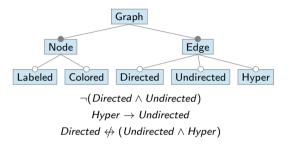


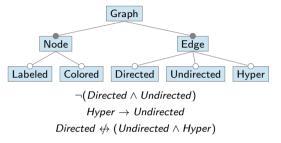
# How Easy is SAT-Based Analysis of a Feature Model?





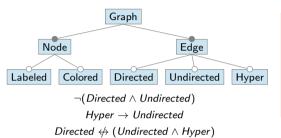


```
node.cpp
                                edge.cpp
class Node {
                                class Edge {
 #ifdef LABELED
                                  #ifdef DIRECTED
   std::string label:
                                    Node from, to:
 #endif
                                  #elif UNDIRECTED &&
 #ifdef COLORED
                                         HYPER
   std::string color:
                                    std::set < Node > nodes:
                                  #endif
  #endif
```



```
Product-Line Analyses . . . [ref]
```

```
node.cpp
                                 edge.cpp
class Node {
                                 class Edge {
 #ifdef LABELED
                                   #ifdef DIRECTED
   std::string label:
                                    Node from, to:
 #endif
                                   #elif UNDIRECTED &&
 #ifdef COLORED
                                         HYPER
   std::string color:
                                    std::set < Node > nodes:
                                  #endif
  #endif
```



```
Product-Line Analyses . . . [ref]

Which product(s) . . .

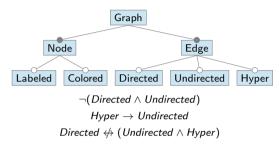
• . . . has the most lines of code? [ref]

• . . . is the fastest or smallest? [ref, ref]

• . . . have type or logic errors? [ref, ref]

• . . . have unsafe data flows? [ref]
```

```
node.cpp
                                edge.cpp
                                class Edge {
class Node {
 #ifdef LABELED
                                  #ifdef DIRECTED
   std::string label:
                                    Node from, to:
                                  #elif UNDIRECTED &&
 #endif
 #ifdef COLORED
                                         HYPER
                                    std::set < Node > nodes:
   std::string color:
  #endif
                                  #endif
```



```
node.cpp
                                edge.cpp
class Node {
                                class Edge {
 #ifdef LABELED
                                  #ifdef DIRECTED
   std::string label:
                                    Node from to:
                                  #elif UNDIRECTED &&
 #endif
 #ifdef COLORED
                                         HYPER
   std::string color:
                                    std::set < Node > nodes:
  #endif
                                  #endif
```

```
Product-Line Analyses . . . [ref]

Which product(s) . . .

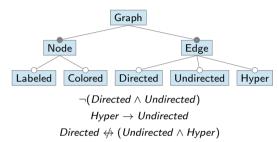
• . . . has the most lines of code? [ref]

• . . . is the fastest or smallest? [ref, ref]

• . . . have type or logic errors? [ref, ref]

• . . . have unsafe data flows? [ref]
```

```
\dots Often Rely on Feature-Model Analyses \dots
```



```
node.cpp
                                edge.cpp
class Node {
                                class Edge {
 #ifdef LABELED
                                  #ifdef DIRECTED
   std::string label:
                                    Node from to:
                                  #elif UNDIRECTED &&
 #endif
 #ifdef COLORED
                                         HYDER
   std::string color:
                                    std::set < Node > nodes:
  #endif
                                  #endif
```

```
Product-Line Analyses . . . [ref]

Which product(s) . . .

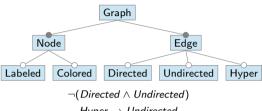
• . . . has the most lines of code? [ref]

• . . . is the fastest or smallest? [ref, ref]

• . . . have type or logic errors? [ref, ref]

• . . . have unsafe data flows? [ref]
```

```
    Often Rely on Feature-Model Analyses ...
    Which features are core/dead? [ref]
    A sample covering all t-wise interactions? [ref]
    How large is the configuration space? [ref]
    How has the configuration space evolved? [ref]
```



