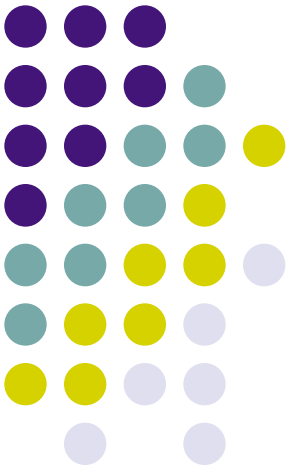
SAT-Based Analysis of Feature Models is Easy

Marcilio Mendonca[1], Andrzej Wasowski[2], Krzysztof Czarnecki[1]

[1]University of Waterloo, Canada
[2]IT University of Copenhagen, Denmark



SPLC, Aug 2009 – San Francisco, USA

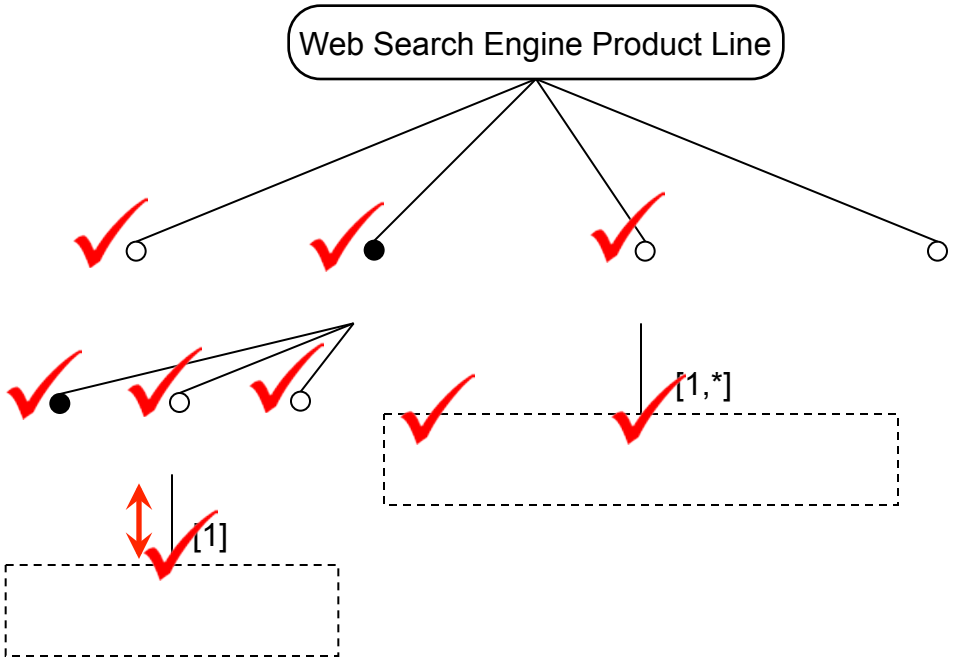
Outline

UNIVERSITY OF Waterloo

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

