Resolution based [Nadel, FMCAD 2010]
  - Constructive or destructive [Belov et al, AI Com 2012]. The tool MUSER
Working with Assumptions

- A formula $F$
- A set of assumptions, $\ell_1, \ell_2, \ldots, \ell_n$ with $\ell_i$ are (fresh) literals
- Solve $F \land \ell_1 \land \ell_2 \ldots \land \ell_n$
- Incremental SAT solving: the process can be repeated with new assumptions
Working with Assumptions

- A formula $F$
- A set of assumptions, $\ell_1, \ell_2, \ldots, \ell_n$ with $\ell_i$ are (fresh) literals
- Solve $F \land \ell_1 \land \ell_2 \ldots \land \ell_n$
- Incremental SAT solving: the process can be repeated with new assumptions

First solution

- Simplify: $F' = F \land \ell_1 \land \ell_2 \ldots \land \ell_n$
- Solve $F'$
- Learnt clauses can not be kept
Working with Assumptions

- A formula $F$
- A set of assumptions, $\ell_1, \ell_2, \ldots, \ell_n$ with $\ell_i$ are (fresh) literals
- Solve $F \land \ell_1 \land \ell_2 \ldots \land \ell_n$
- Incremental SAT solving: the process can be repeated with new assumptions
Working with Assumptions

- A formula $F$
- A set of assumptions, $\ell_1, \ell_2, \ldots, \ell_n$ with $\ell_i$ are (fresh) literals
- Solve $F \land \ell_1 \land \ell_2 \ldots \land \ell_n$
- Incremental SAT solving: the process can be repeated with new assumptions

Second Solution

- First, selects all assumptions as decision variables:
  one level $\Rightarrow$ one assumption

- Second, Run the SAT solver as usual

- All learnt clauses can be kept

- One can explain unsatisfiability wrt set of assumptions!
Forget some clauses

- Add one selector (fresh variable) $\ell_i$ per clause

\[
\begin{align*}
\ell_1 & \lor x \lor y \lor z \\
\ell_4 & \lor \neg x \lor y \lor z \\
\ell_7 & \lor \neg x \lor \neg y \\
\ell_2 & \lor x \lor \neg y \\
\ell_5 & \lor x \lor w \\
\ell_8 & \lor \neg x \lor \neg z \\
\ell_3 & \lor x \lor \neg z \\
\ell_6 & \lor w \lor z \lor \neg y \\
\ell_9 & \lor w \lor \neg x \lor \neg z
\end{align*}
\]

- Learnt clause contains selectors of all original clauses used to generate it
Muser Architecture

Incremental SAT

One of the best MUS extractor
- Successive calls to a SAT oracle
- Non independent calls
- Informations between two calls are preserved
  - Heuristics: VSIDS, phase saving, restarts...
  - Learnt clauses
Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to **activate** the clause $i$
- Assign $\ell_j$ (as an assumption) to true to **disable** the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too!

\[\ell_1 \lor x \lor y \lor z\]
\[\ell_2 \lor x \lor \neg y\]
\[\ell_3 \lor x \lor \neg z\]
\[\ell_4 \lor \neg x \lor y \lor z\]
\[\ell_5 \lor x \lor w\]
\[\ell_6 \lor w \lor z \lor \neg y\]
\[\ell_7 \lor \neg x \lor \neg y\]
\[\ell_8 \lor \neg x \lor \neg z\]
\[\ell_9 \lor w \lor \neg x \lor \neg z\]
\[\ell_1 \lor \ell_2 \lor x \lor z\]
Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to activate the clause $i$
- Assign $\ell_j$ (as an assumption) to true to disable the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too!

\[
\begin{align*}
\ell_1 & \lor x \lor y \lor z \\
\ell_2 & \lor x \lor \neg y \\
\ell_3 & \lor x \lor \neg z \\
\ell_4 & \lor \neg x \lor y \lor z \\
\ell_5 & \lor x \lor w \\
\ell_6 & \lor w \lor z \lor \neg y \\
\ell_7 & \lor \neg x \lor \neg y \\
\ell_8 & \lor \neg x \lor \neg z \\
\ell_9 & \lor w \lor \neg x \lor \neg z \\
\ell_1 \lor \ell_2 & \lor x \lor z
\end{align*}
\]
Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to **activate** the clause $i$
- Assign $\ell_j$ (as an assumption) to true to **disable** the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too!

$$x \lor y \lor z$$

$$\ell_2 \lor x \lor \neg y$$

$$\ell_3 \lor x \lor \neg z$$

$$\ell_4 \lor \neg x \lor y \lor z$$

$$\ell_5 \lor x \lor w$$

$$\ell_6 \lor w \lor z \lor \neg y$$

$$\ell_7 \lor \neg x \lor \neg y$$

$$\ell_8 \lor \neg x \lor \neg z$$

$$\ell_9 \lor w \lor \neg x \lor \neg z$$

$$\ell_1 \lor \ell_2 \lor x \lor z$$
Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to **activate** the clause $i$
- Assign $\ell_j$ (as an assumption) to true to **disable** the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too!

\[
x \vee y \vee z
\]

\[
\ell_1 \vee \ell_2 \vee x \vee z
\]
Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to **activate** the clause $i$
- Assign $\ell_j$ (as an assumption) to true to **disable** the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too!