 $Hyper \rightarrow Undirected$   $Directed \Leftrightarrow (Undirected \land Hyper)$ 

```
node.cpp
                                edge.cpp
class Node {
                                class Edge {
 #ifdef LABELED
                                  #ifdef DIRECTED
   std::string label:
                                    Node from to:
                                  #elif UNDIRECTED &&
 #endif
 #ifdef COLORED
                                         HYDER
   std::string color:
                                    std::set < Node > nodes:
  #endif
                                  #endif
```

```
Product-Line Analyses . . . [ref]

Which product(s) . . .

• . . . has the most lines of code? [ref]

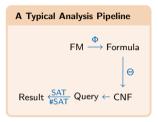
• . . . is the fastest or smallest? [ref, ref]

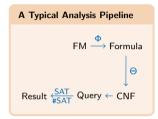
• . . . have type or logic errors? [ref, ref]

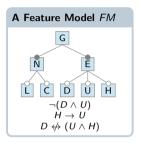
• . . . have unsafe data flows? [ref]
```

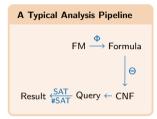
```
    Often Rely on Feature-Model Analyses ...
    Which features are core/dead? [ref]
    A sample covering all t-wise interactions? [ref]
    How large is the configuration space? [ref]
    How has the configuration space evolved? [ref]
```

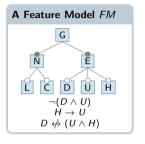
... which often rely on SAT solving (et al.)!

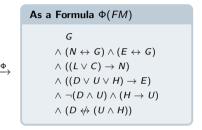


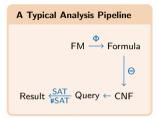


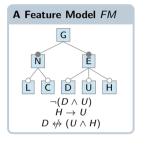


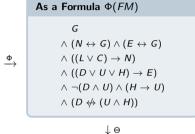


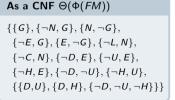


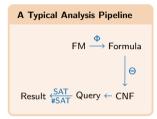


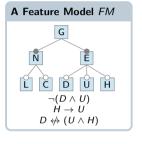


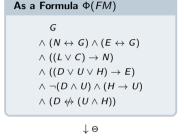










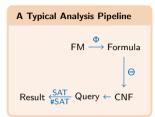


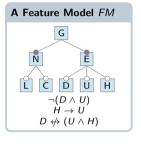


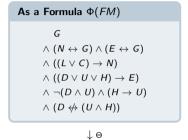
Elias Kuiter et al.

Core Feature 
$$F$$
?
$$SAT(\Theta(\Phi(FM)) \land \neg F)$$

As a CNF  $\Theta(\Phi(FM))$  $\{\{G\}, \{\neg N, G\}, \{N, \neg G\}, \{\neg E, G\}, \{E, \neg G\}, \{\neg L, N\}, \{\neg C, N\}, \{\neg D, E\}, \{\neg U, E\}, \{\neg H, E\}, \{\neg D, \neg U\}, \{\neg H, U\}, \{\{D, U\}, \{D, H\}, \{\neg D, \neg U, \neg H\}\}\}$ 







Core Features 
$$\{G, N, E\}$$

8

$$\leftarrow \qquad \begin{array}{c} \textbf{Products in } FM? \\ \#SAT(\Theta(\Phi(FM))) \end{array}$$

Core Feature F?