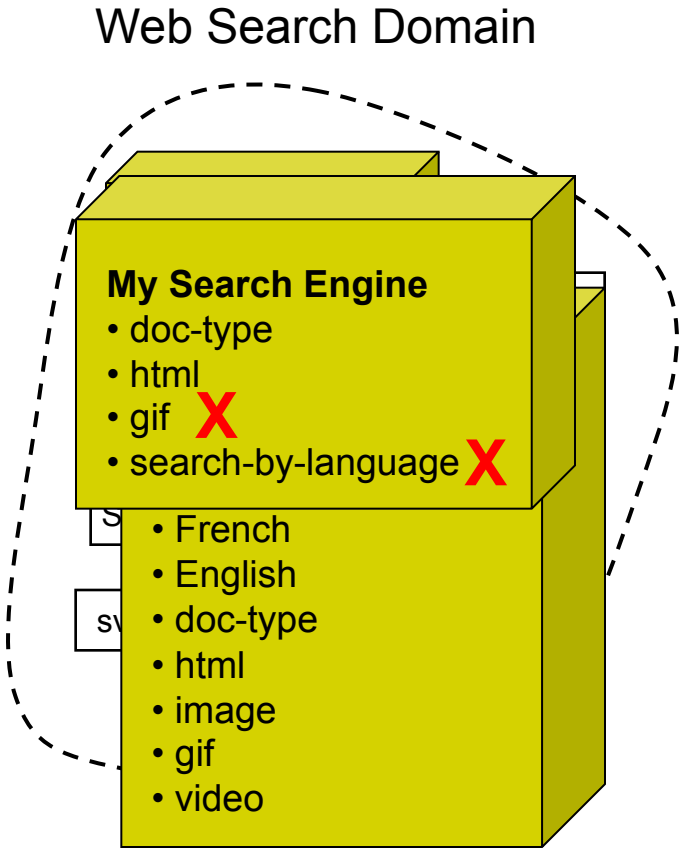
Feature Models

Representing the similarities and differences within a system family



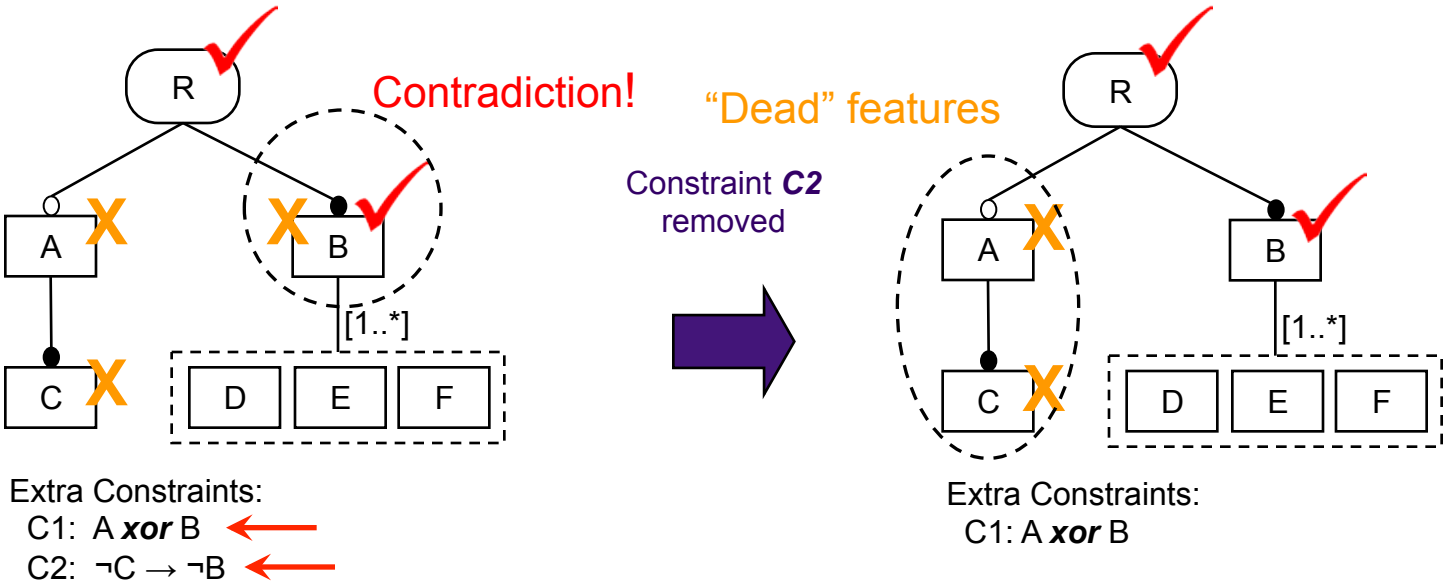
search-by-language ⟶ page-translation ⟵

page-preview ⟶ ¬svg



Feature Model Analysis

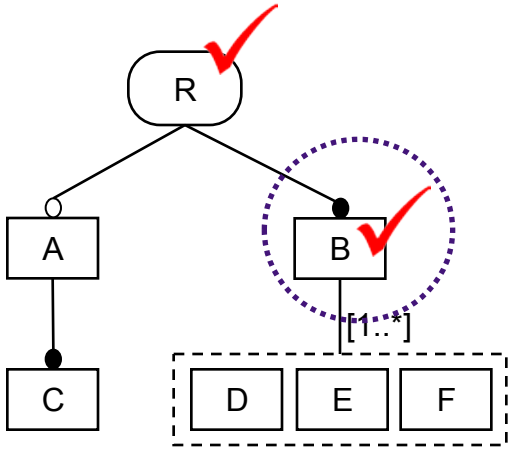
Constructing and Maintaining feature models can be a laborious task



