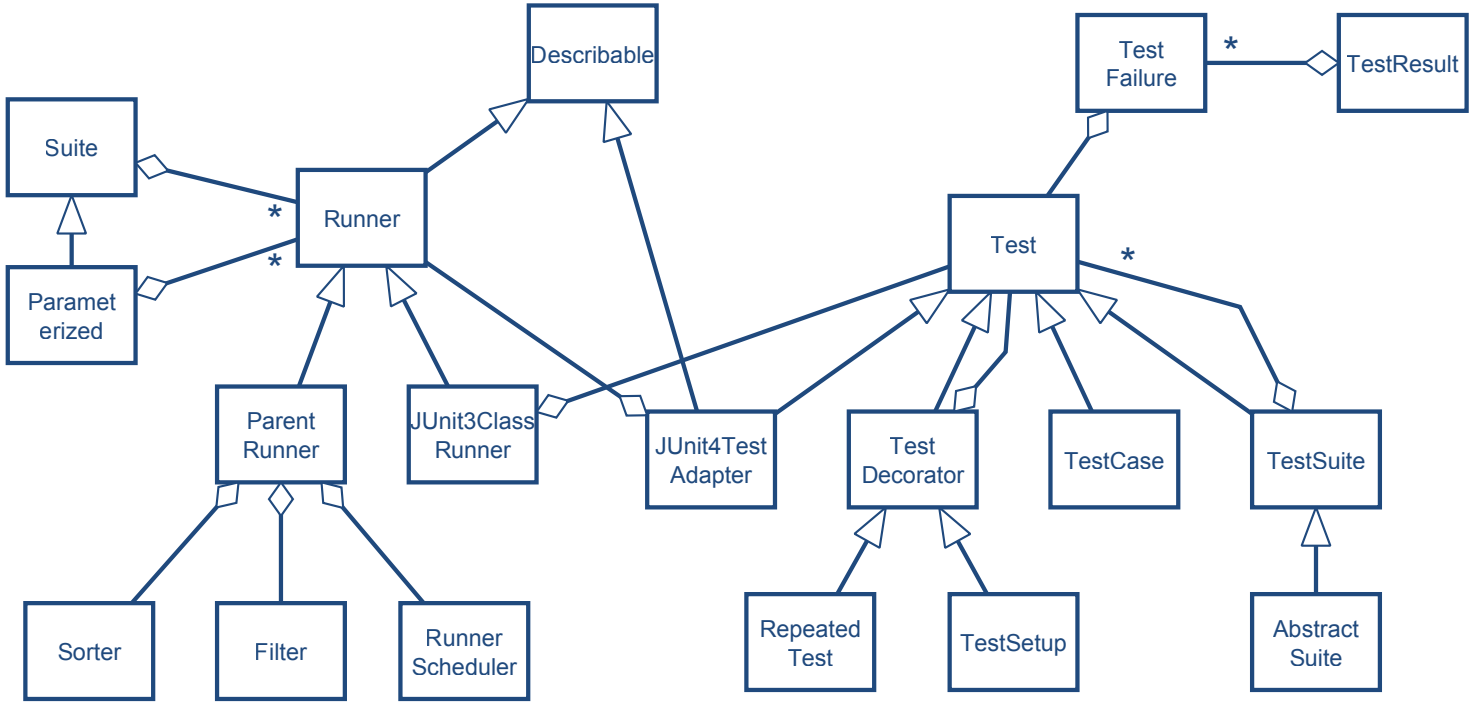
Assisted Software Exploration using Formal Concept Analysis

