Thesis Topics – Feature Modeling and Sampling

Software Engineering
Programming Languages

universität ulm

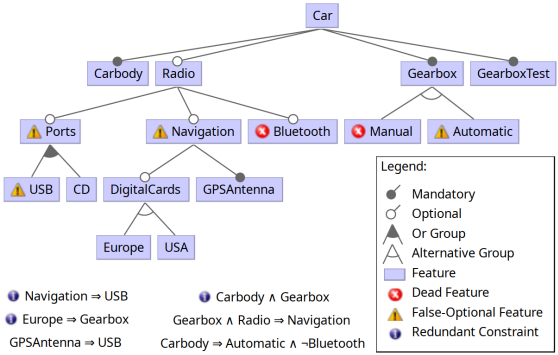
About Me

- **Short CV**
  - **2015** Master in Computer Science (Magdeburg)
  - **2022** Defended PhD Thesis (Magdeburg)
  - **Since 2022** Researcher at Uni Ulm
- **Research interests**
  - Feature modeling and analysis
  - Configuration sampling and testing
- **Email**
  - sebastian.krieter@uni-ulm.de

Cleaning Feature Models (B/P)

- Feature models may contains anomalies (e.g., dead features, false-optional features, redundant constraints...)
- Detecting them can be automatized
- Fixing them currently requires user decisions and manual effort
- **To which degree can this be automatized?**

- **Goals:**
  1. Compare and discuss suitable strategies for fixing (e.g., which redundant constraints to remove)
  2. Implement promising strategies in FeatureIDE



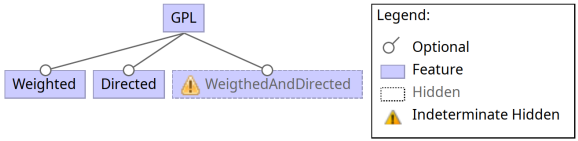
Efficient Analyses for Indeterminate Hidden Features (M)

- Hidden features cannot be configured directly

⇒ Whether a hidden feature is selected must **always** be determined by other features

- Otherwise it is **indeterminate**
- Current analyses require much computational effort
- **Are there more efficient analyses for detecting indeterminate hidden features?**

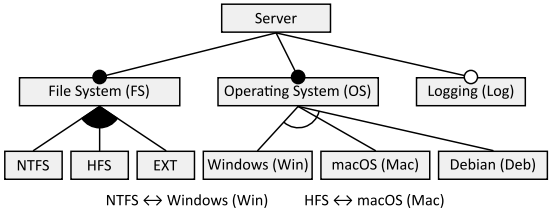
- **Goals:**
  1. Improve the current analysis for finding indeterminate hidden features in FeatureIDE
     a. Using a SAT-Solver
     b. Using a d-DNNF
  2. Evaluate the new analysis



Weighted ∧ Directed ⇒ WeigthedAndDirected

Comparing Different Coverage Criteria (M)

Background: Sampling in the Problem Space



$\Rightarrow$

$\{Server, FS, OS, HFS, Mac\}$
$\{Server, FS, OS, NTFS, EXT, Win\}$
$\{Server, FS, OS, EXT, Deb, Log\}$
...

- Create a representative list of configurations (e.g., for testing)
  - Random
  - Coverage criteria
  - ...

Comparing Different Coverage Criteria (M)

- T-Wise coverage of samples can be measured
- In literature the definition of this metrics is ambiguous or implicitly assumed

$\Rightarrow$ There many alternative variants

  - Include core/dead features? Include abstract features? Merge atomic sets? Count invalid interactions?

- **Which concrete metric should be used in which use case?**

- **Goals:**
  1. Research which variants of the metric are used in literature
  2. Evaluate the impact of using different variants on the same samples

Sampling with Evolving Presence Conditions (M)

Background: Sampling in the Solution Space

Presence Conditions	Source File
Log	`//#if Logging`
Log	`public class Logger {`
Log	`  public void log(String message) {`
$(Log \wedge EXT)$	`    //#if EXT`
$(Log \wedge EXT)$	`    print("FS is EXT");`
$(Log \wedge EXT)$	`    //#endif`
$(Log \wedge Mac) \vee (Log \wedge Win)$	`    //#if macOS || Windows`
$(Log \wedge Mac) \vee (Log \wedge Win)$	`    print("OS is not Linux");`
$(Log \wedge Mac) \vee (Log \wedge Win)$	`    //#endif`
Log	`  }`
Log	`}`
Log	`//#endif`

$\Rightarrow$ Use presence conditions instead of features for sampling

Sampling with Evolving Presence Conditions (M)

- Samples can be based on presence conditions of implementation artifacts
- Presence conditions may also change when a product line evolves
- **How often and to what degree do presence conditions change on average?**

- **Goals:**
  1. Measure the rate at which presence conditions change during the history of multiple product lines
  2. Compare the results and try to find correlations and trends

FeatureIDE und FeatJAR (P / HiWi)

- We integrate most of our research in our SPL tools
  - **FeatureIDE** is an Eclipse-based IDE for SPL development
  - **FeatJAR** is a library for SPL analysis
- These tools are frequently extended, maintained, and improved
- **Want to contribute?**