Automated support is required for **Debugging** feature models

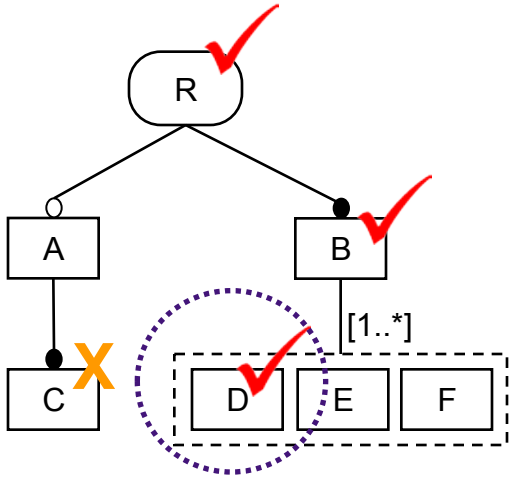
Feature Model Analysis

Product Configuration



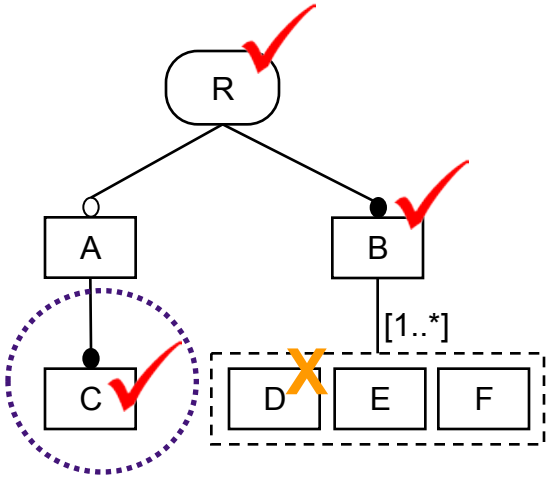
Extra Constraints:
C1: C ***xor*** D

Propagation of R selection



Extra Constraints:
C1: C ***xor*** D

Propagation of C deselection



Extra Constraints:
C1: C ***xor*** D

Toggling D selection

Automated support is required for **Configuring** feature models

Feature Model Analysis

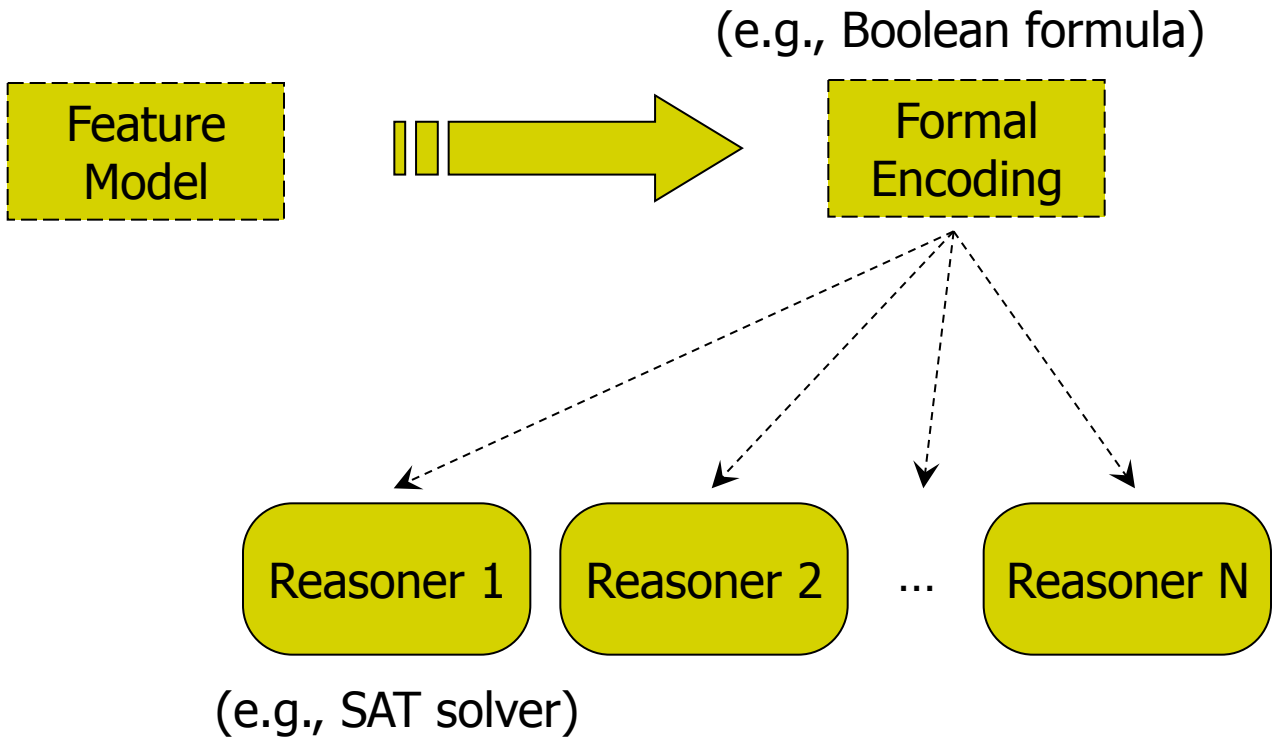
Automated Reasoning and Configuration

UNIVERSITY OF Waterloo

- 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



SAT-Based Analysis

Encoding feature models as Boolean formulas

UNIVERSITY OF Waterloo

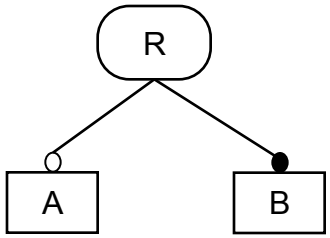
f = (**R**) and (**~A** or **R**) and (**~B** or **R**) and (**~R** or **B**)

- **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**) and (~**A** or **R**) and (~**B** or **R**) and (~**R** or **B**)

Analysis	Answer	SAT formulation
Consistency	Yes!	f is SAT (e.g., R=true, A=true, B=true)
Dead Features	None!	(f & A) is SAT → A is dead
Common Features	{R,B}	(f & ~B) is UNSAT → B is common
Valid Domains	R={true}, A={true, false}, B={true}	…
List Configurations	1. {R,A,B}, 2. {R,~A,B}	…