 $SAT(\Theta(\Phi(FM)) \wedge \neg F)$ 

As a CNF  $\Theta(\Phi(FM))$  $\{\{G\}, \{\neg N, G\}, \{N, \neg G\}, \{\neg E, G\}, \{E, \neg G\}, \{\neg L, N\}, \{\neg C, N\}, \{\neg D, E\}, \{\neg U, E\}, \{\neg H, E\}, \{\neg D, \neg U\}, \{\neg H, U\}, \{\{D, U\}, \{D, H\}, \{\neg D, \neg U, \neg H\}\}\}$ 

#### SAT-based Analysis of Feature Models is Easy

Marcilio Mendonca University of Waterloo, Canada marcilio@csg.uwaterloo.ca Andrzej Wąsowski IT University of Copenhagen, Denmark wasowski@itu.dk

Krzysztof Czarnecki University of Waterloo, Canada kczarnec@gsd.uwaterloo.ca

#### SAT-based Analysis of Feature Models is Easy

Marcilio Mendonca University of Waterloo, Canada marcilio@csg.uwaterloo.ca Andrzej Wąsowski IT University of Copenhagen, Denmark wasowski@itu.dk

Krzysztof Czarnecki
University of Waterloo, Canada
kczarnec@gsd.uwaterloo.ca

### SAT-based Analysis of Large Real-world Feature Models is Easy

Jia Hui Liang University of Waterloo, Canada

Krzysztof Czarnecki University of Waterloo, Canada Vijay Ganesh University of Waterloo, Canada

Venkatesh Raman Institute of Mathematical Sciences, India

#### SAT-based Analysis of Feature Models is Easy

Marcilio Mendonca University of Waterloo, Canada marcilio@csg.uwaterloo.ca Andrzej Wąsowski IT University of Copenhagen, Denmark wasowski@itu.dk

Krzysztof Czarnecki University of Waterloo, Canada kczarnec@gsd.uwaterloo.ca

### SAT-based Analysis of Large Real-world Feature Models is Easy

Jia Hui Liang University of Waterloo, Canada

Krzysztof Czarnecki University of Waterloo, Canada Vijay Ganesh University of Waterloo, Canada

Venkatesh Raman Institute of Mathematical Sciences, India

### "easy"

as in

"performs much better than expected despite being NP-complete"

(because no phase transition is observed on typical feature models)

#### SAT-based Analysis of Feature Models is Easy

Marcilio Mendonca University of Waterloo, Canada marcilio@csg.uwaterloo.ca Andrzej Wąsowski IT University of Copenhagen, Denmark wasowski@itu.dk

Krzysztof Czarnecki University of Waterloo, Canada kczarnec@gsd.uwaterloo.ca

## SAT-based Analysis of Large Real-world Feature Models is Easy

Jia Hui Liang University of Waterloo, Canada

Krzysztof Czarnecki University of Waterloo, Canada Vijay Ganesh University of Waterloo, Canada

Venkatesh Raman Institute of Mathematical Sciences, India

### "easy"

as in

"performs much better than expected despite being NP-complete"

(because no phase transition is observed on typical feature models)

### Yes, But ...

- easy = fast?
  - what about pre-solving steps?
  - what about repeated solver calls?
- are non-SAT analyses also easy?
- are all feature models equally easy?

#### SAT-based Analysis of Feature Models is Easy

Marcilio Mendonca University of Waterloo, Canada marcilio@csg.uwaterloo.ca Andrzej Wąsowski IT University of Copenhagen, Denmark wasowski@itu.dk

Krzysztof Czarnecki University of Waterloo, Canada kczarnec@gsd.uwaterloo.ca

### SAT-based Analysis of Large Real-world Feature Models is Easy

Jia Hui Liang University of Waterloo, Canada

Krzysztof Czarnecki University of Waterloo, Canada Vijay Ganesh University of Waterloo, Canada

Venkatesh Raman Institute of Mathematical Sciences, India as in

"performs much better than expected despite being NP-complete"

(because no phase transition is observed on typical feature models)

### Yes, But ...

- $\blacksquare$  easy = fast?
  - what about pre-solving steps?
  - what about repeated solver calls?
- are non-SAT analyses also easy?
- are all feature models equally easy?

