Analyzing Industrial Feature Models with #SAT: Are we there yet?

Chico Sundermann[1], Tobias Heß[1], Michael Nieke[2], Paul M. Bittner[1], Jeffrey M. Young[3], Ina Schaefer[2], Thomas Thüm[1]

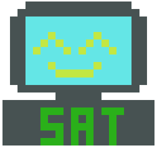
FOSD'21 | April 14, 2021

1) ulm university universität uulm

2) Technische Universität Braunschweig

3) Oregon State University

What is SAT?



Formula is satisfiable.

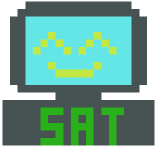
input

$$\boxed{(A \vee B) \wedge C}$$

What is #SAT?

$(A \vee B) \wedge C$

input

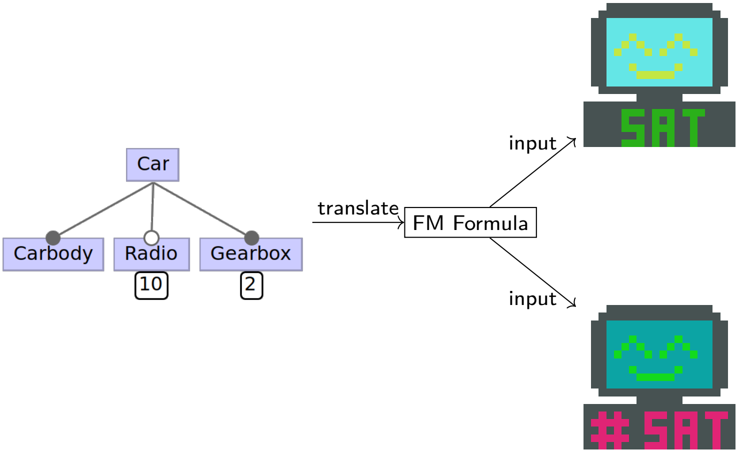


Formula is satisfiable.

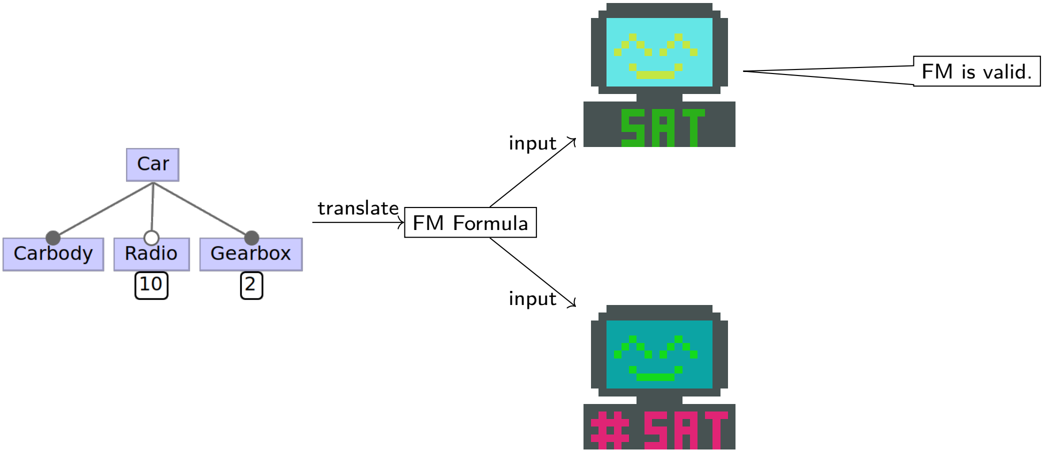
input

Formula has 3 solutions.

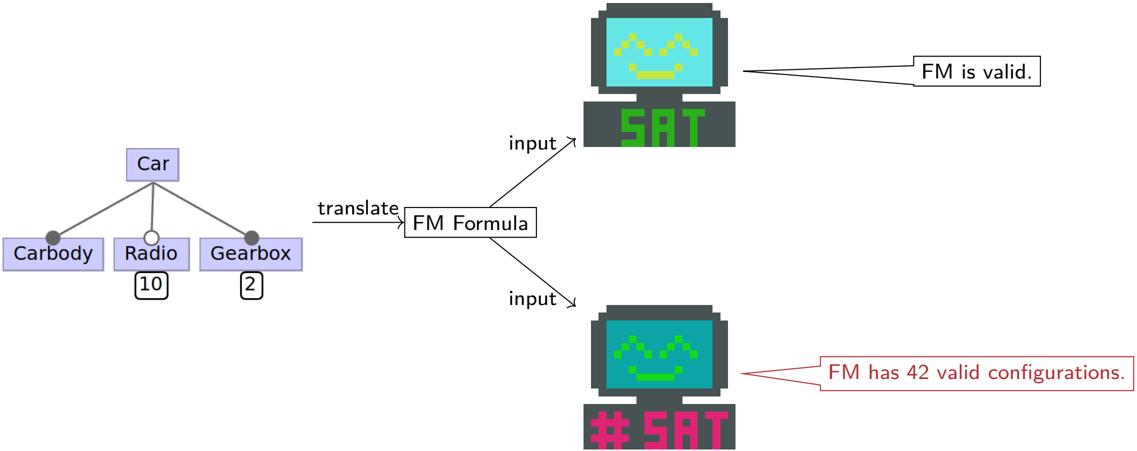
What is #SAT?


Car
Carbody
Radio
Gearbox
10
2
translate
FM Formula
input
input
SAT
#SAT

What is #SAT?


Car
Carbody
Radio
Gearbox
10
2
translate
FM Formula
input
input
SAT
FM is valid.
SAT

What is #SAT?



Car
Carbody
Radio
Gearbox
10
2
translate
FM Formula
input
input
SAT
FM is valid.
#SAT
FM has 42 valid configurations.