\[
x \lor y \lor z
\]
\[
\ell_2 \lor x \lor \neg y
\]
\[
\ell_3 \lor x \lor \neg z
\]
\[
\ell_4 \lor \neg x \lor y \lor z
\]
\[
\ell_5 \lor x \lor w
\]
\[
\ell_6 \lor w \lor z \lor \neg y
\]
\[
\ell_7 \lor \neg x \lor \neg y
\]
\[
\ell_8 \lor \neg x \lor \neg z
\]
\[
\ell_9 \lor w \lor \neg x \lor \neg z
\]
\[
\ell_2 \lor x \lor z
\]
Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to **activate** the clause $i$
- Assign $\ell_j$ (as an assumption) to true to **disable** the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too!

\[
x \lor y \lor z
\]

\[
\begin{align*}
\ell_3 & \lor x \lor \neg z \\
\ell_4 & \lor \neg x \lor y \lor z \\
\ell_5 & \lor x \lor w \\
\ell_6 & \lor w \lor z \lor \neg y \\
\ell_7 & \lor \neg x \lor \neg y \\
\ell_8 & \lor \neg x \lor \neg z \\
\ell_9 & \lor w \lor \neg x \lor \neg z \\
\ell_2 & \lor x \lor z
\end{align*}
\]

DL 1

\begin{tikzpicture}[shorten >=1pt,node distance=2cm,on grid,auto]
    
    				node[fill=yellow!50] (1) at (0,0) {$\ell_1$};

    
    				node[fill=yellow!50] (2) at (1,0) {$\ell_2$};

    
    				draw (1) -- (2);

\end{tikzpicture}

DL 2
Forget Some Clauses

- Assign $\ell_i$ (as an assumption) to false to activate the clause $i$
- Assign $\ell_j$ (as an assumption) to true to disable the clause $j$
- All learnt clauses related to the clause $j$ a disable clause are disabled too!

$x \lor y \lor z$

$\ell_3 \lor x \lor \neg z$
$\ell_4 \lor \neg x \lor y \lor z$
$\ell_5 \lor x \lor w$
$\ell_6 \lor w \lor z \lor \neg y$
$\ell_7 \lor \neg x \lor \neg y$
$\ell_8 \lor \neg x \lor \neg z$
$\ell_9 \lor w \lor \neg x \lor \neg z$

DL 1

$\ell_1$

DL 2

$\ell_2$
Glucose inside Muser

MUSER ($\Sigma$) \rightarrow MUS \rightarrow SAT/UNSAT

$\Sigma' \subseteq \Sigma$ 

MINISAT ($\Sigma'$)
Glucose inside Muser

- Plug GLUCOSE in MUSER
- Adapt and modify GLUCOSE to improve MUSER performances

Improve SAT oracle in order to improve the MUSER tool
Test Set

- 300 instances from the SAT competition 2011, MUS category
- timeout set to 2400 seconds
- MUSER is used with default options (destructive approach, model rotation)
A First Attempt

Resolution time

Glucose 2.1
(261 solved)

(259 points)

Minisat
(273 solved)
Disappointing Results

Trying to explain these bad results
Disappointing Results

![Graph showing Nb SAT calls vs. points for Minisat and Glucose 2.1](image)

- **Minisat**: (273 solved)
- **Glucose 2.1**: (261 solved) (259 points)

SAT - Encodings

VTSA School - Liege - 2021
Disappointing Results

Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls
- Difficult UNSAT ones
- **GLUCOSE is supposed to be good on UNSAT formulas**
Disappointing Results

Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

- GLUCOSE uses LBD for cleaning, restarts...
- Each assumption uses its own decision level
Disappointing Results

- Each point represents an instance
- x-axis is the average number of initial variables in learnt clauses
- y-axis is the average number of selector variables in learnt clauses
Disappointing Results

<table>
<thead>
<tr>
<th>Instance</th>
<th>#C</th>
<th>time</th>
<th>avg size</th>
<th>max size</th>
<th>LBD avg</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>fdmus_b21_96</td>
<td>8541</td>
<td>29</td>
<td>1145</td>
<td>5980</td>
<td>1095</td>
<td>5945</td>
</tr>
<tr>
<td>longmult6</td>
<td>8853</td>
<td>46</td>
<td>694</td>
<td>3104</td>
<td>672</td>
<td>3013</td>
</tr>
<tr>
<td>dump_vc950</td>
<td>360419</td>
<td>110</td>
<td>522</td>
<td>36309</td>
<td>498</td>
<td>35873</td>
</tr>
<tr>
<td>g7n</td>
<td>70492</td>
<td>190</td>
<td>1098</td>
<td>16338</td>
<td>1049</td>
<td>16268</td>
</tr>
</tbody>
</table>

- LBD looks like size
- Clauses are very long
Disappointing Results

Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

- GLUCOSE uses LBD for cleaning, restarts...
- Each assumption uses its own decision level
- The LBD of a clause looks like its size!
Disappointing Results

Trying to explain these bad results

- Comparable number of oracle calls
- Easy SAT calls
- Difficult UNSAT ones
- GLUCOSE is supposed to be good on UNSAT formulas

- GLUCOSE uses LBD for cleaning, restarts...
- Each assumption uses its own decision level
- The LBD of a clause looks like its size!

Refine LBD: Do not take into account selectors
Many algorithms need to traverse clauses

- Dynamic computing of LBD (useful but costly)
  - Store the number of selectors in the clause
  - Stop when all initial literals have been tested

- Conflict analysis
  - Force initial literals to be placed at the beginning

- Unit propagation
  - Look for a non selector literal or a satisfied one
  - Push selectors at the end of the clause

- Deleting satisfiable clauses
  - Take only watched literals into account
Comparison

Focus on the SAT oracle
Study special cases
Try to improve it in consequence
Comparison

Focus on the SAT oracle
Study special cases
Try to improve it in consequence
Take a look at a CDCL solver (the essentials of Minisat in fact) : assumptions branch
Exercise
A simple MUS extractor