Paul Heckmann
Daniel Speicher

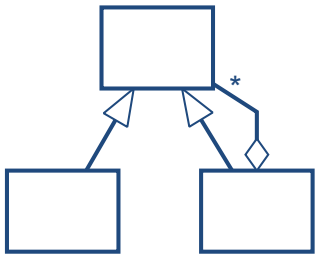
{heckmann, dsp}@cs.uni-bonn.de

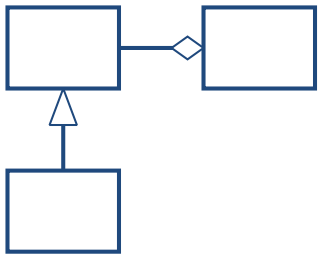
universitätbonn

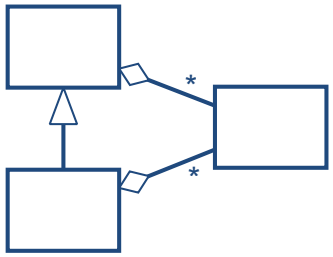
Motivation


Describable
Test Failure
*
TestResult
Suite
Runner
*
*
Test
*
Paramet erized
Parent Runner
JUnit3Class Runner
JUnit4Test Adapter
Test Decorator
TestCase
TestSuite
Sorter
Filter
Runner Scheduler
Repeated Test
TestSetup
Abstract Suite

Structure "Catalog"







1. Formal Concept Analysis

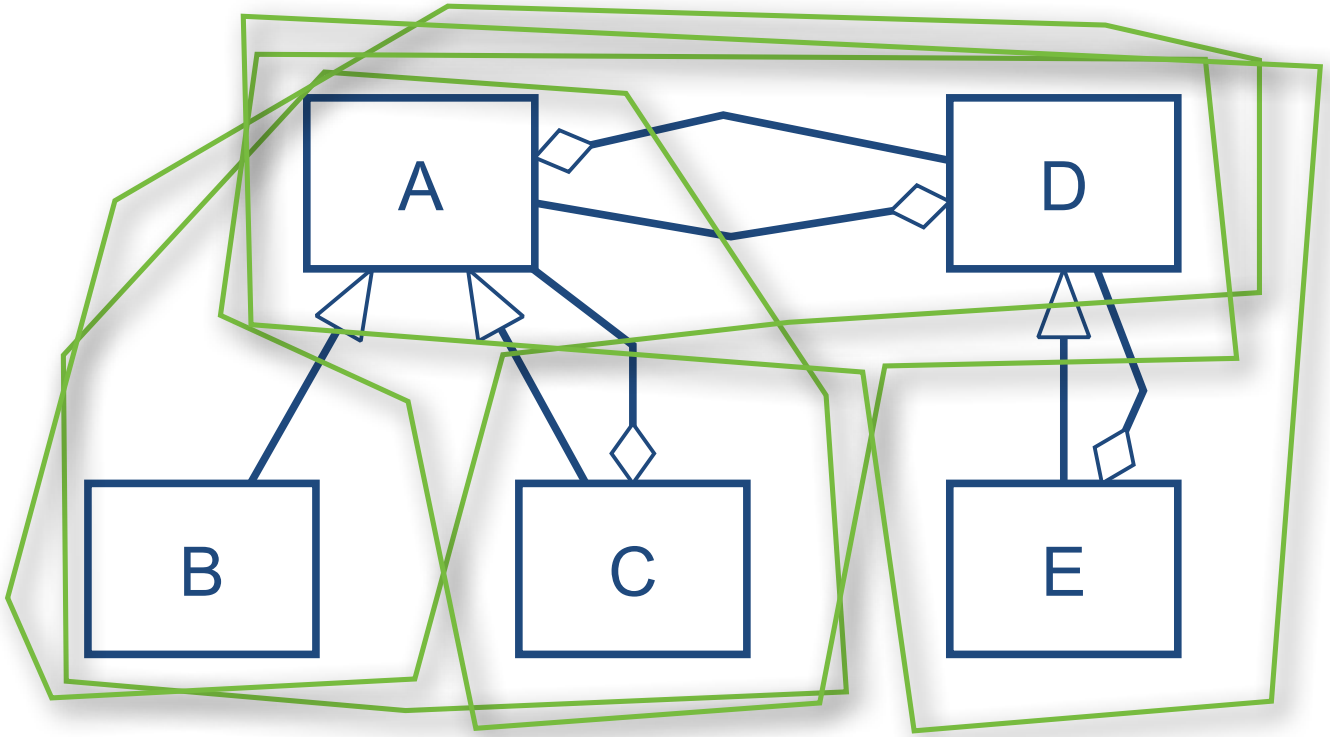
2. Exploration Support

FCA▸ Background

Formal Concept Analysis
- B. Ganter, R. Wille; 1996

Structural Software Decomposition
- P. Tonella, G. Antonial; 1998
- G. Arévalo; 2004