Variability Factor

Homogeneity

Uniform Random Sampling

Applications of #SAT Solvers on Feature Models

Chico Sundermann
University of Ulm, Germany

Michael Nieke
TU Braunschweig, Germany

Paul Maximilian Bittner
University of Ulm, Germany

Tobias Heß
University of Ulm, Germany

Thomas Thüm
University of Ulm, Germany

Ina Schaefer
TU Braunschweig, Germany

Configuration Relevance

Feature Prioritization

Rating Errors

CTC Restrictiveness

Variability Reduction

Optimize Configuring

Subset Variability

Variability Factor

Void Feature Moddel

Degree of Orthogonality

Cost Savings

Maintainability Prediction

Configuration Relevance

Rating Errors

Variability Reduction

Subset Variability

Homogeneity

Payoff Threshold

Degree of Reuse

Core, Dead & False-Optional

Atomic Set Candidates

Feature Prioritization

CTC Restrictiveness

Optimize Configuring

Uniform Random Sampling

Atomic Sets

Rate Interactions

Configuration Derivation

Variability Factor

Void Feature Moddel

Degree of Orthogonality

Cost Savings

Maintainability Prediction

Configuration Relevance

Rating Errors

Variability Reduction

Subset Variability

Homogeneity

Payoff Threshold

Degree of Reuse

Core, Dead & False-Optional

Atomic Set Candidates

Feature Prioritization

CTC Restrictiveness

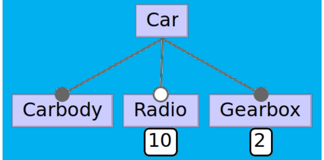
Optimize Configuring

Uniform Random Sampling

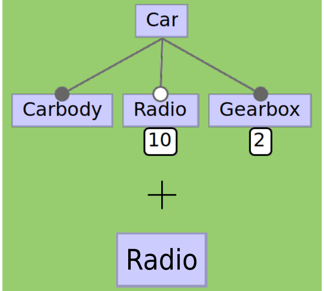
Atomic Sets

Rate Interactions

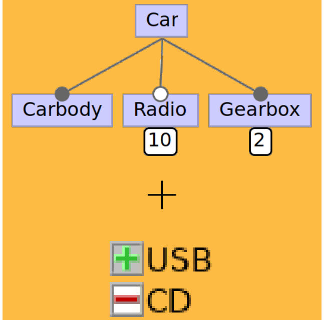
Configuration Derivation



Cardinality of
Feature Models

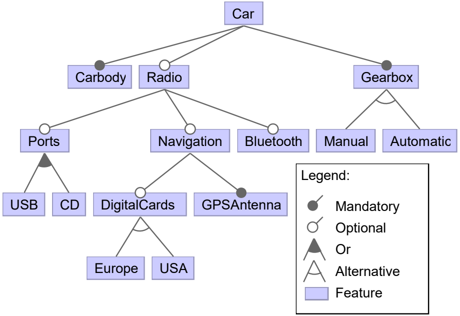


Cardinality of
Features



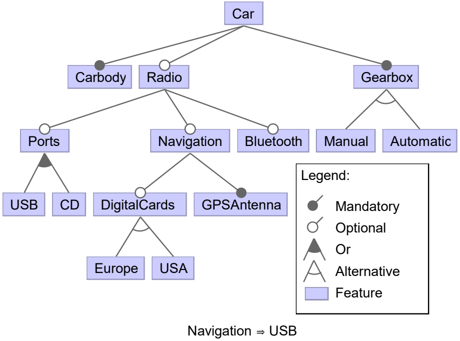
Cardinality of
Partial Configurations

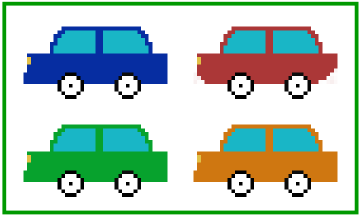
Uniform Random Sampling



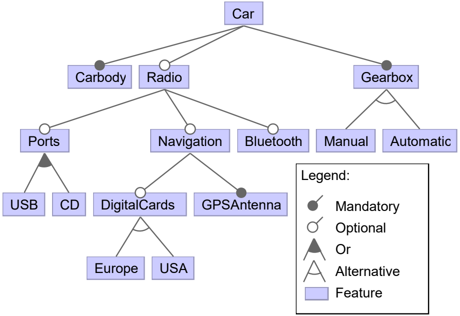
Navigation ⇒ USB

Uniform Random Sampling

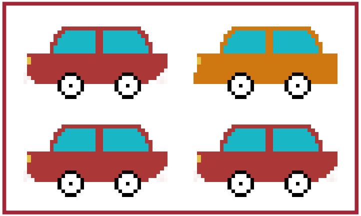




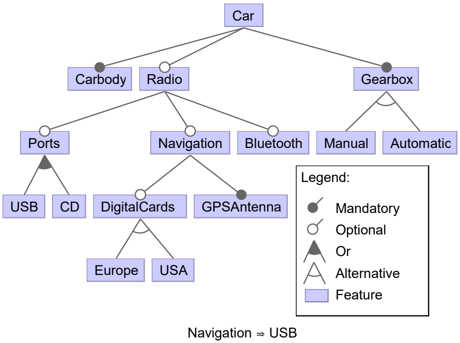
Uniform Random Sampling



Navigation ⇒ USB



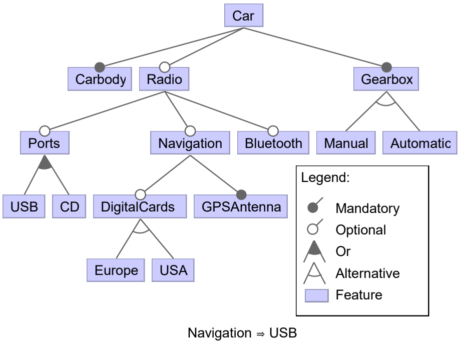
Bias



Uniform Random Sampling

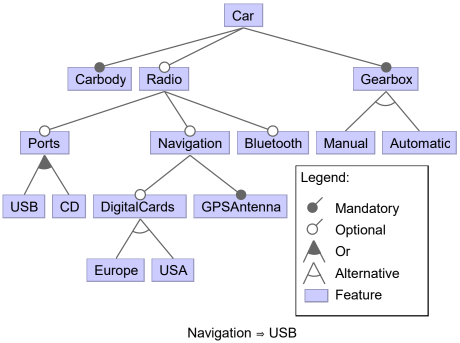
Uniform Random Sampling

- Same chance for each configuration



Uniform Random Sampling

- Same chance for each configuration
- Representative sample



Uniform Random Sampling

Uniform Random Sampling

- Same chance for each configuration
- Representative sample
- Dependent on #SAT

Experiment Design

- 17 Solvers (15 exact, 2 approximate)
  - 7 DPLL Solvers (Single query)
  - 8 Knowledge Compilers (3 d-DNNF, 3 BDD, 2 other)