<sup>&</sup>quot;easy"

<sup>\*</sup>on most instances, for most purposes

Contact Association (Personal Association (P 

(II. 0.00.) CONTROL (CONTROL CONTROL C 

MENDEN, DONAMETICIES, DAM MARTETO (COMPLISOS, DETER EMICIANISMO DE DESENTICIES, DAM MARTETO (CONTRA DE PROPRIO TECNO DE LA MESTE COMMETTE CARSONALISTO (CONTRA DE LA MESTE DAM TEST, DA MARTETO (CONTRA DE PROPRIO DE MARTETO TECNO DE LA MESTE COMPLETE CARSONALISTO (CONTRA DE LA MESTE DA MARTETO (CONTRA DE LA MESTE DA MARTETO (CONTRA DE

Office and individual file (Control of the Control of the Control

(IX)()(der1903). JCIANT (13)(MATSIA) ((der0913). ELONGTI ((der0913). ELONGTI ((der0913). ELONGTI ((der0913). ELONGTI ((der0913). ELO

THE PART OF THE PA BECCES (Control and Association Association (Control and Association Association) (Control and Assoc

CONTRIBUTION (DEPT AGE (LDC)) [ (AF CONTRIBUTION) AND FERT AGE (CONTRIBUTION) AND FERT [ (AF CONTRIBUTION) AGE (CONTRIBUTION) A 

Company (1971) (Company (1971)

### On the Advanced service of the Control of the Contr Linux kernel

CONTROL SECTION DE L'ANDIE DE L'A AN ART OF A MARKED CONTROL OF THE ART OF A CONTROL OF THE ART OF A CONTROL OF THE ART OF A CONTROL OF THE ART 

- number of configurations unknown past 2007
- family-based type checking infeasible

| Control | Cont

- as is uniform random sampling, slicing, diffing, . . .
- even core/dead features are challenging
- same for Freetz-NG. Buildroot. Automotive02. . . .

SOURCE CLASSIANS TASSIANS

NCSAMA(TIRSAMA DEXIVADAMA(TIRSAMA LISAMA(TIRSAMA

III (derfc8P80511)Ade MAR, E2C)AderfcE1804 III (derfc58E588)Ader ISO))(cdef

11) HAPCHE, SECIA

OCCUPANT DESCRIP

(OR AND RELEASED LIGHTYNES OF THE CONTROL OF THE CO

Office and individual file (Control of the Control of the Control

(IX)()(der1903). JCIANT (13)(MATSIA) ((der0913). ELONGTI ((der0913). ELONGTI ((der0913). ELONGTI ((der0913). ELONGTI ((der0913). ELO

A property of the control of the con

CONTRIBUTION (DEPT AGE (LDC)) [ (AF CONTRIBUTION) AND FERT AGE (CONTRIBUTION) AND FERT [ (AF CONTRIBUTION) AGE (CONTRIBUTION) A Committee (12.1) [[100] Charles [10.1] Charles [10.

ACCOUNT OF MALES AND ACCOUNT O

OFFICE AND SYMMETRY CONSTRUCTION OF STREET AND ADMINISTRATION OF STREET ADMINISTRATION OF STREET AND ADMINISTRATION OF STREET ADMINISTRATION OF

CITE\_00029\_DCubed(100see(100))(renotion\_100see(100s

### Linux kernel

CONTROL SECTION DE L'ANDIE DE L'A AN ART OF A MARKED CONTROL OF THE ART OF A CONTROL OF THE ART OF A CONTROL OF THE ART OF A CONTROL OF THE ART TOTAL SECTION AND ADDRESS OF THE PROPERTY OF T 

- number of configurations unknown past 2007
- family-based type checking infeasible
- as is uniform random sampling, slicing, diffing, . . .
- even core/dead features are challenging
- same for Freetz-NG. Buildroot. Automotive02. . . .

solvers get faster – but feature models grow more complex!

