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

Yueling Zhang[1] Min Zhang[1] Geguang Pu[1] Fu Song[2]
Jianwen Li[1,3]

[1]Shanghai Key Laboratory of Trustworthy Computing / East China Normal University
[2] Shanghai Tech University
[3] Rice University

ylzhang@sei.ecnu.edu.cn

August 22, 2016

Overview

Backbones

Definition (Model)

Models of a propositional formula Φ are truth assignments that make the formula to TRUE.

Definition (Backbones)

Backbones of a propositional formula Φ are literals that are true in every model

Definition (Satisfied Literal)

Given a model λ, 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

$$\underbrace{(a \lor \neg b \lor c)}_{} \land \overset{\text{Clause}}{\boxed{(a \lor b \lor \neg c)}} \land (\overset{\text{Literal}}{a} \lor \overset{\text{Literal}}{\neg b} \lor \neg c) \land (a \lor b \lor c)$$

Backbones

Example

CNF: $(a \vee \neg b \vee c) \wedge (a \vee b \vee \neg c) \wedge (a \vee \neg b \vee \neg c) \wedge (a \vee b \vee c)$
Model Table:

a	b	c
1	0	0
1	0	1
1	1	0
1	1	1

Given a model $a \wedge b \wedge c$, Literal a, c is called the satisfied literals of clause $(a \vee \neg b \vee c)$.

Motivations

- Upper bound for number of models
- Product configuration

Example

gas engine $\vee$ electric engine
electric engine $\Longrightarrow$ automatic
$\neg$ automatic $\vee \neg$ manual
electric engine
Backbones: *automatic*, $\neg$*manual*

Existing Approaches

- Iterative SAT Testing
  - $\Phi \wedge l$ is SAT, l is not a backbone
  - $\Phi \wedge l$ is UNSAT, the negation of l is a backbone.
- Upper Bound Estimation
  - $\Phi \wedge \phi$ is SAT, using $\phi \cup \lambda$ to remove non-backbones
  - $\Phi \wedge \phi$ is UNSAT, Backbones estimation terminates.
- Core Based SAT Testing
  - Φ is SAT under the assumption of $\phi, \phi \subseteq \neg\lambda$, none of the literals in λ is backbones.
  - Φ is UNSAT under the assumption of ϕ, if the core length is 1, the negation of core is a backbone.

Multi Estimations Based Backbones Extracting

- Non-Backbones under-approximation, $\overline{\mathrm{BL}}_u$
- Backbones estimation, HDBS
- Iterative test literals in HDBS

Non-Backbone Estimation $\overline{\mathrm{BL}}_u$

Algorithm 1 Under-approximation of Non-Backbones

$\Psi = \overline{\mathrm{BL}}_u(\Phi) = \emptyset$
$(b, \lambda) = \mathrm{SAT}(\Phi)$
if $b == 0$ **then**
  return $\mathrm{lit}(\Phi)$
end if
for $\phi \in \Phi$ **do**
  $\Psi = \Psi \cup \{\phi \in \Phi \mid \exists x_1, x_2, x_1 \neq x_2, \lambda(x_1), \lambda(x_2) \models \phi\}$
  $\overline{\mathrm{BL}}_u(\Phi) = \overline{\mathrm{BL}}_u(\Phi) \cup \{x \in \mathrm{lit}(\Phi) \mid \forall \phi \in \Phi : \lambda(x) \models \phi \implies \phi \in \Psi\}$
end for
return $\overline{\mathrm{BL}}_u(\Phi)$

Theorem

If a literal $l \in \overline{BL}_u$, l is not a backbone of the given formula.

Under-approximation of Non-Backbones

- A clause is put into Ψ iff there exists at least two satisfy literals
- A literal is put into $\overline{\mathrm{BL}}_u$ iff it only satisfy clauses in Ψ.
- 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

Backbones Estimation

Algorithm 2 Extend non-backbones estimation

$\overline{\mathrm{BC}} = \mathrm{HDBS}(\Phi) = \emptyset$

$\overline{\mathrm{BL}}_e = \overline{\mathrm{BL}}_u$

$(b, \lambda) = \mathrm{SAT}(\Phi)$

if $b == 0$ **then**

  return $\mathrm{lit}(\Phi)$

end if

for $\phi \in \Phi$ **do**

  $\overline{\mathrm{BC}} = \overline{\mathrm{BC}} \cup \{\phi \in \Phi \mid \exists x \in \phi, x \in \overline{\mathrm{BL}}_u\}$

  $\overline{\mathrm{BL}}_e(\Phi) = \overline{\mathrm{BL}}_e(\Phi) \cup \{x \in \mathrm{lit}(\Phi) \mid \forall \phi \in \Phi : \lambda(x) \models \phi \implies \phi \in \overline{\mathrm{BC}}\}$

end for

for $\overline{\mathrm{BL}}_e$ is updating **do**

  $\overline{\mathrm{BC}} = \overline{\mathrm{BC}} \cup \{\phi \in \Phi \setminus \overline{\mathrm{BC}} \mid \exists x \in \overline{\mathrm{BL}}_e(\Phi) \vee \exists \neg x \in \overline{\mathrm{BL}}_e, x \in \mathrm{lit}(\phi)\}$

