High T-Wise Coverage From Uniform Sampling

FOSD'24 | Tobias Heß, Tim Jannik Schmidt, Lukas Ostheimer, Sebastian Krieter, Thomas Thüm | 9.-12.04.2024

SP Software Engineering Programming Languages

SoftVarE

UNIVERSITÄT PADERBORN

universität ulm

UnWise: High T-Wise Coverage from Uniform Sampling

Tobias Heß
University of Ulm
Germany

Tim Jannik Schmidt
University of Ulm
Germany

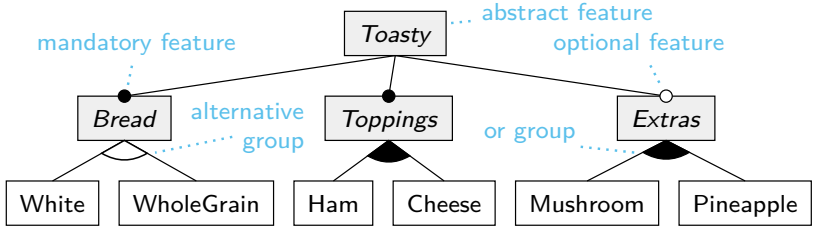
Lukas Ostheimer
University of Ulm
Germany

Sebastian Krieter
University of Ulm
Germany

Thomas Thüm
University of Ulm
Germany

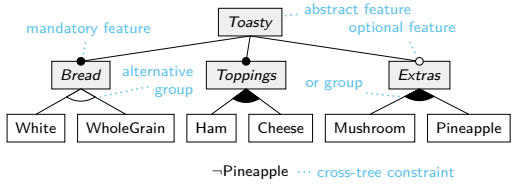
VaMoS'24

Feature Models



Valid Configuration: $c_1 = \{\text{White}, \text{Cheese}, \text{Ham}, \text{Mushroom}\}$

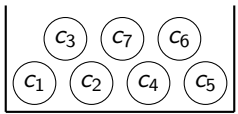
Sampling Methods



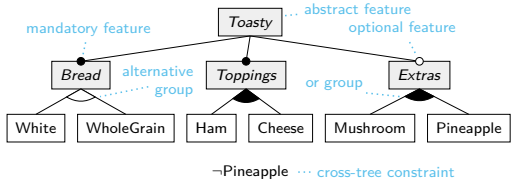
Valid Configuration: $c_1 = \{\text{White}, \text{Cheese}, \text{Ham}, \text{Mushroom}\}$

Uniform Sampling

Idea: Draw configurations at random from a urn of all configurations.



Sampling Methods



Valid Configuration: $c_1 = \{\text{White}, \text{Cheese}, \text{Ham}, \text{Mushroom}\}$

T-Wise Sampling

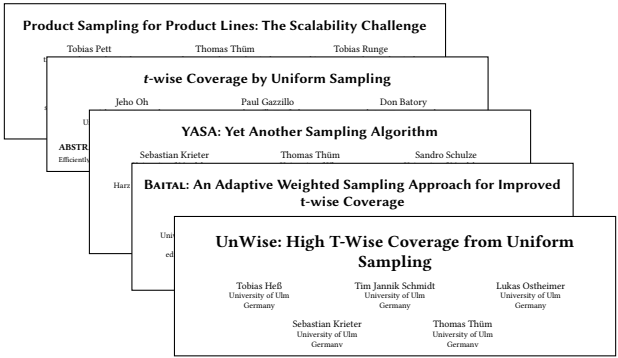
Idea: Cover all valid combinations of interactions between t features.

Pair-Wise interactions for Ham & Cheese: $H \wedge C$ $H \wedge \neg C$ $\neg H \wedge C$ $\neg H \wedge \neg C$

Sample: $\{c_1, c_2, c_3\}$ **Pair-Wise Coverage:** 100%

Find all errors caused by interactions of size 2?