Formal Context



n = 3

(A, D, E)

(A, C, D)

(A, B, D)

(A, B, C)

Formal Context

n = 3

data

attributes

(A, D, E)

(A, C, D)

(A, B, D)

(A, B, C)

inheritance

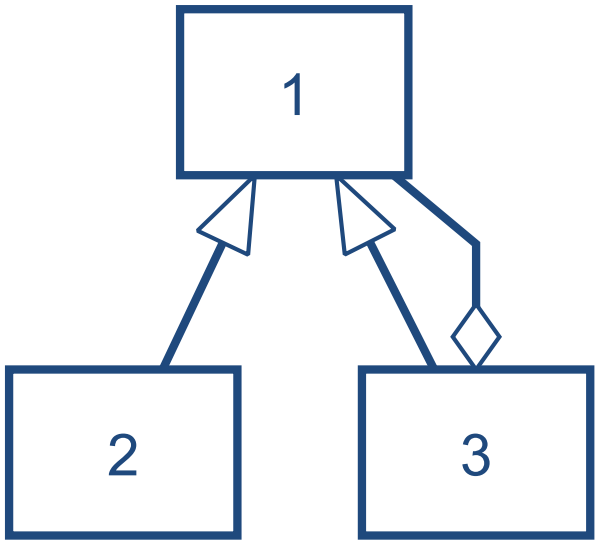
association

Formal Context

	inh(2, 1)	inh(3, 1)	assoc(3, 1)	inh(3, 2)	...
(A, B, C)	x	x	x		
(A, B, D)	x		x		
(A, C, D)	x		x	x	
(A, D, E)			x	x	

(A, B, C), inh(2, 1) ⇒ B inherits from A

Formal Concept



Intent:

{inh(2,1), inh(3,1), assoc(3,1)}

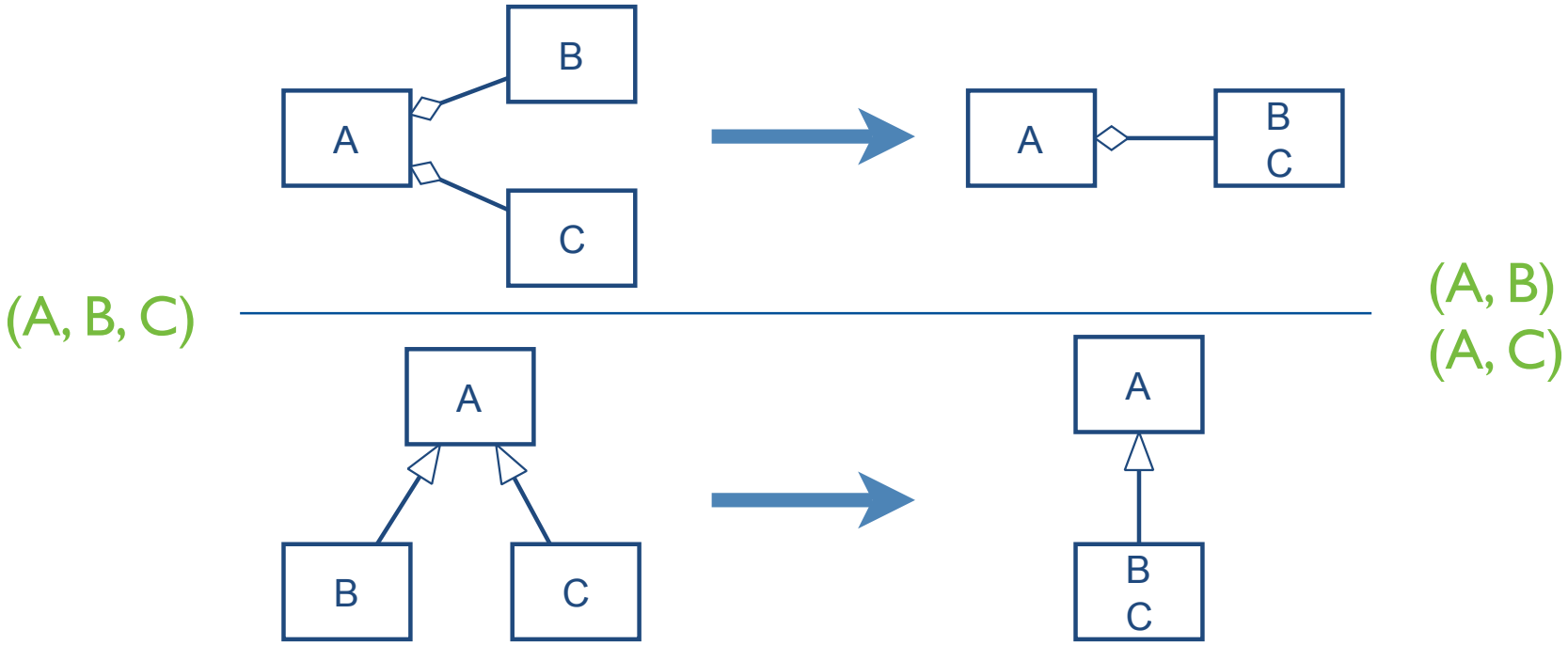
Extent:

{(A,B,C)}

+ (A, B, D)

FCA▶

Role Model Transformation



FCA▶

Runtime

	JUnit 4.7	Cultivate	JHotDraw 7
#classes	143	607	625
n = 2	<1s	2s	3s
n = 3	<1s	3s	14s
n = 4	1s	187s	130s
n = 5	13s	x	2901s

Quad @2.83GHz, 4GB RAM

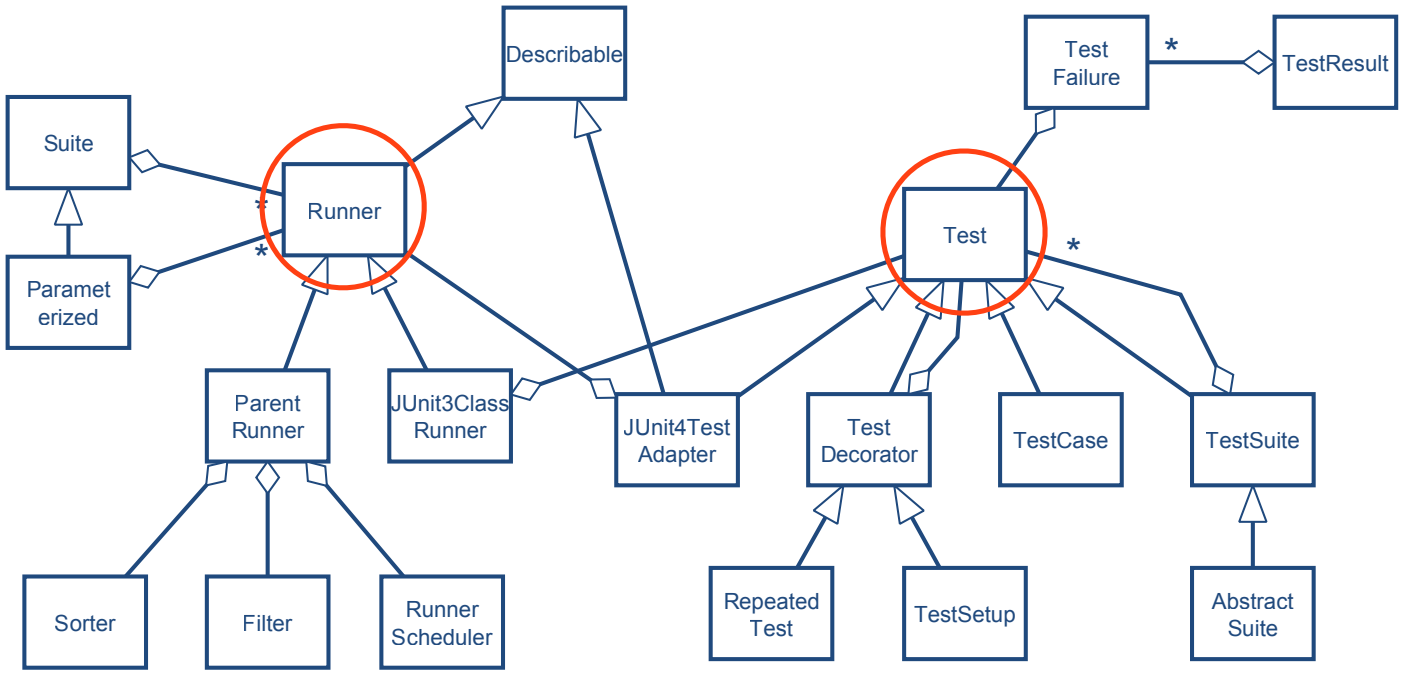
Exploration Support▶

Concept Assessment

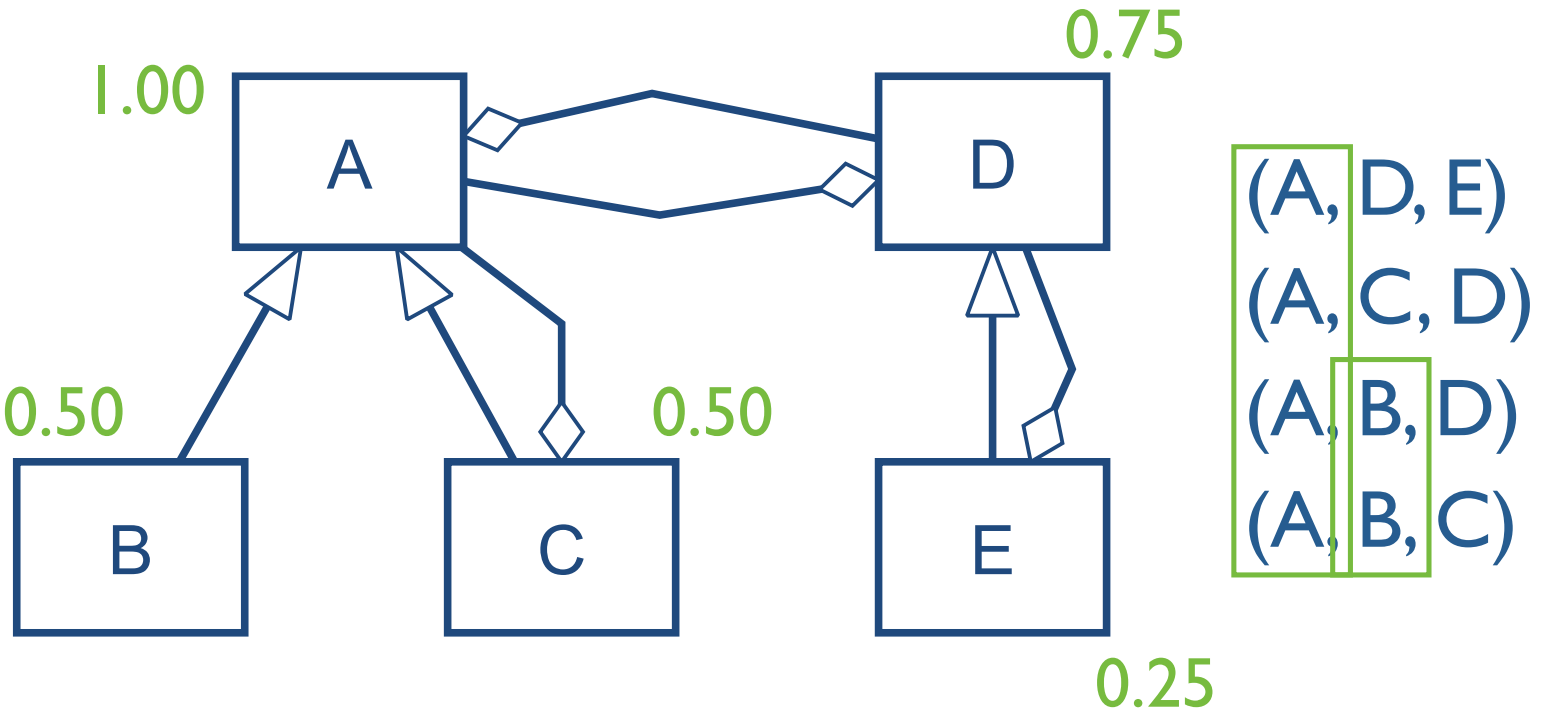
- Specificity
- Prevalence
- Importance
- ...