DPLL

PicoSAT Relsat

SharpCDCL Cachet SharpSAT

countAntom Ganak

Knowledge Compilation

c2d d4 dShaarp

Minic2d CNF2EADT

CNF2OBDD BuDDy Cudd

,

Experiment Design

- 17 Solvers (15 exact, 2 approximate)
  - 7 DPLL Solvers (Single query)
  - 8 Knowledge Compilers (3 d-DNNF, 3 BDD, 2 other)
- 15 Subject Systems
  - 6 Evolutions
  - 373 Feature Models

Systems	#Vers.	#Feat.	#Const.
BerkeleyDB	1	76	20
axTLS	1	96	14
uClibc	1	313	56
uClinux-base	1	380	3,455
Automotive04	50	127–531	0–623
Automotive03	5	149–588	0–1,184
BusyBox	37	439–631	463–691
FinancialServices	10	557–771	1,001–1,148
Embtoolkit	1	1,179	323
CDL	116	1,178–1,408	816–956
uClinux-dist.	1	1,580	197
Automotive05	136	246–1,663	0-11,632
Automotive01	1	2,513	2,833
Linux	1	6,467	3,545
Automotive02	4	14,010–18,616	666–1,369

,

Experiment Design

- 17 Solvers (15 exact, 2 approximate)
  - 7 DPLL Solvers (Single query)
  - 8 Knowledge Compilers (3 d-DNNF, 3 BDD, 2 other)
- 15 Subject Systems
  - 6 Evolutions
  - 373 Feature Models
- Objectives
  - Scalability #SAT solvers
  - Recommendations: Solvers/Techniques
  - Future work

Insights

...

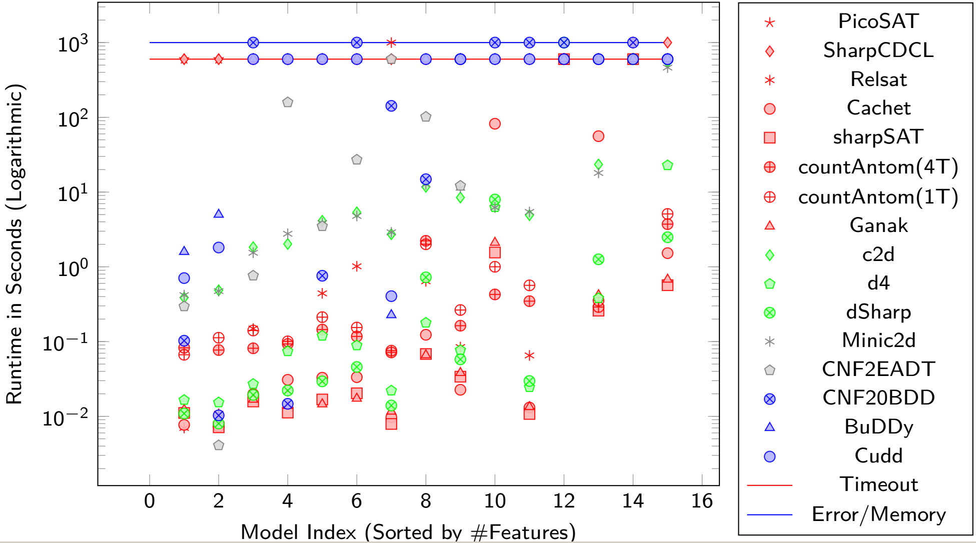
Limits

...

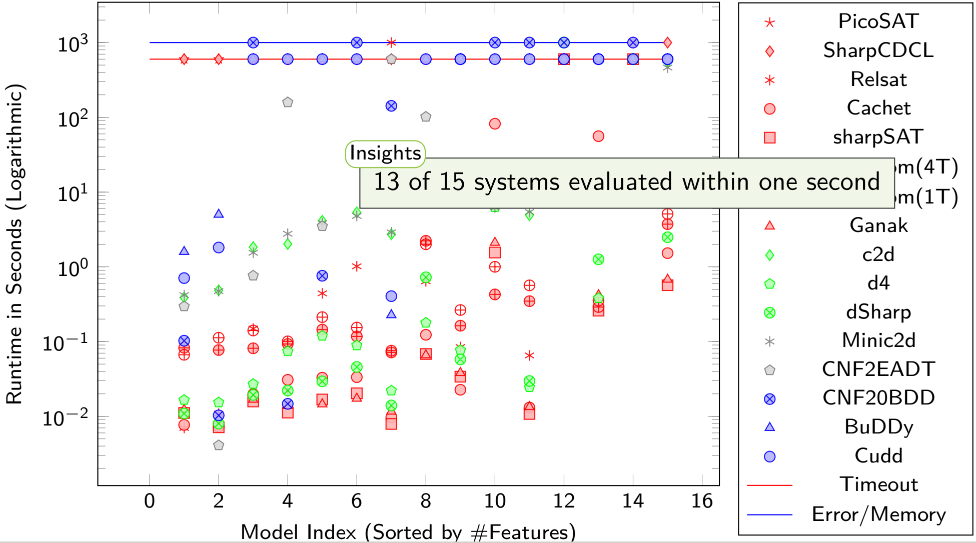
Future Work

...