…

Research Goal and Relevance

UNIVERSITY OF Waterloo

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

UNIVERSITY OF Waterloo

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

UNIVERSITY OF Waterloo

Random 3-CNF Formula

$(\overline{A}$ *or* B *or* $\overline{C})$ *and* $(\overline{A}$ *or* $\overline{D}$ *or* C) *and* (D *or* $\overline{C}$ *or* $\overline{B})$

Clauses has exactly 3 literals
Clause Generation:

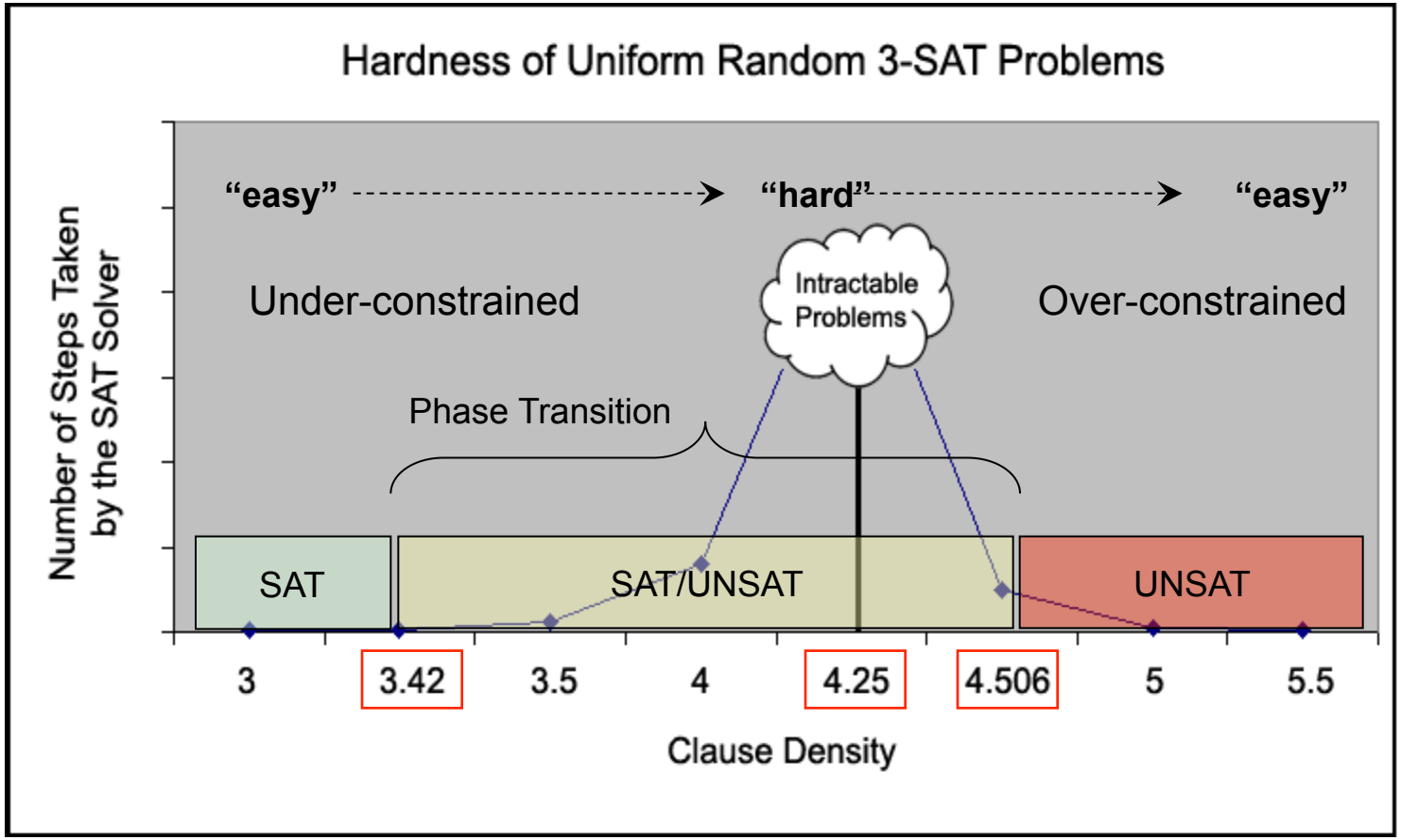
- Select 3 distinct variables randomly
- Negate each with probability 50%
- Create a distinct clause

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

- Low density → under-constrained
- High density → over-constrained

Hardness of SAT Problems

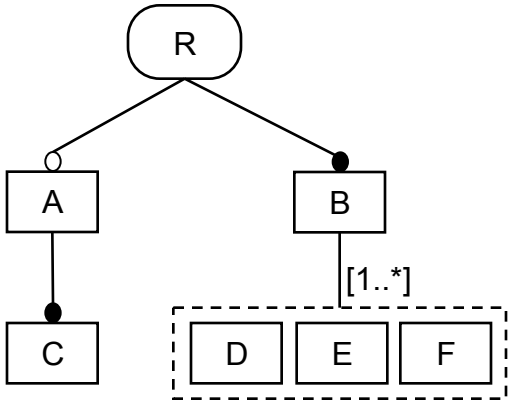
What we known about the hardness of 3-SAT problems?