Exploration Support▶

Class Prominence

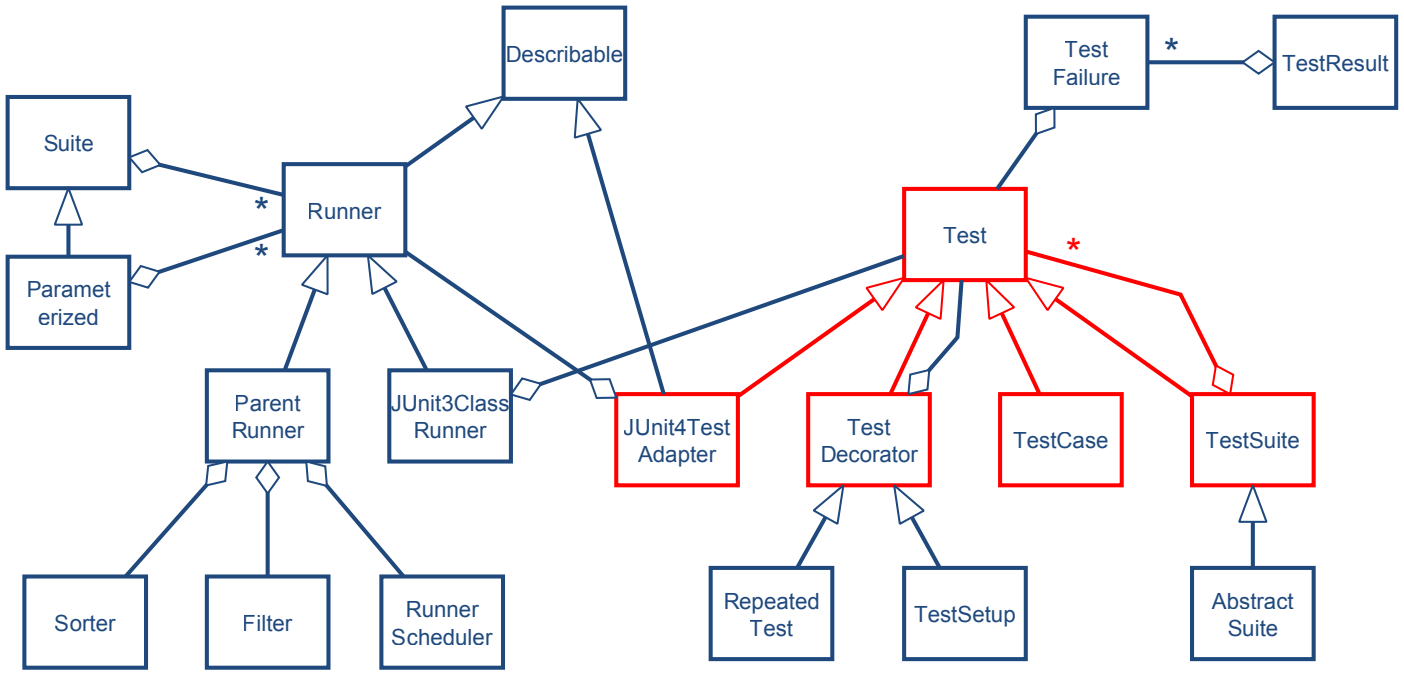


Class Prominence

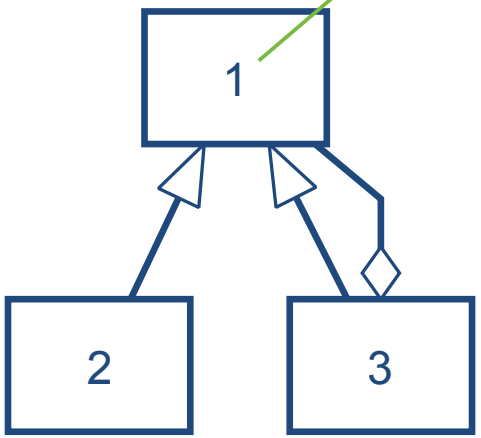


Exploration Support▶

Core Concepts



Core Concepts



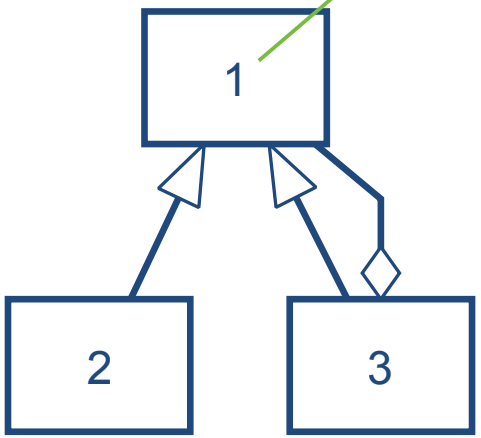
1	2	3
(Test,	TestSuite,	JUnit4TestAdapter)
(Test,	TestSuite,	TestCase)
(Test,	TestSuite,	TestDecorator)

role players

$r \in \{1, 2, 3\} \;\hat{=}\; \text{roles}$

$p_{c,r} \;\hat{=}\; \text{set of role players}$

Core Concepts



$$p_{c,1} = \{\texttt{Test}\}$$
$$p_{c,2} = \{\texttt{TestSuite}\}$$