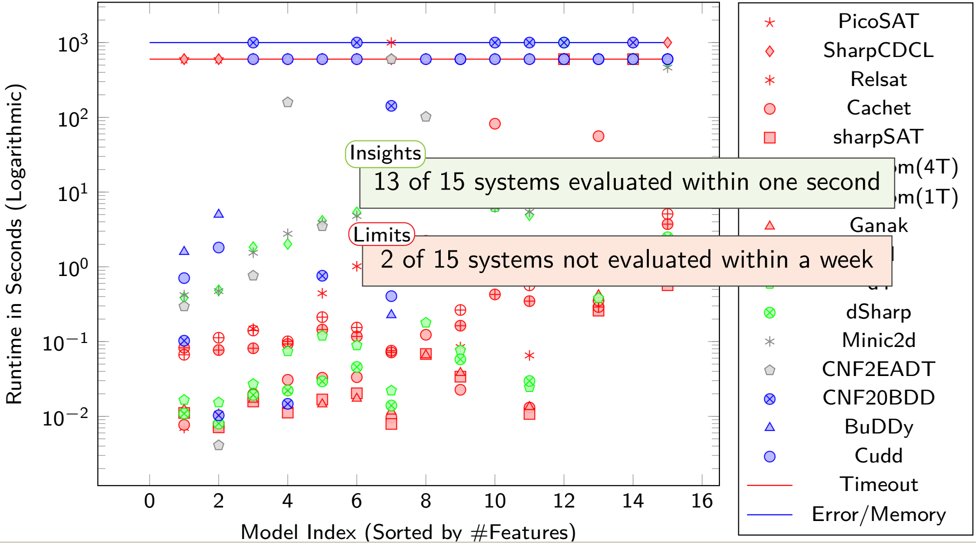
Scalability #SAT



Scalability #SAT



Scalability #SAT



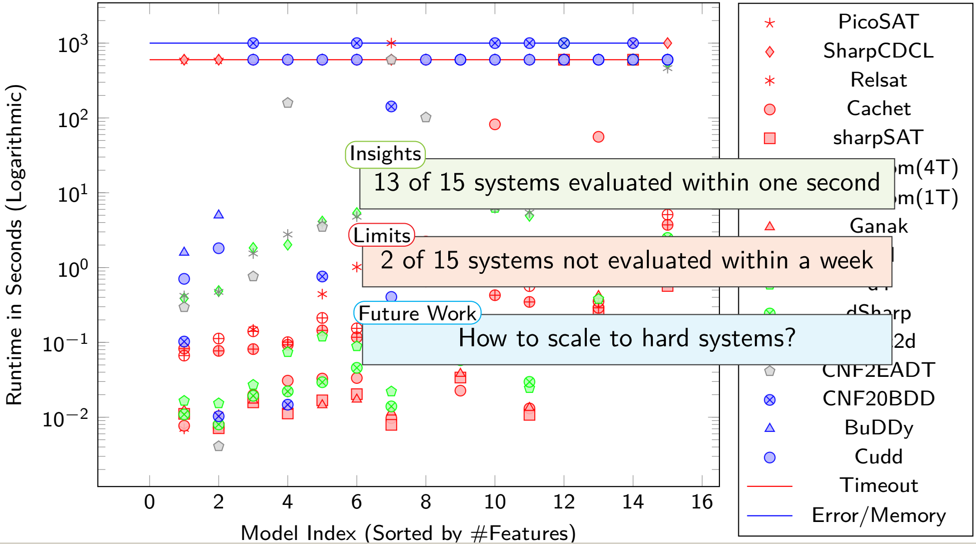
Insights

13 of 15 systems evaluated within one second

Limits

2 of 15 systems not evaluated within a week

Scalability #SAT



Insights

13 of 15 systems evaluated within one second

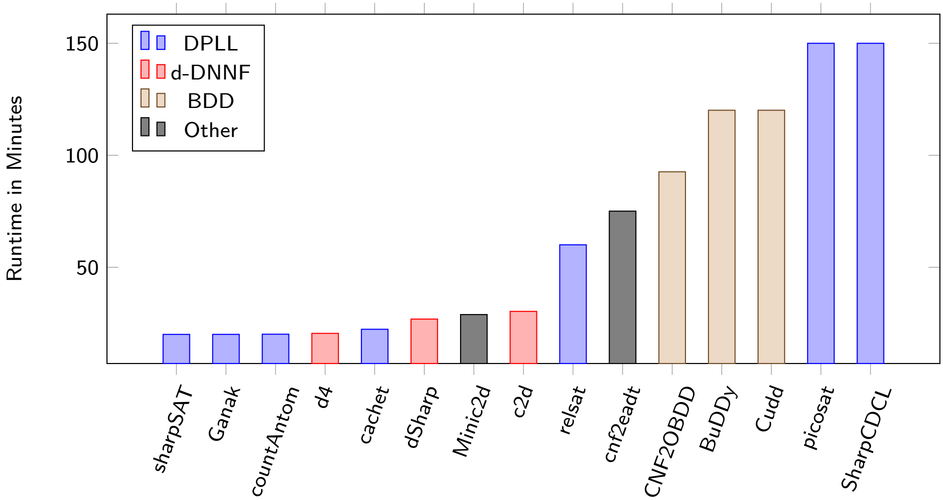
Limits

2 of 15 systems not evaluated within a week

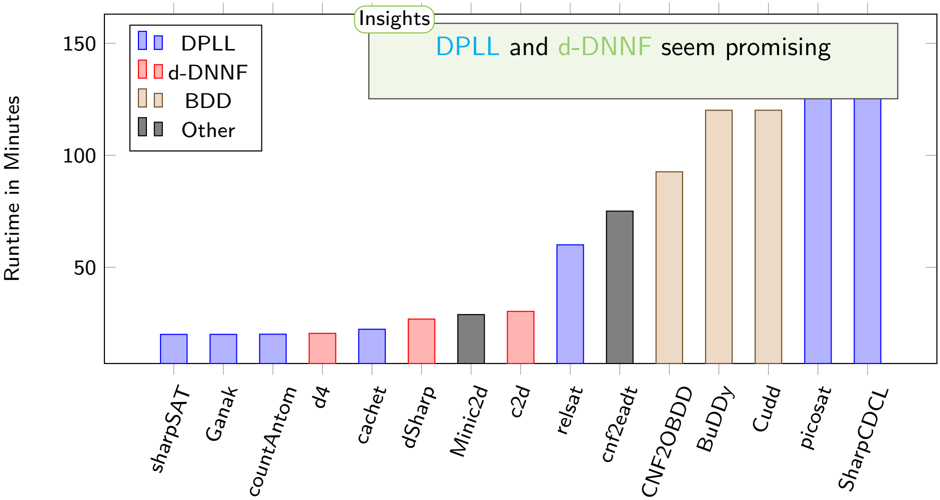
Future Work

How to scale to hard systems?