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: ¬C → ¬B

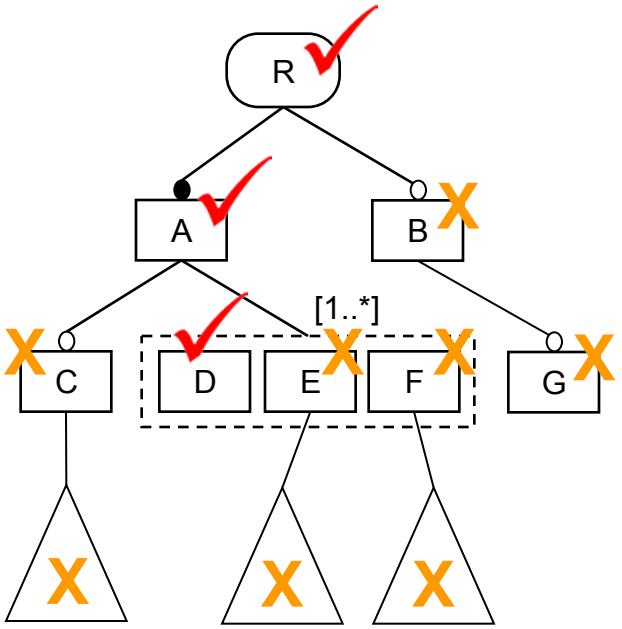
Facts

- Ft is larger (# of variables) than Ctc
- Ft is structured (hierarchy of variables)
- Ctc is an arbitrary propositional formula

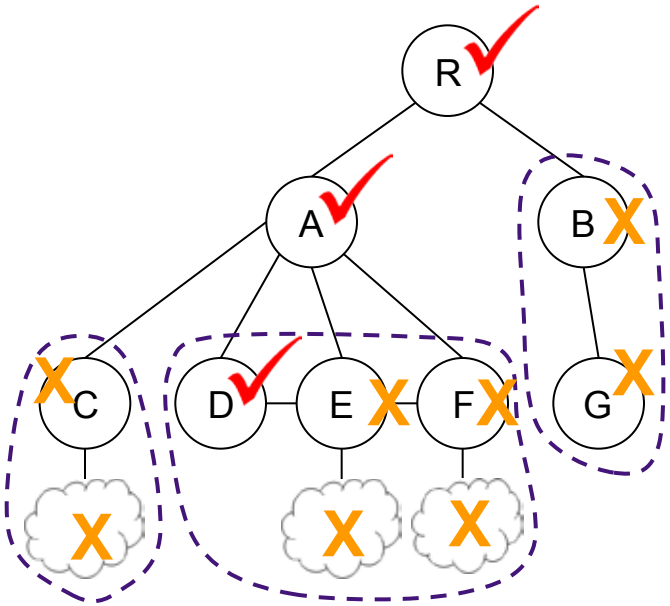
Hardness of Feature Model SAT Problems

Satisfiability properties of **feature tree** formulas

Feature Tree



Constraint Graph



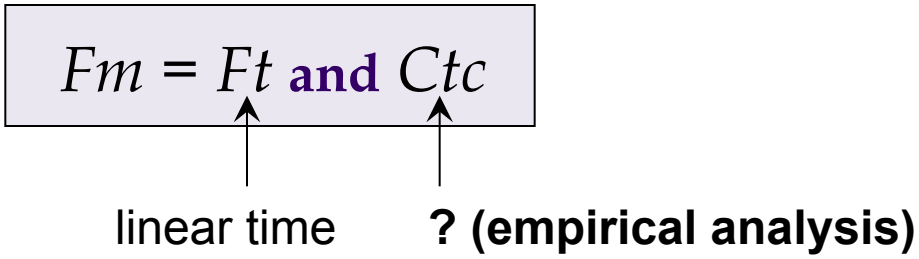
» A well-formed feature tree is always satisfiable
» Complexity: O(1)

» 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



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

What is the representativeness of the Ctc?

CTCR = vars(*Ctc*) / vars(*Ft*)

