SAT-Based Analysis of Feature Models is Easy

Marcilio Mendonca\textsuperscript{1}, Andrzej Wasowski\textsuperscript{2}, Krzysztof Czarnecki\textsuperscript{1}

\textsuperscript{1}University of Waterloo, Canada
\textsuperscript{2}IT University of Copenhagen, Denmark

SPLC, Aug 2009 – San Francisco, USA
Outline

- Feature Models
- Feature Model Analysis
- SAT-Based Analysis
- Research Goal
- Hardness of SAT Problems
- Hardness of Feature Model SAT Problems
- Conclusion and Research Impact
Feature Models

Representing the similarities and differences within a system family

Web Search Engine Product Line

My Search Engine
- doc-type
- html
- gif X
- search-by-language X

Web Search Domain
- French
- English
- doc-type
- html
- image
- gif
- video

search-by-language → page-translation
page-preview → ¬svg
Feature Model Analysis

Constructing and Maintaining feature models can be a laborious task.

Contradiction!

Extra Constraints:
C1: A \( \text{xor} \) B
C2: \( \neg C \rightarrow \neg B \)

Model is Inconsistent!

“Dead” features

Constraint C2 removed

Extra Constraints:
C1: A \( \text{xor} \) B

Model contains “dead” features

Automated support is required for Debugging feature models
Feature Model Analysis

Product Configuration

Extra Constraints:
C1: C \texttt{xor} D

**Propagation** of R selection

**Propagation** of C deselection

**Toggling** D selection

Automated support is required for **Configuring** feature models
Feature Model Analysis
Automated Reasoning and Configuration

• A long list …
  • Checking consistency
  • Detecting dead/common features
  • Filtering/listing configurations
  • Computing valid domains
  • Resolving conflicts
  • Counting configurations
  • Checking refactoring/edits (e.g., equivalence, implication, difference)
  • Computing metrics (e.g., variability degree, commonality of a feature)
• …
Feature Model Analysis
An effective strategy to automate FM analysis is the use of formalisms

- Feature Model
- Formal Encoding
- Reasoner 1
- Reasoner 2
- ... Reasoner N

(e.g., Boolean formula)
(e.g., SAT solver)
SAT-Based Analysis
Encoding feature models as Boolean formulas

\[ f = (R) \text{ and } (\neg A \text{ or } R) \text{ and } (\neg B \text{ or } R) \text{ and } (\neg R \text{ or } B) \]

• Boolean formula
  • Variables; Domain = \{true, false\}
  • Logic operators = \{and, or, not, implication, bi-implication\}

• CNF formula (conjunctive normal form)
  • Disjunction of clauses; Clause is a conjunction of literals;
  • Literal is a variable or its negation

• Satisfiability Problem
  • Is there an assignment to \(f\)'s variables that evaluates \(f\) to true?
  • If so, \(f\) is satisfiable, otherwise \(f\) is unsatisfiable

• SAT solvers
  • Systems that check formula satisfiability
  • E.g., \(f\) is satisfiable (R=A=B=true)
SAT-Based Analysis

Encoding feature models as Boolean formulas

Feature Model

\[ f = (R) \land (\lnot A \lor R) \land (\lnot B \lor R) \land (\lnot R \lor B) \]

Analysis

<table>
<thead>
<tr>
<th>Consistency</th>
<th>Yes!</th>
<th>( f ) is SAT (e.g., ( R=true, A=true, B=true ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dead Features</td>
<td>None!</td>
<td>( (f \land A) ) is SAT ( \rightarrow A ) is dead</td>
</tr>
<tr>
<td>Common Features</td>
<td>{R,B}</td>
<td>( (f \land \lnot B) ) is UNSAT ( \rightarrow B ) is common</td>
</tr>
<tr>
<td>Valid Domains</td>
<td></td>
<td>( R={true}, A={true, false}, B={true} )</td>
</tr>
<tr>
<td>List Configurations</td>
<td>1. {R,A,B}, 2. {R,\lnot A,B}</td>
<td>...</td>
</tr>
</tbody>
</table>

...
Research Goal and Relevance

Problem
Continuous emergence of SAT-based tools in the SPL field (e.g. configurators, reasoners) however the fundamental underlying problem tacked (SAT) is well-known to be hard

➤ Satisfiability is NP-complete (e.g., 3-SAT)

Therefore, it is fair to ask…
• How well do SAT-based reasoners and configurators scale?
• Can we rely on the efficiency of SAT-based tools?
Research Goal and Relevance

Good news!
• Some SAT problems are tractable (e.g., 2-SAT)
• Yet for others tractability boundaries are known (e.g., 3-SAT)

Relevant Research Question:

Are SAT problems induced from feature models tractable as observed by researchers? If so, what evidences support this fact?
Hardness of 3-SAT Problems

Random 3-CNF Formula

\[ (\overline{A} \text{ or } B \text{ or } \overline{C}) \text{ and } (A \text{ or } D \text{ or } C) \text{ and } (D \text{ or } \overline{C} \text{ or } B) \]

Clauses has exactly 3 literals

Clause Generation:

- Select 3 distinct variables randomly
- Negate each with probability 50%
- Create a distinct clause

**Clause density**: \#clauses / \#variables = 3/4 = 0.75

- Low density \rightarrow under-constrained
- High density \rightarrow over-constrained
Hardness of SAT Problems
What we known about the hardness of 3-SAT problems?