$$p_{c,3} = \{\texttt{JUnit4TestAdapter}, \texttt{TestCase}, \texttt{TestDecorator}\}$$

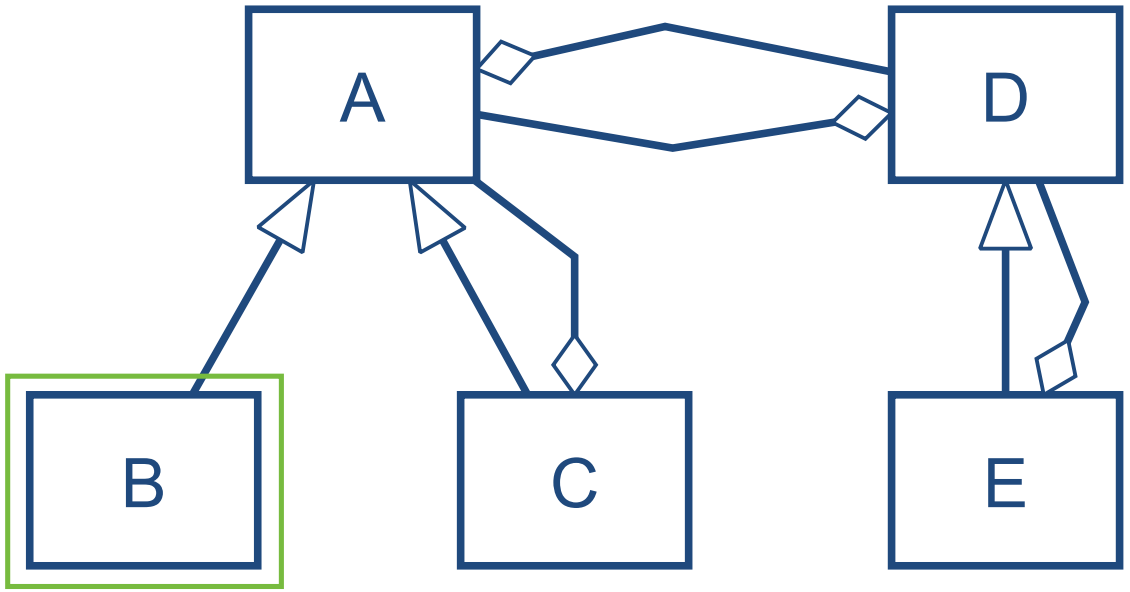
Core Concepts

$$core_c = \max_{r \in \{1,\ldots,n\}} \frac{\sum_{i \in p_{c,r}} prom_{n,i}}{|p_{c,r}|}$$

↑ prominence of role players

↓ # locations in the code

Filtering



(A, D, E)

(A, C, D)

(A, B, D)

(A, B, C)

Coming up▶

Future Work

- Comparison of ways to compute class prominence
- Evaluation
- Further concept metrics
- Extension/Refinement of Relationship Model
- Automatic name suggestion for concepts/roles