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

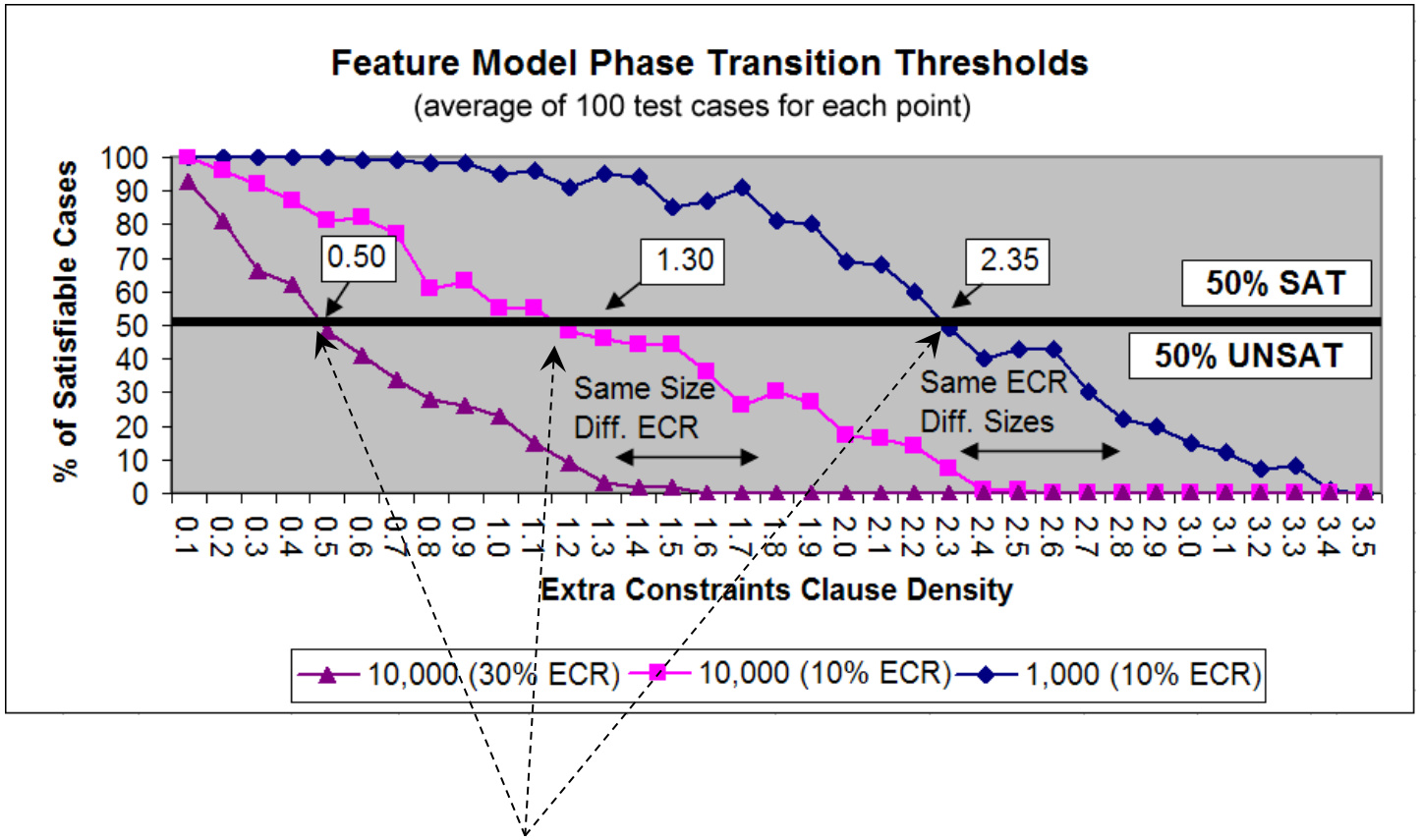
Harder SAT Problems

and also …

hardness parameters are known for random 3-SAT problems

Hardness of Feature Model SAT Problems

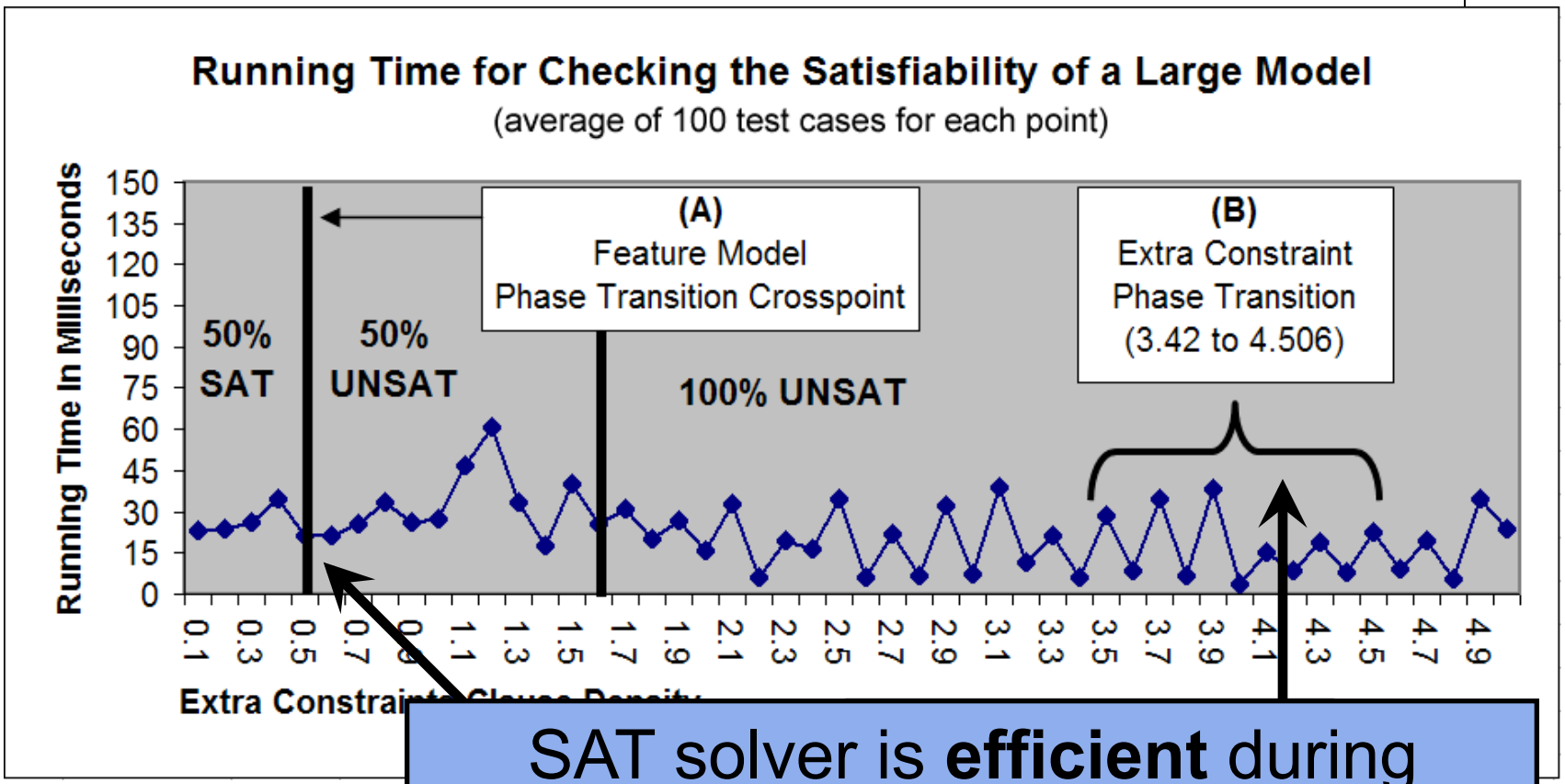
Phase transition cross-over points for 3-CNF-FMs