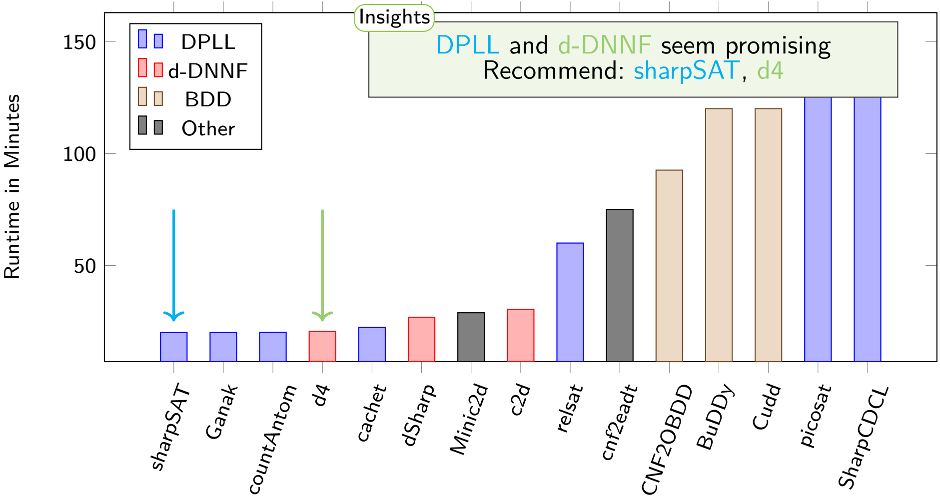
Performance Solvers



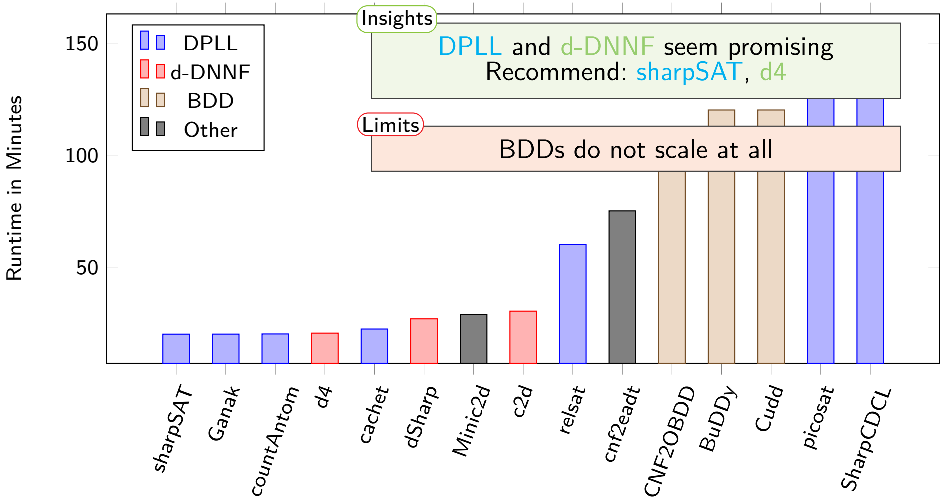
Performance Solvers



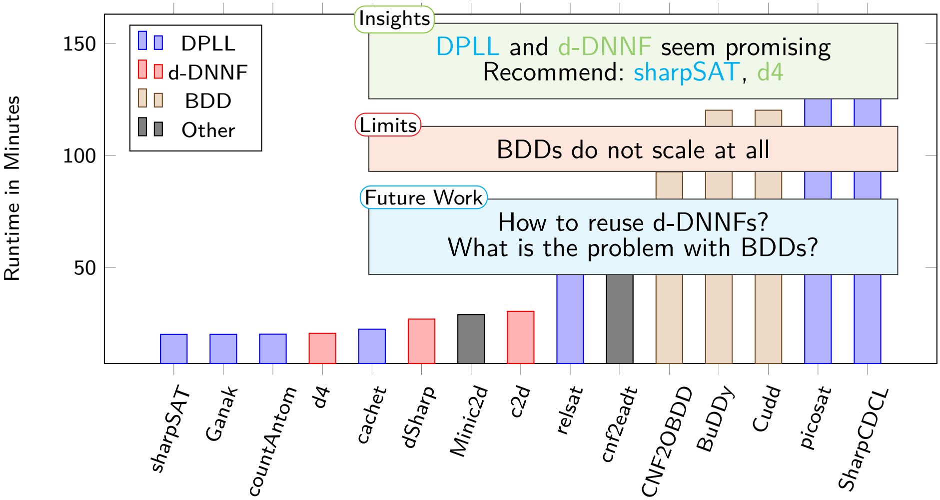
Performance Solvers



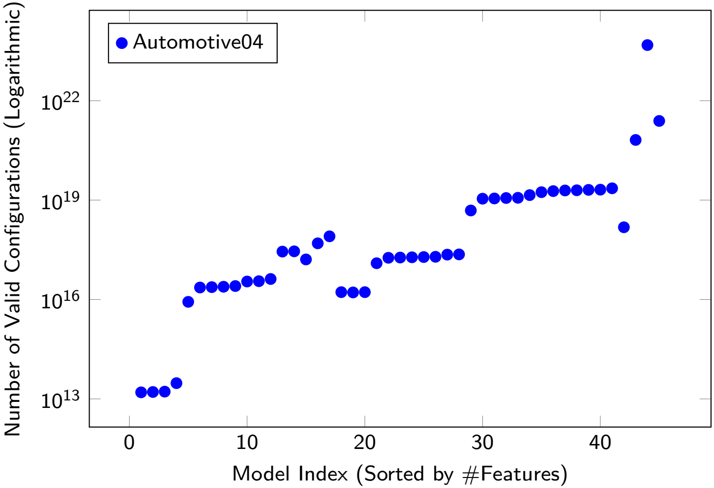
Performance Solvers



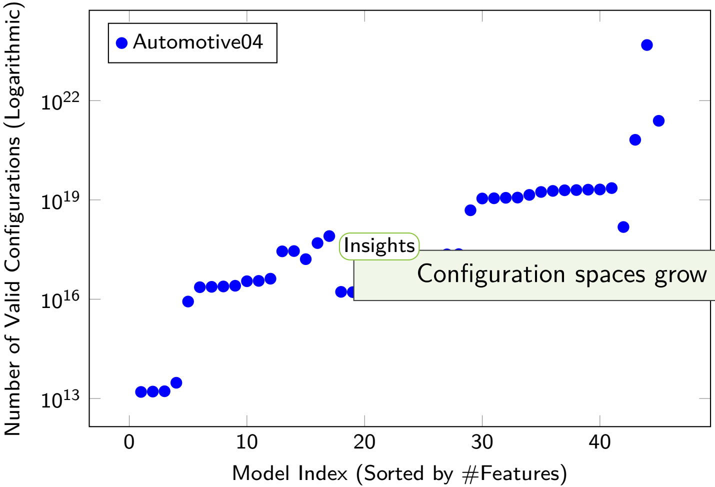
Performance Solvers



Evolution: Sizes



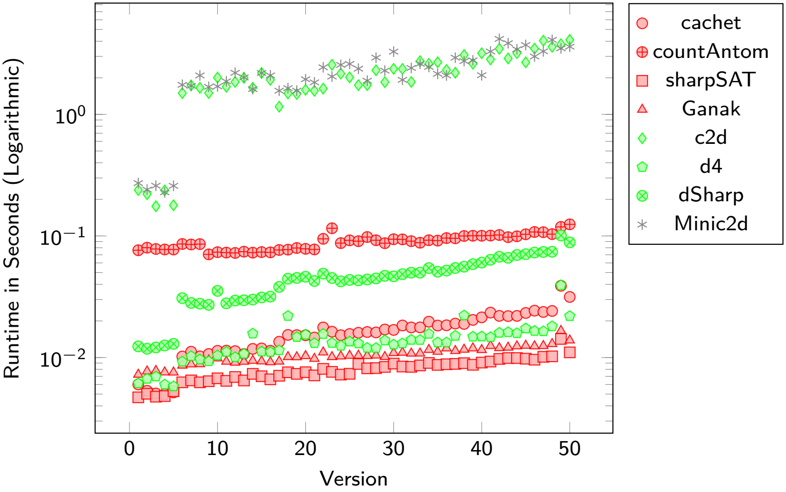
Evolution: Sizes



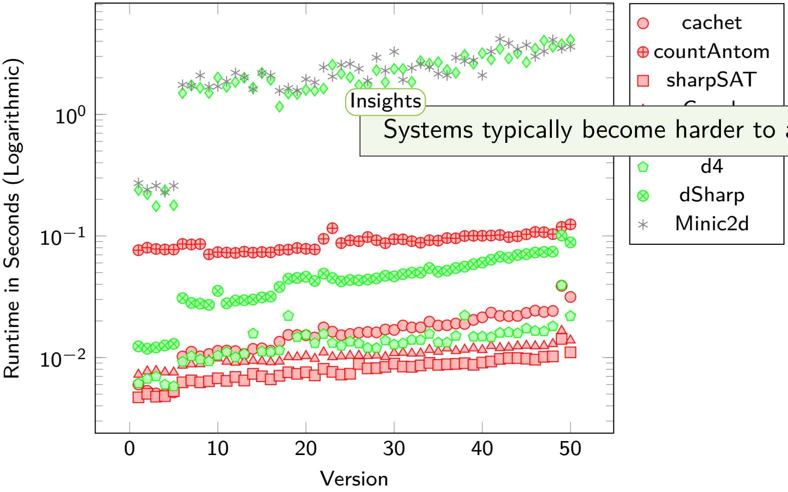
Insights

Configuration spaces grow over time

Evolution: Performance



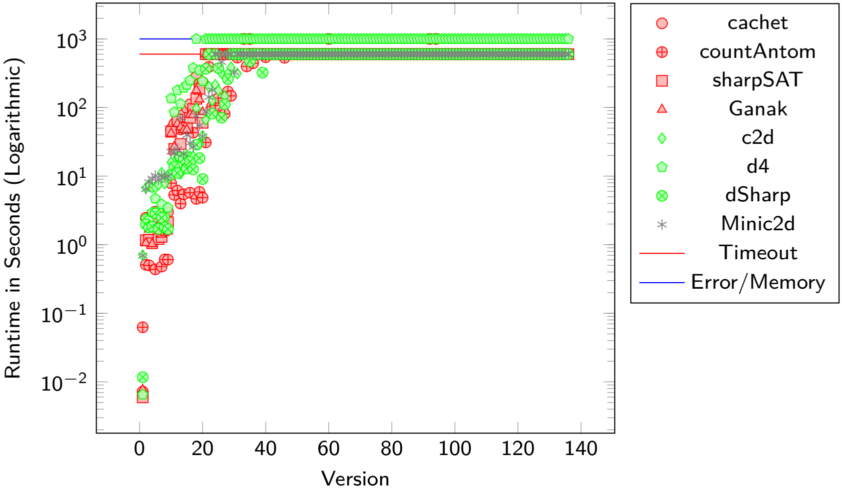
Evolution: Performance



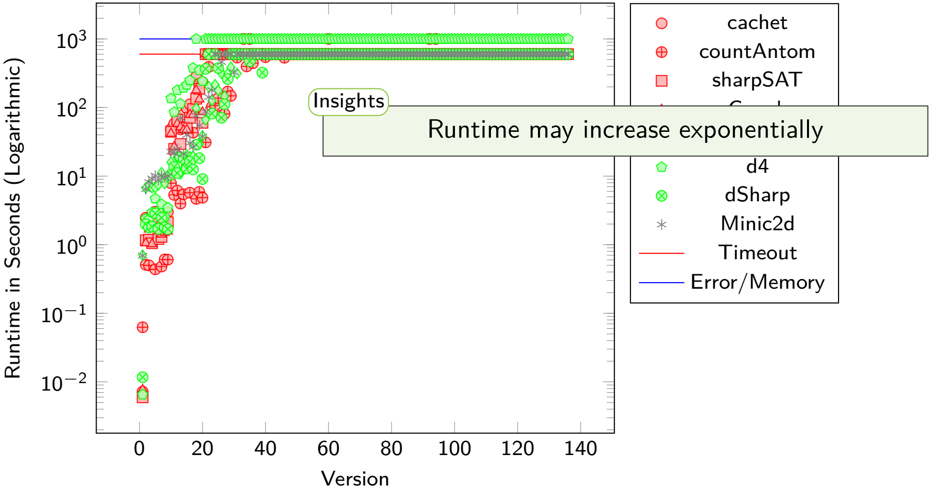
Insights

Systems typically become harder to analyze

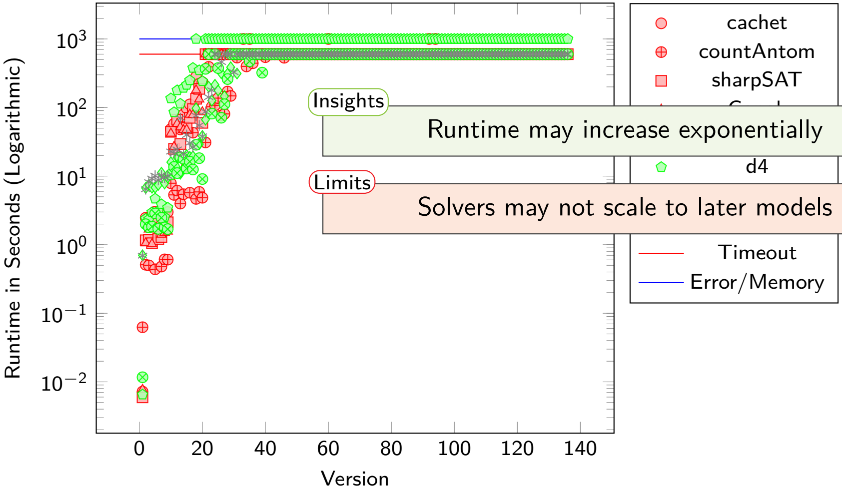
