A Multi Estimations Approach for Computing Backbones of Hard and Dense Propositional Formulae

Yueling Zhang\textsuperscript{1} Min Zhang\textsuperscript{1} Geguang Pu\textsuperscript{1} Fu Song\textsuperscript{2} Jianwen Li\textsuperscript{1,3}

\textsuperscript{1}Shanghai Key Laboratory of Trustworthy Computing / East China Normal University
\textsuperscript{2} Shanghai Tech University
\textsuperscript{3} Rice University

ylzhang@sei.ecnu.edu.cn

August 22, 2016
Overview

1. Backbones

2. Motivations

3. Multi Estimations Based Backbones Extracting
   - Non-Backbone Estimation
   - Backbone Estimation \textit{HDBS}

4. Results
   - SAT Comp. Benchmark
   - MSS Formulae

5. Conclusion
Definition (Model)
Models of a propositional formula $\Phi$ are truth assignments that make the formula to TRUE.

Definition (Backbones)
Backbones of a propositional formula $\Phi$ are literals that are true in every model.

Definition (Satisfied Literal)
Given a model $\lambda$, a clause $\phi \in \Phi$. A literal $l \in \phi$ is called a satisfied literal iff $l \in \lambda$. 
Notations

- Conjunction Normal Form
- Clause
- Literal

Example

\[(a \lor \neg b \lor c) \land (a \lor b \lor \neg c) \land (a \lor b \lor c)\]
Example

CNF: \((a \lor \neg b \lor c) \land (a \lor b \lor \neg c) \land (a \lor \neg b \lor \neg c) \land (a \lor b \lor c)\)

Model Table:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Given a model \(a \land b \land c\), Literal \(a, c\) is called the satisfied literals of clause \((a \lor \neg b \lor c)\).
Motivations

- Upper bound for number of models
- Product configuration

Example

gas engine ∨ electric engine
electric engine ⟷ automatic
¬ automatic ∨ ¬ manual
electric engine

Backbones: automatic, ¬manual
Existing Approaches

- **Iterative SAT Testing**
  - $\Phi \land I$ is SAT, $I$ is not a backbone
  - $\Phi \land I$ is UNSAT, the negation of $I$ is a backbone.

- **Upper Bound Estimation**
  - $\Phi \land \phi$ is SAT, using $\phi \cup \lambda$ to remove non-backbones
  - $\Phi \land \phi$ is UNSAT, Backbones estimation terminates.

- **Core Based SAT Testing**
  - $\Phi$ is SAT under the assumption of $\phi$, $\phi \subseteq \neg \lambda$, none of the literals in $\lambda$ is backbones.
  - $\Phi$ is UNSAT under the assumption of $\phi$, if the core length is 1, the negation of core is a backbone.
Multi Estimations Based Backbones Extracting

- Non-Backbones under-approximation, $\overline{BL}_u$
- Backbones estimation, HDBS
- Iterative test literals in HDBS
Algorithm 1 Under-approximation of Non-Backbones

1: $\Psi = \overline{\text{BL}}_u(\Phi) = \emptyset$
2: $(b, \lambda) = \text{SAT}(\Phi)$
3: if $b == 0$ then
4: return $\text{lit}(\Phi)$
5: end if
6: for $\phi \in \Phi$ do
7: $\Psi = \Psi \cup \{\phi \in \Phi \mid \exists x_1, x_2, x_1 \neq x_2, \lambda(x_1), \lambda(x_2) \models \phi\}$
8: $\overline{\text{BL}}_u(\Phi) = \overline{\text{BL}}_u(\Phi) \cup \{x \in \text{lit}(\Phi) \mid \forall \phi \in \Phi : \lambda(x) \models \phi \implies \phi \in \Psi\}$
9: end for
10: return $\overline{\text{BL}}_u(\Phi)$

Theorem

If a literal $l \in \overline{\text{BL}}_u$, $l$ is not a backbone of the given formula.
A clause is put into $\Psi$ iff there exists at least two satisfy literals.

A literal is put into $\overline{BL}_u$ iff it only satisfy clauses in $\Psi$.