Phase Transition Crosspoints were computed for Feature Models (3-CNF-FMs)

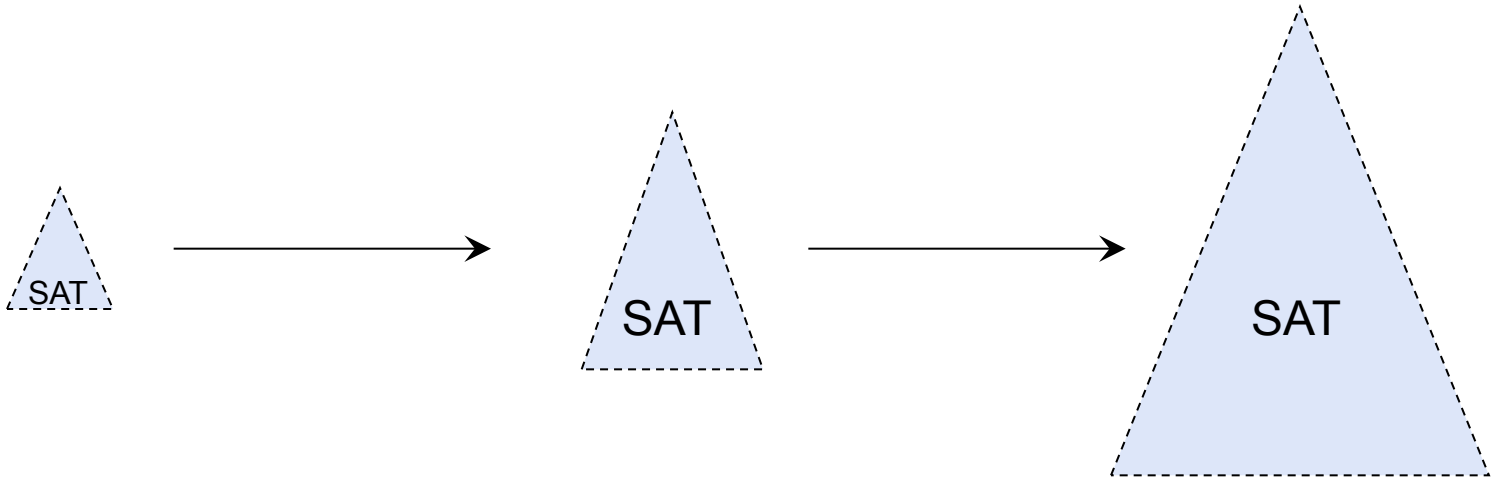
Hardness of Feature Model SAT Problems

Tractability Empirical Analysis



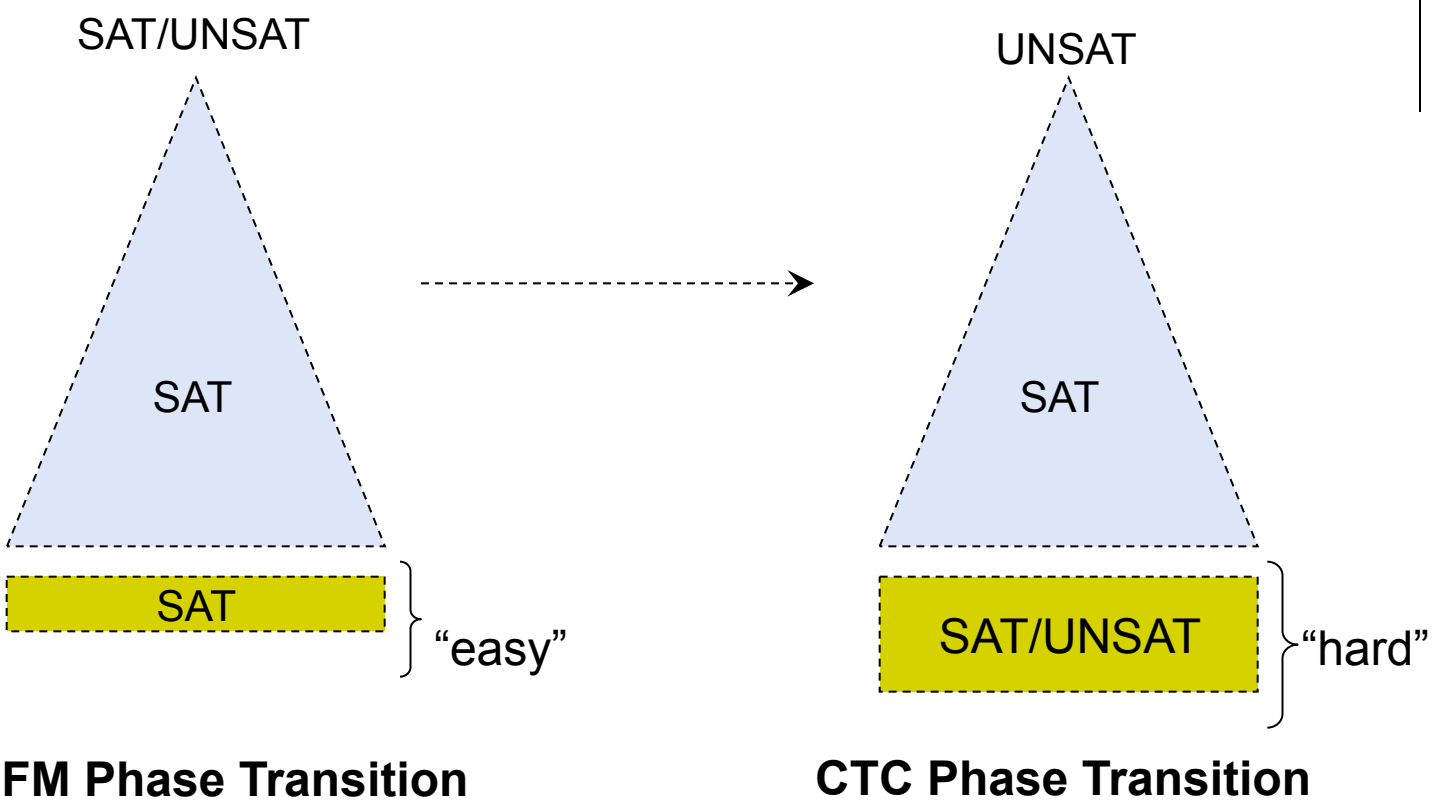
SAT solver is **efficient** during cross-tree constraints phase transition!

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

UNIVERSITY OF Waterloo

- 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**

(www.splot-research.org)

Weather_Station (23 features)

- Weather Station
  - Measurements
    - [1..3]
      - Pressure
      - ~~Temperature~~
      - ~~Wind Speed~~
  - Data Source
    - [1..1]
      - ~~Internet~~
      - ~~Demo~~
      - External Services
  - Output Format
    - [1..1]
      - Text
      - ~~Web Server~~
      - ~~File~~
  - Format
    - [1..1]
      - ~~XML~~
      - ~~HTML~~
      - Plain Text
  - Language
    - [1..1]
      - English
      - ~~German~~
  - ~~Alarm~~
    - [1..2]
      - ~~Freeze Point~~
      - ~~Storm Alert~~

Configuration Steps [reset]

100%

Step	Decision	#Decisions (cummulative)	#Propagations (at step)	#SAT checks (at step)	SAT time (at step)
1	✔ Weather Station	6 (26.1%)	5	9	1 ms
2	✔ Text	10 (43.5%)	3	6	1 ms
3	✘ Demo	11 (47.8%)	0	2	0 ms
4	✘ Wind Speed	12 (52.2%)	0	2	0 ms
5	✔ Pressure	13 (56.5%)	0	3	0 ms
6	✘ Temperature	15 (65.2%)	1	3	0 ms
7	✔ English	17 (73.9%)	1	3	0 ms
8	✘ Alarm	19 (82.6%)	1	3	0 ms
9	auto-completion	23 (100.0%)	4	1	0 ms

Done! (Export configuration: CSV file | XML)

SPLOT

(www.splot-research.org)

Thank you!

Questions?