- ⇒ need better understanding of feature-model complexity
- ⇒ perspective shift to instance-based meta-analysis

SOURCE CLASSIANS TASSIANS

NCSAMA(TIRSAMA DEXIVADAMA(TIRSAMA LISAMA(TIRSAMA

III (derfc8P80511)Ade MAR, E2C)AderfcE1804 III (derfc58E588)Ader ISO))(cdef

11) HAPCHE, SECIA

OCCUPANT DESCRIP

ITE CLASSIBLE!

IN CLK VCS (BAH)

#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

### Feature-Model Meta-Analysis

the practice of **asking** and answering **questions about** feature-model analyses

⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

From Class-Based Meta-Analysis . . .

#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis . . .

[ref, ref]

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis . . .

[ref, ref]

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

- "How much time does analysis X need on feature model Y when using solver Z?"
- "Which algorithm is most memory-efficient for computing X on Y?"

#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis . . .

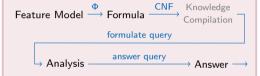
[ref, ref]

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

#### ... to Instance-Based Meta-Analysis

- "How much time does analysis X need on feature model Y when using solver Z?"
- "Which algorithm is most memory-efficient for computing X on Y?"

### Influence Factors for Feature-Model Analysis



#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

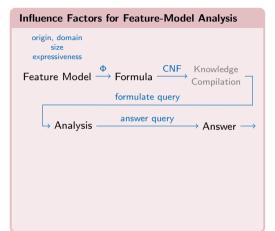
⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis . . .

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

#### ... to Instance-Based Meta-Analysis

- "How much time does analysis X need on feature model Y when using solver Z?"
- "Which algorithm is most memory-efficient for computing X on Y?"



[ref. ref]

#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis ...

[ref, ref]

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

#### ... to Instance-Based Meta-Analysis

- "How much time does analysis X need on feature model Y when using solver Z?"
- "Which algorithm is most memory-efficient for computing X on Y?"

# Influence Factors for Feature-Model Analysis origin, domain KConfig extractor size non-Boolean variability expressiveness preprocessing CNF Knowledge Feature Model $\xrightarrow{}$ Formula -Compilation formulate query answer query → Analysis $\rightarrow$ Answer $\longrightarrow$

#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

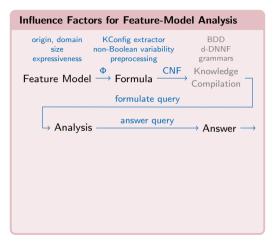
⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis . . .

[ref, ref]

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

- "How much time does analysis X need on feature model Y when using solver Z?"
- "Which algorithm is most memory-efficient for computing X on Y?"



#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

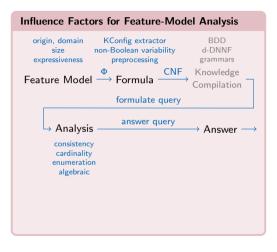
⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis . . .

[ref, ref]

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

- "How much time does analysis X need on feature model Y when using solver Z?"
- "Which algorithm is most memory-efficient for computing X on Y?"



#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

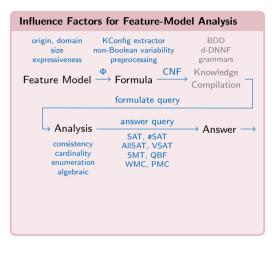
⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis ...

[ref, ref]

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

- "How much time does analysis X need on feature model Y when using solver Z?"
- "Which algorithm is most memory-efficient for computing X on Y?"



#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis . . .

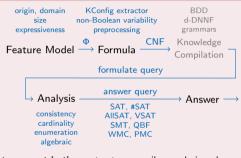
[ref, ref]

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

#### ... to Instance-Based Meta-Analysis

- "How much time does analysis X need on feature model Y when using solver Z?"
- "Which algorithm is most memory-efficient for computing X on Y?"