The Story so far



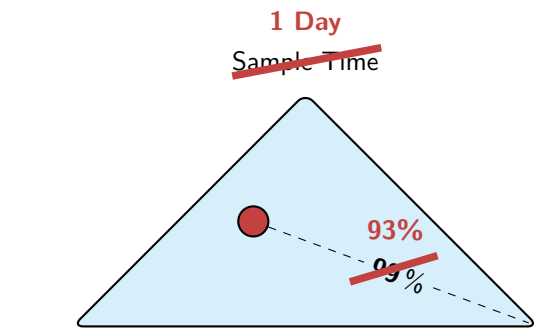
Evaluate papers using 49 real-world feature models + Post Processing

Results

Sampler	No post-processing	
	Size	Cov
Yasa (t = 2)	**272**±744.7	**99.98**±0.1 %
Baital (t = 2)	484±55.9	99.65±1.5 %
Quicksampler (s = 1024)	781±500.4	51.58±16.3 %
Spur (s = 1024)	982±145.2	88.91±11.7 %

- **YASA** as clear winner
  - even without post processing
  - Smaller Samples
  - Larger Coverage
- Uniform Sampling: Better than expected

Trade-Off & Problem in Practice



Test them all, is it worth it? Assessing configuration sampling on the JHipster Web development stack

Axel Halin[1] · Alexandre Nuttinck[2] · Mathieu Acher[3] · Xavier Devroey[4] · Gilles Perrouin[5] · Benoit Baudry[6]

A Comparison of 10 Sampling Algorithms for Configurable Systems

Flávio Medeiros, Fed. Univ. of Campina Grande, Paraíba, Brazil
Christian Kästner, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA
Márcio Ribeiro, Federal University of Alagoas, Maceió, Alagoas, Brazil
Rohit Gheyi, Fed. Univ. of Campina Grande, Paraíba, Brazil
Sven Apel, Universität Passau, Passau, Germany

⇒100% 4/5/6-wise needed

Open Questions

Current State

- What is the baseline?
- We evaluated: 93% 2-wise with restrictions → Enough?

The end of Uniform Sampling in Fault Detection?
Is T-Wise the way to go?

Planned Thesis

Configuration: {White, ¬Ham, Cheese, ¬Mushroom, ...}

```
#ifdef Ham
int f(int x){...}
#endif
#ifdef Cheese
int g() {
        return f(42);
}
#endif
```

Compile Error

```
void h(int x){
        cout << (2/x) << endl;
}
int main() {
        int x = 1;
        #ifdef Mushroom
        x = x + 1;
        #endif
        #ifdef Cheese
        x = x - 1;
        #endif
        h(x);
}
```

Runtime Error

Error Masking in SPLs

Planned Thesis

Error Masking in Software Product Lines

Significance

- Error Masking affects **all** Sampling techniques
- Problem might be **bigger** than we think
- Looks at the **Solution Space**
- Apply to **real world** fault detection

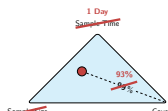
What do YOU think?

Results

Sampler	No post-processing	
	Size	Cov
Yasa ($t = 2$)	**272**±744.7	**99.98**±0.1 %
Baital ($t = 2$)	484±55.9	90.65±1.5 %
Quicksampler ($t = 1024$)	761±508.4	57.59±16.3 %
Spur ($t = 1024$)	982±145.2	88.91±11.7 %

- **YASA** as clear winner
  - even without post processing
  - Smaller Samples
  - Larger Coverage
- Uniform Sampling: Better than expected

Trade-Off & Problem in Practice



Test them all, is it worth it? Assessing configuration sampling on the JHipster Web development stack.

A Comparison of 10 Sampling Algorithms for Configurable Systems

Open Questions

Current State

- What is the baseline?
- We evaluated: 93% 2-wise with restrictions → Enough?

The end of Uniform Sampling in Fault Detection?
Is T-Wise the way to go?

Planned Thesis

Error Masking in Software Product Lines

Significance

- Error Masking affects **all** Sampling techniques
- Problem might be **bigger** than we think
- Looks at the **Solution Space**
- Apply to **real world** fault detection