  $\overline{\mathrm{BL}}_e(\Phi) = \overline{\mathrm{BL}}_e(\Phi) \cup \{x \in \mathrm{lit}(\Phi) \mid \forall \phi \in \Phi : \lambda(x) \models \phi \implies \phi \in \Psi\}$

end for

return $\overline{\mathrm{BL}}_e$

Backbones Estimation

- $\overline{\mathrm{BC}}$ is the extension of Ψ
- $\overline{\mathrm{BL}}_e$ is the extension of $\overline{\mathrm{BL}}_u$
- a clause ϕ is in $\overline{\mathrm{BC}}$ iff it contains at least one literal $l \in \overline{\mathrm{BL}}_e$ or its negation $\neg l \in \overline{\mathrm{BL}}_e$
- a literal l and its negation is in $\overline{\mathrm{BL}}_e$ iff l only appears in clauses that belongs to $\overline{\mathrm{BC}}$.
- $\overline{\mathrm{BC}}$ and $\overline{\mathrm{BL}}_e$ are updated iteratively.

Example

Formula:

$(\neg x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (x_4) \wedge (x_1 \vee x_4)$

Model: $x_1 \wedge x_2 \wedge x_3 \wedge x_4$

Backbones: x_4

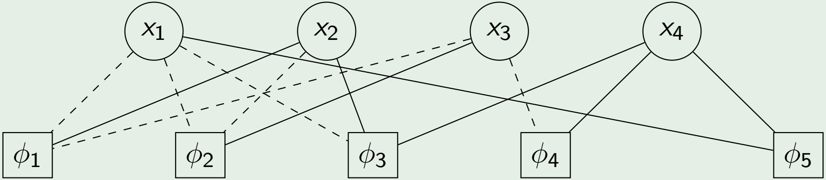

Figure: Conflicts with extension

Example

Formula: $(a \vee \neg b) \wedge (b \vee \neg c) \wedge (c \vee \neg a)$
Model: $a \wedge b \wedge c$
$\overline{\mathrm{BL}}_u$: $\emptyset$

No backbones!
New model: $\neg a \wedge \neg b \wedge \neg c$

Backbones Estimation

Algorithm 3 Find Model Rotation Chain

for $x \in \mathrm{HDBS}(\Phi)$ **do**
  $k = 1$
  $\Phi_x^0 = \{x\}$
end for
for $\Phi_{l_x}^k \subset \Phi \setminus \overline{\mathrm{BC}}$ **do**
  $\Phi_x^k = \{\phi \in \Phi \setminus \overline{\mathrm{BC}} \mid \neg \mathrm{lit}(\Phi_x^{k-1}) \in \phi\}$
  if $\neg x \in \mathrm{lit}(\Phi_x^k)$ **then**
    $\mathrm{HDBS}(\Phi) = \mathrm{HDBS}(\Phi) \setminus \{x\}$, *Break*
    $k = k + 1$
  end if
end for
return $\mathrm{HDBS}(\Phi)$

SAT Comp. Benchmark

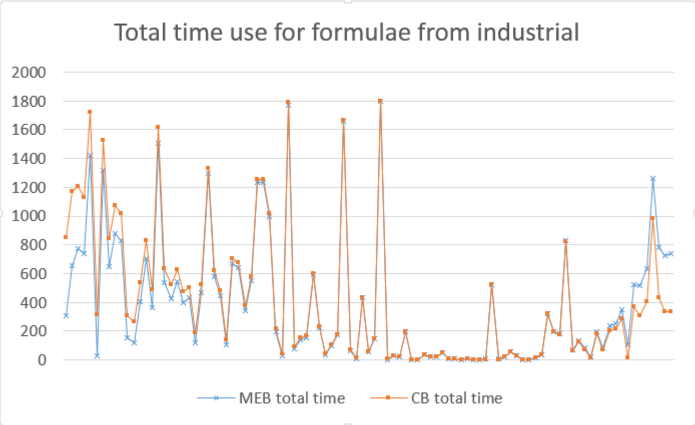

Figure: Total time use for SAT Comp. Benchmark

MSS Formulae

Definition (Maximal Satisfiable Subset)

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

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

MSS Formulae

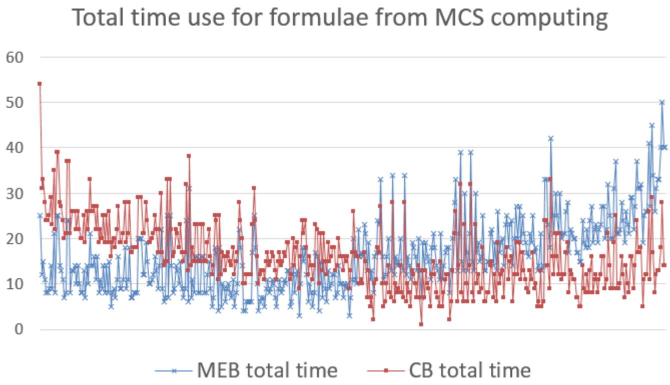

Figure: Total time use for MSS formulae

Conclusion

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