Evolution: Performance



Evolution: Performance



Evolution: Performance



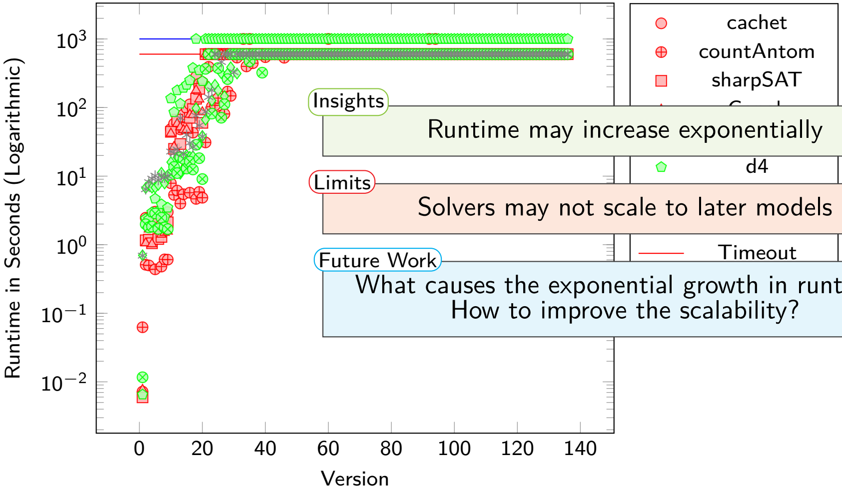
Insights

Runtime may increase exponentially

Limits

Solvers may not scale to later models

Evolution: Performance



Insights

Runtime may increase exponentially

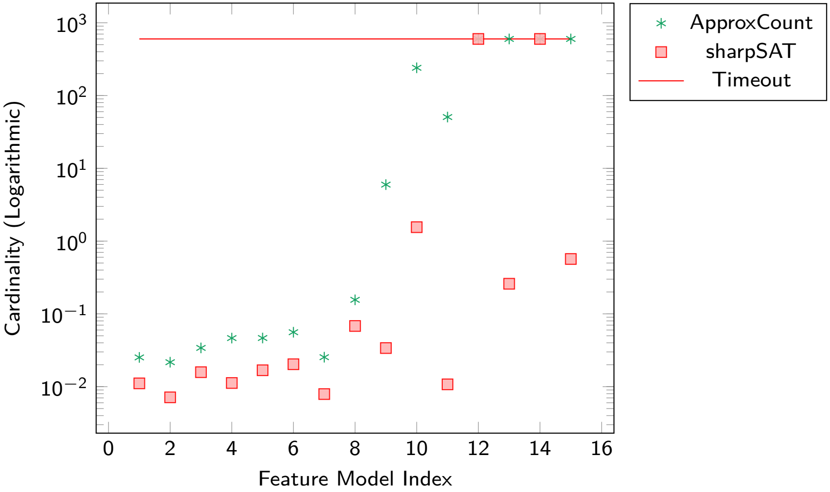
Limits

Solvers may not scale to later models

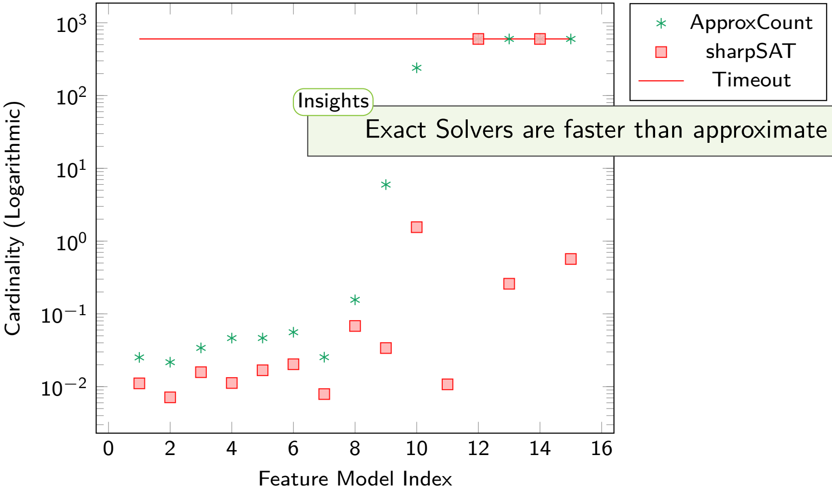
Future Work

What causes the exponential growth in runtime?
How to improve the scalability?

Approximate #SAT Solvers



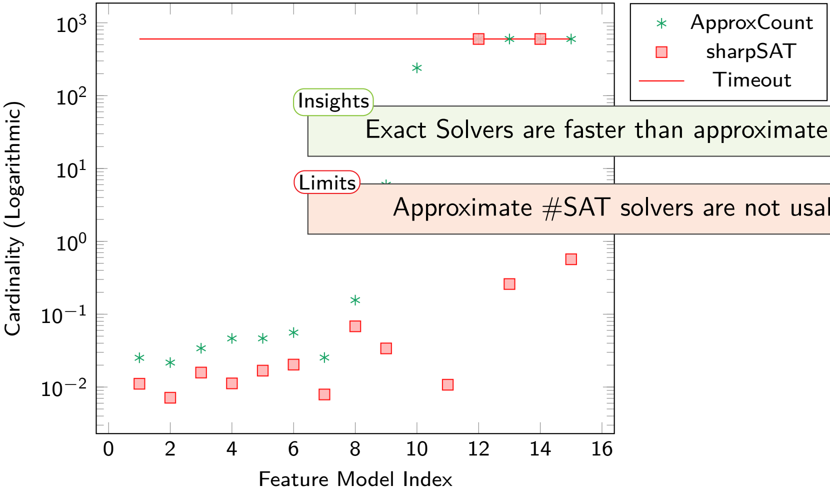
Approximate #SAT Solvers



Insights

Exact Solvers are faster than approximate ones

Approximate #SAT Solvers



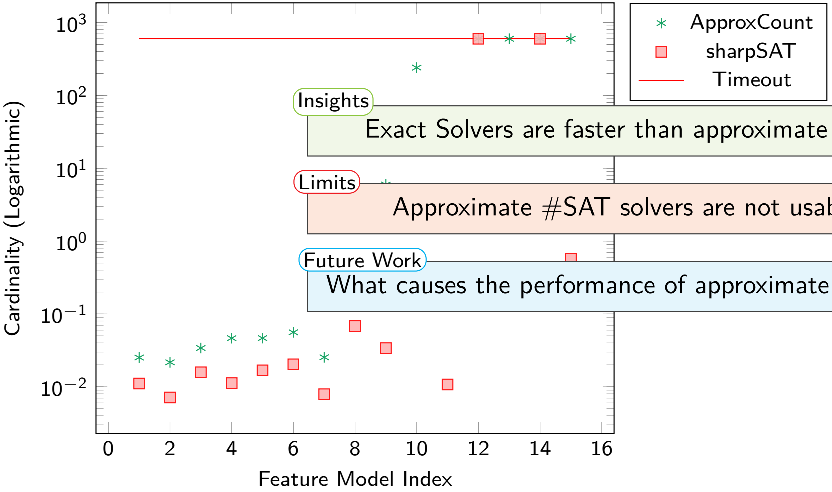