Hardness of Uniform Random 3-SAT Problems

“easy” ——— “hard” ——— “easy”

Under-constrained

Phase Transition

Over-constrained

Number of Steps Taken by the SAT Solver

Clause Density

SAT

3

3.42

3.5

4

4.25

4.506

5

5.5

SAT/UNSAT

UNSAT
Hardness of Feature Model SAT Problems
Properties of Feature Models as SAT problems

Feature Tree:

\[ Fm = Ft \text{ and } Ctc \]

Cross-tree Constraints:

C1: A xor B
C2: \( \neg C \rightarrow \neg B \)

Facts
- Ft is larger (# of variables) than Ctc
- Ft is structured (hierarchy of variables)
- Ctc is an arbitrary propositional formula
Hardness of Feature Model SAT Problems
Satisfiability properties of feature tree formulas

A well-formed feature tree is always satisfiable
» Complexity: O(1)

SAT solver complexity: O(n)
» Backtrack-free

The satisfiability of feature trees can be checked in linear time by a SAT solver!
Hardness of Feature Model SAT Problems

The cross-tree constraints are arbitrary propositional formulas

\[ F_m = F_t \text{ and } C_{tc} \]

linear time

\(? (\text{empirical analysis})\)

Critical Question: Can the CTC formula cause the feature model SAT problem to be intractable?

What is the representativeness of the Ctc?

\[ \text{CTCR} = \text{vars}(C_{tc}) / \text{vars}(F_t) \]
Hardness of Feature Model SAT Problems
3-CNF feature models

3-CNF-FM = Ft and 3-CNF

Facts about 20 real feature models studied:
1. CTCR in is usually < 30%
2. Clause density is usually < 2
3. Cross-tree constraints are usually a conjunction of binary & ternary clauses
4. Model sizes usually lower than 1,000 features

Easier SAT Problems

Facts about 3-CNF feature models used in the experiments:
1. CTCR values of up to 30%
2. Clause density values of up to 7
3. All clauses have 3 variables
4. Model sizes up to 10,000 features

Harder SAT Problems

and also …

hardness parameters are known for random 3-SAT problems
Hardness of Feature Model SAT Problems

Phase transition cross-over points for 3-CNF-FMs

Phase Transition Crosspoints were computed for Feature Models (3-CNF-FMs)
Hardness of Feature Model SAT Problems
Tractability Empirical Analysis

Running Time for Checking the Satisfiability of a Large Model
(average of 100 test cases for each point)

SAT solver is **efficient** during cross-tree constraints phase transition!
Conclusion

Satisfiability of feature tree SAT problems can be checked efficiently by a SAT solver!
Conclusion

Satisfiability of 3-CNF-FM SAT problems can be checked efficiently by a SAT solver!
Research Impact

- Directly Benefits Tool Developers!
  - Increased confidence in the use of SAT technology
  - Encouragement for new features based on SAT
  - Well-built evidence of tool efficiency and scalability

- Directly Benefits Tool End-users!
  - More efficient reasoners
  - More efficient configurators
  - Improved scalability
Want an Example?

Software Product Lines Online Tools

- Web-based reasoning & configuration system
- Includes a repository of Feature Models

SPLIT

(www.splot-research.org)
# Weather_Station (23 features)

## Configuration Steps

<table>
<thead>
<tr>
<th>Step</th>
<th>Decision</th>
<th>#Decisions (cumulative)</th>
<th>#Propagations (at step)</th>
<th>SAT checks (at step)</th>
<th>SAT time (at step)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Weather Station</td>
<td>6 (26.1%)</td>
<td>5</td>
<td>9</td>
<td>1 ms</td>
</tr>
<tr>
<td>2</td>
<td>Text</td>
<td>10 (43.5%)</td>
<td>3</td>
<td>6</td>
<td>1 ms</td>
</tr>
<tr>
<td>3</td>
<td>Demo</td>
<td>11 (47.8%)</td>
<td>0</td>
<td>2</td>
<td>0 ms</td>
</tr>
<tr>
<td>4</td>
<td>Wnd Speed</td>
<td>12 (52.2%)</td>
<td>0</td>
<td>2</td>
<td>0 ms</td>
</tr>
<tr>
<td>5</td>
<td>Pressure</td>
<td>13 (56.6%)</td>
<td>0</td>
<td>3</td>
<td>0 ms</td>
</tr>
<tr>
<td>6</td>
<td>Temperature</td>
<td>15 (65.2%)</td>
<td>1</td>
<td>3</td>
<td>0 ms</td>
</tr>
<tr>
<td>7</td>
<td>English</td>
<td>17 (73.9%)</td>
<td>1</td>
<td>3</td>
<td>0 ms</td>
</tr>
<tr>
<td>8</td>
<td>Alarm</td>
<td>19 (82.6%)</td>
<td>1</td>
<td>3</td>
<td>0 ms</td>
</tr>
<tr>
<td>9</td>
<td>auto-completion</td>
<td>23 (100.0%)</td>
<td>4</td>
<td>1</td>
<td>0 ms</td>
</tr>
</tbody>
</table>

Done! (Export configuration: CSV file | XML)

---

**SPLOT**

(www.splot-research.org)
Thank you!

Questions?