Heuristic strategies

- DFS, choose a chain of literals and change the assignment of them one by one.
  - literal-clause coverage
  - literal weight given by configuration or SAT solvers
Algorithm 2 Extend non-backbones estimation

1: $\overline{BC} = \text{HDBS}(\Phi) = \emptyset$
2: $\overline{BL}_e = \overline{BL}_u$
3: $(b, \lambda) = \text{SAT}(\Phi)$
4: if $b == 0$ then
   5: return $\text{lit}(\Phi)$
5: end if
6: for $\phi \in \Phi$ do
5: $BC = BC \cup \{\phi \in \Phi \mid \exists x \in \phi, x \in \overline{BL}_u\}$
7: $BL_e(\Phi) = BL_e(\Phi) \cup \{x \in \text{lit}(\Phi) \mid \forall \phi \in \Phi : \lambda(x) \models \phi \Rightarrow \phi \in BC\}$
8: end for
9: for $BL_e$ is updating do
10: $BC = BC \cup \{\phi \in \Phi \setminus BC \mid \exists x \in \overline{BL}_e(\Phi) \lor \exists \neg x \in \overline{BL}_e, x \in \text{lit}(\phi)\}$
11: $BL_e(\Phi) = BL_e(\Phi) \cup \{x \in \text{lit}(\Phi) \mid \forall \phi \in \Phi : \lambda(x) \models \phi \Rightarrow \phi \in \Psi\}$
12: end for
13: return $\overline{BL}_e$
• $\overline{BC}$ is the extension of $\Psi$
• $\overline{BL}_e$ is the extension of $\overline{BL}_u$
• A clause $\phi$ is in $\overline{BC}$ iff it contains at least one literal $l \in \overline{BL}_e$ or its negation $\neg l \in \overline{BL}_e$
• A literal $l$ and its negation is in $\overline{BL}_e$ iff $l$ only appears in clauses that belong to $\overline{BC}$.
• $\overline{BC}$ and $\overline{BL}_e$ are updated iteratively.
Example

Formula:

\[(\neg x_1 \lor x_2 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3) \land (\neg x_1 \lor x_2 \lor x_4) \land (x_4) \land (x_1 \lor x_4)\]

Model: \(x_1 \land x_2 \land x_3 \land x_4\)

Backbones: \(x_4\)

Figure: Conflicts with extension
Example

Formula: \((a \lor \neg b) \land (b \lor \neg c) \land (c \lor \neg a)\)
Model: \(a \land b \land c\)
BL\(u\): \(\emptyset\)

No backbones!
New model: \(\neg a \land \neg b \land \neg c\)
Algorithm 3 Find Model Rotation Chain

1: for $x \in \text{HDBS}(\Phi)$ do
2: \hspace{1em} $k = 1$
3: \hspace{1em} $\Phi^0_x = \{x\}$
4: end for
5: for $\Phi^k_{lx} \subset \Phi \setminus \overline{BC}$ do
6: \hspace{1em} $\Phi^k_x = \{\phi \in \Phi \setminus \overline{BC} \mid \neg \text{lit}(\Phi^{k-1}_x) \in \phi\}$
7: \hspace{1em} if $\neg x \in \text{lit}(\Phi^k_x)$ then
8: \hspace{2em} HDBS($\Phi$) = HDBS($\Phi$) \setminus $\{x\}$, Break
9: \hspace{2em} $k = k + 1$
10: end if
11: end for
12: return HDBS($\Phi$)
Figure: Total time use for SAT Comp. Benchmark
Definition (Maximal Satisfiable Subset)

Given an unsatisfied propositional formula $\Phi$, $\Phi' \subset \Phi$ is a MSS, iff $\Phi'$ is satisfiable and $\Phi' \land \phi$, $\phi \in \Phi \setminus \Phi'$ is unsatisfiable.

- UUF250 family from SATLIB
- Using LBX tool
- Generate hard and dense backbones formulae
- 1065 clauses, 250 variables.
MSS Formulae

Figure: Total time use for formulae from MCS computing

Figure: Total time use for MSS formulae
We present a novel approach to compute backbones of propositional formulae using estimations of backbones and non-backbones. Experiments show that our approach is compatible with the state-of-art approaches on backbones computing.
Thank you