#### Influence Factors for Feature-Model Analysis



- + parametrization: extractor, compiler, analysis, solver
- + prior information: incremental analysis, interfaces
- + execution environment: CPU, RAM, deep variability

#### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ ask for (non-)functional requirements (e.g., correctness, runtime, memory, energy)

#### From Class-Based Meta-Analysis . . .

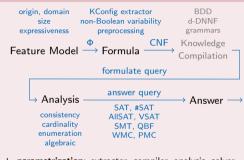
[ref, ref]

- "Is SAT-based analysis of feature models easy?"
- "Is SAT-based analysis of large real-world feature models easy?"

#### ... to Instance-Based Meta-Analysis

- "How much time does analysis X need on feature model Y when using solver Z?"
- "Which algorithm is most memory-efficient for computing X on Y?"

#### Influence Factors for Feature-Model Analysis



- + parametrization: extractor, compiler, analysis, solver
- + prior information: incremental analysis, interfaces
- + execution environment: CPU, RAM, deep variability

for practical analysis tasks, there are many analysis plans

## Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

## Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

## Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

Surrogate Metrics for Avoiding the Computation

Choosing Criteria & an Algorithm

### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

### Surrogate Metrics for Avoiding the Computation

- syntactic (e.g., number of features, variables, constraints, clauses, literals; constraint size, density)
- semantic (e.g., phase transition, community structure, self-similarity)

### Choosing Criteria & an Algorithm

- syntactic metrics only give rough estimates
- semantic metrics probably better, but NP-hard
   ⇒ try approximating

### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

### Surrogate Metrics for Avoiding the Computation

- syntactic (e.g., number of features, variables, constraints, clauses, literals; constraint size, density)
- semantic (e.g., phase transition, community structure, self-similarity)

## Choosing Criteria & an Algorithm

- binary (e.g., "yes/no" for phase transition)
- linear (e.g., number of features ∼ analysis time)
- machine learning techniques

- syntactic metrics only give rough estimates
- semantic metrics probably better, but NP-hard
   ⇒ try approximating
- ML is sensitive to small changes in the input

### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

### Surrogate Metrics for Avoiding the Computation

- syntactic (e.g., number of features, variables, constraints, clauses, literals; constraint size, density)
- semantic (e.g., phase transition, community structure, self-similarity)

#### Choosing Criteria & an Algorithm

- binary (e.g., "yes/no" for phase transition)
- linear (e.g., number of features  $\sim$  analysis time)
- machine learning techniques

- syntactic metrics only give rough estimates
- semantic metrics probably better, but NP-hard
   ⇒ try approximating
- ML is sensitive to small changes in the input

## Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

### Surrogate Metrics for Avoiding the Computation

- syntactic (e.g., number of features, variables, constraints, clauses, literals; constraint size, density)
- semantic (e.g., phase transition, community structure, self-similarity)

#### Choosing Criteria & an Algorithm

- binary (e.g., "yes/no" for phase transition)
- linear (e.g., number of features ∼ analysis time)
- machine learning techniques

- syntactic metrics only give rough estimates
- semantic metrics probably better, but NP-hard
   ⇒ try approximating
- ML is sensitive to small changes in the input

#### Discussion & Outlook

• what meta-analysis questions are worth asking?

### Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

### Surrogate Metrics for Avoiding the Computation

- syntactic (e.g., number of features, variables, constraints, clauses, literals; constraint size, density)
- semantic (e.g., phase transition, community structure, self-similarity)

#### Choosing Criteria & an Algorithm

- binary (e.g., "yes/no" for phase transition)
- linear (e.g., number of features ∼ analysis time)
- machine learning techniques

- syntactic metrics only give rough estimates
- semantic metrics probably better, but NP-hard
   ⇒ try approximating
- ML is sensitive to small changes in the input

- what meta-analysis questions are worth asking?
- which factors are relevant, how do they interact?

## Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

### Surrogate Metrics for Avoiding the Computation

- syntactic (e.g., number of features, variables, constraints, clauses, literals; constraint size, density)
- semantic (e.g., phase transition, community structure, self-similarity)

## Choosing Criteria & an Algorithm

- binary (e.g., "yes/no" for phase transition)
- linear (e.g., number of features ∼ analysis time)
- machine learning techniques

- syntactic metrics only give rough estimates
- semantic metrics probably better, but NP-hard
   ⇒ try approximating
- ML is sensitive to small changes in the input

- what meta-analysis questions are worth asking?
- which factors are relevant, how do they interact?
- is feature-model complexity intrinsic?

## Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

### Surrogate Metrics for Avoiding the Computation

- syntactic (e.g., number of features, variables, constraints, clauses, literals; constraint size, density)
- semantic (e.g., phase transition, community structure, self-similarity)

#### Choosing Criteria & an Algorithm

- binary (e.g., "yes/no" for phase transition)
- linear (e.g., number of features ∼ analysis time)
- machine learning techniques

- syntactic metrics only give rough estimates
- semantic metrics probably better, but NP-hard
   ⇒ try approximating
- ML is sensitive to small changes in the input

- what meta-analysis questions are worth asking?
- which factors are relevant, how do they interact?
- is feature-model complexity intrinsic?
- when do advanced techniques (knowledge computation, incremental analysis, ...) pay off?

## Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

### Surrogate Metrics for Avoiding the Computation

- syntactic (e.g., number of features, variables, constraints, clauses, literals; constraint size, density)
- semantic (e.g., phase transition, community structure, self-similarity)

#### Choosing Criteria & an Algorithm

- binary (e.g., "yes/no" for phase transition)
- linear (e.g., number of features ∼ analysis time)
- machine learning techniques

- syntactic metrics only give rough estimates
- semantic metrics probably better, but NP-hard
   ⇒ try approximating
- ML is sensitive to small changes in the input

- what meta-analysis questions are worth asking?
- which factors are relevant, how do they interact?
- is feature-model complexity intrinsic?
- when do advanced techniques (knowledge computation, incremental analysis, . . .) pay off?
- long-term goal: a meta-analyzer that finds the best analysis plan for a given analysis task (cf. portfolio solving, relational query optimization)

## Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

⇒ define criteria and an algorithm to answer the question (either exactly or an estimate)

### Surrogate Metrics for Avoiding the Computation

- syntactic (e.g., number of features, variables, constraints, clauses, literals; constraint size, density)
- semantic (e.g., phase transition, community structure, self-similarity)

### Choosing Criteria & an Algorithm

- binary (e.g., "yes/no" for phase transition)
- ullet linear (e.g., number of features  $\sim$  analysis time)
- machine learning techniques

- syntactic metrics only give rough estimates
- semantic metrics probably better, but NP-hard
   ⇒ try approximating
- ML is sensitive to small changes in the input

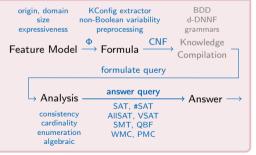
- what meta-analysis questions are worth asking?
- which factors are relevant, how do they interact?
- is feature-model complexity intrinsic?
- when do advanced techniques (knowledge computation, incremental analysis, ...) pay off?
- long-term goal: a meta-analyzer that finds the best analysis plan for a given analysis task (cf. portfolio solving, relational query optimization)
- tool support: FeatJAR (FeatureIDE 4.0), torte, clausy, KeYPI, PCLocator, Course on SPLs, . . .

## Conclusion

## Feature-Model Meta-Analysis

the practice of asking and answering questions about feature-model analyses

## Influence Factors for Feature-Model Analysis



# Your opinion?

Does feature-model complexity matter for your work? Are you doing meta-analysis?

How would you answer a meta-analysis question?





\(\sigma\)/SoftVarE-Group/Papers

\(\sigma\)/SoftVarE-Group/Slides