Insights

Exact Solvers are faster than approximate ones

Limits

Approximate #SAT solvers are not usable

Approximate #SAT Solvers



Insights
Exact Solvers are faster than approximate ones

Limits
Approximate #SAT solvers are not usable

Future Work
What causes the performance of approximate solvers?

Analyzing Industrial Feature Models with #SAT. *Are we there yet?*

Insights

Majority of systems can be analyzed with #SAT solvers

DPLL: ✓d-DNNF: ✓

Analyzing Industrial Feature Models with #SAT. *Are we there yet?*

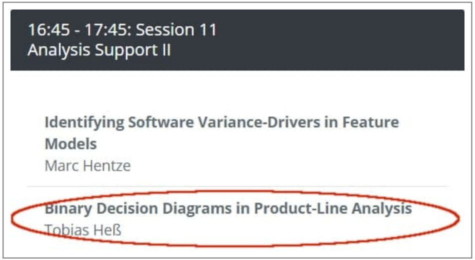
Insights

Majority of systems can be analyzed with #SAT solvers
DPLL: ✓ d-DNNF: ✓ BDD: ✗

Analyzing Industrial Feature Models with #SAT. *Are we there yet?*

Insights

Majority of systems can be analyzed with #SAT solvers
DPLL: ✓ d-DNNF: ✓ BDD: ✗



Analyzing Industrial Feature Models with #SAT. *Are we there yet?*

Insights

Majority of systems can be analyzed with #SAT solvers

DPLL: ✓d-DNNF: ✓BDD: ✗

Limits

Two systems could not be analyzed

Solvers may not scale to later models in evolution

Analyzing Industrial Feature Models with #SAT. *Are we there yet?*

Insights

Majority of systems can be analyzed with #SAT solvers
DPLL: ✓d-DNNF: ✓BDD: ✗

Limits

Two systems could not be analyzed
Solvers may not scale to later models in evolution

Future Work

How to scale to hard systems?
What causes exponential growth in runtime during evolution?