Summary▶

Contributions

- Formal Concept Analysis:
  - ✓Usable even for classes > 600.
- Concept Model
  - ✓Structural + Behavioral Relationships.
  - ✓Engaged FCA complexity by Iterative Analysis.
  - ✓Abstraction from Structures of Classes to Concepts with Roles and -Players.
- Exploration Support
  - ✓Filtering by Classes.
  - ✓Efficient Filter Suggestion through Class Prominence.
  - ✓Structure Importance Assessment

FCA▶

Structural Relations

```
public class ClassA extends SuperType {
public class ClassA implements Abstract {
```

```
private ClassB scalarClassB;
```


```
private List<ClassB> vectorClassB;
private ClassB[] arrayClassB;
```

FCA▸ Behavioral Relations

creates

```
public ClassA() {
    scalarClassB = new ClassB();
}
```

forwards

```
public void forwardsCall() {
    scalarClassB.forwardsCall();
}
```

produces

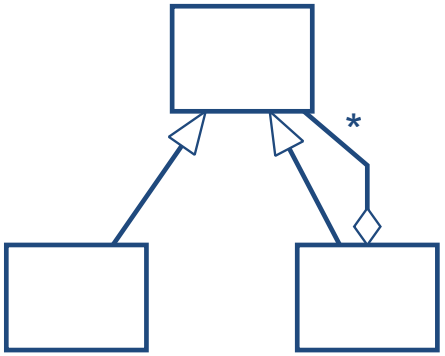
```
public ClassB producesClassB() {
    return new ClassB();
}
```

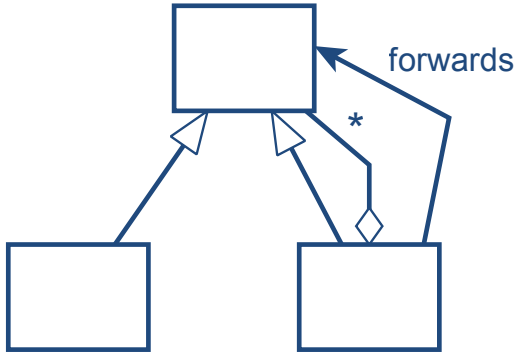
Complexity

$$O(|G|^2 \bullet |M| \bullet |C|)$$

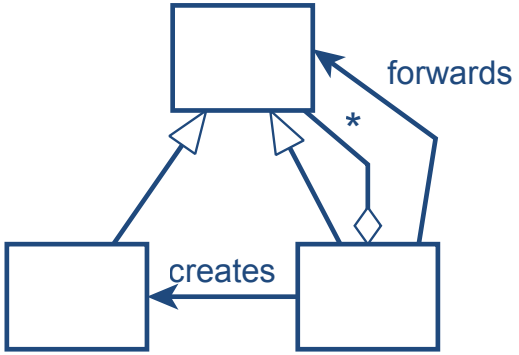
Iterative Analysis

1. Iteration
(Structural)

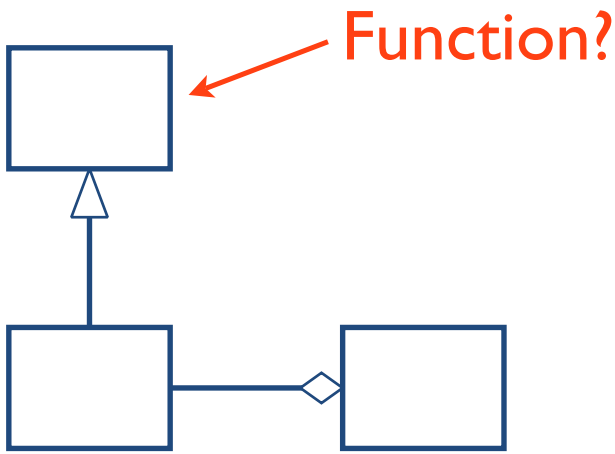




2. Iteration
(Structural +
Behavioral)



FCA▶ Unused Generalization



Appendix ▶

Number Concepts

	JUnit 4.7	Cultivate	JHotDraw 7
#classes	143	607	625
n = 2	5	7	9
n = 3	16	45	38
n = 4	24	154	141