



Motivation

Gliederung

Einleitung

Software Mining
Call-Graphen
Graph Mining

Transforma-
tionen

Ausblick

Mining Software Call Graphs

Frank Eichinger

Lehrstuhl Prof. Böhm
Institut für Programmstrukturen und Datenorganisation (IPD)
Universität Karlsruhe (TH)

2007-03-16



Motivation

Motivation

Gliederung

Einleitung

Software Mining
Call-Graphen
Graph Mining

Transformationen

Ausblick

- Software wird selten fehlerfrei ausgeliefert, auch wenn ausgiebig getestet wurde.
- Besonders anspruchsvoll zu finden sind **Noncrashing-Bugs**, da kein Stack-Trace vorliegt.
- Es stehen begrenzte Ressourcen zum Code-Review zur Verfügung, Software-Projekte sind allerdings meist zu groß für ein vollständiges Review.
- Idee:
 - Lokalisierung von **Noncrashing-Bugs** durch **Data Mining**.



Gliederung

Motivation

Gliederung

Einleitung

Software Mining

Call-Graphen

Graph Mining

Transformationen

Ausblick

- 1 Einleitung
 - Data Mining in der Softwaretechnik
 - Call-Graphen
 - Graph Mining
- 2 Transformationen von Call-Graphen
- 3 Ausblick



Data Mining in der Softwaretechnik (1)

Motivation

Gliederung

Einleitung

Software Mining

Call-Graphen

Graph Mining

Transformationen

Ausblick

- Traditionelles Data Mining arbeitet mit Feature-Vektoren von numerischen und kategorischen Werten.
- In der Softwaretechnik können dies z.B. verschiedene Code-Metriken sein.

| | METRIK 1 | METRIK 2 | METRIK 3 | ... |
|---------------------|----------|----------|----------|-----|
| Software-Artefakt 1 | 123 | 5 | 12 | ... |
| Software-Artefakt 2 | 222 | 8 | 12 | ... |
| ... | ... | ... | ... | ... |

- Es werden Muster oder Eigenschaften gesucht, die in fehlerhafter Software auftreten, nicht aber in korrekter.
 - Dazu muss Software von Benutzern als **korrekt** oder **fehlerhaft** klassifiziert werden.



Data Mining in der Softwaretechnik (2)

Motivation

Gliederung

Einleitung

Software Mining

Call-Graphen

Graph Mining

Transforma-
tionen

Ausblick

- Probleme mit Feature-Vektoren von Code-Metriken:
 - Auch viele Metriken beschreiben ein Programm nur unzureichend.
 - Code-Metriken helfen nur wenig bei der Lokalisierung von Fehlern.
- Idee:
 - Stelle Programmausführungen als **Call-Graph** dar und analysiere die Graph-Struktur.



Call-Graphen

Motivation

Gliederung

Einleitung

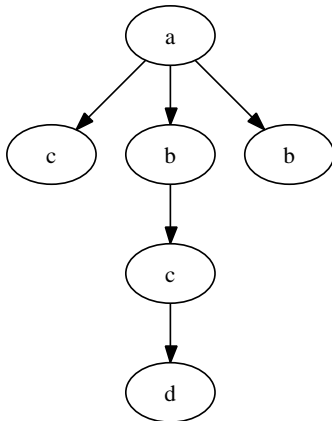
Software Mining

Call-Graphen

Graph Mining

Transformationen

Ausblick



- Programmausführungen lassen sich wie folgt auf Graphen abbilden:
 - Methoden stellen Knoten dar,
 - Methodenaufrufe Kanten.
- Es entsteht ein Wurzelbaum mit der Main-Methode als Wurzelknoten.



Frequent Subgraph Mining (1)

Motivation

Gliederung

Einleitung

Software Mining

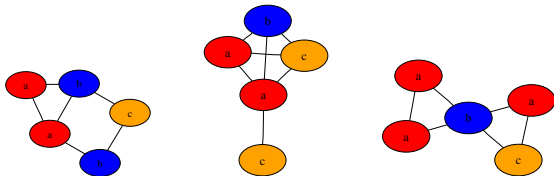
Call-Graphen

Graph Mining

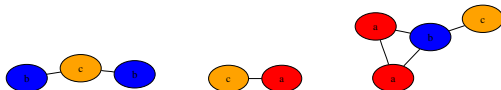
Transformationen

Ausblick

Graph-Datenbank \mathcal{D} :



Teilgraphen (Auswahl):



Support:

1

2

3



Frequent Subgraph Mining (2)

Motivation

Gliederung

Einleitung

Software Mining

Call-Graphen

Graph Mining

Transformationen

Ausblick

- Gegeben:
 - Graph-Datenbank \mathcal{D}
 - Support-Schwellwert $minSup$
- Gesucht:
 - Alle Teilgraphen, die in mindestens $minSup$ Graphen aus \mathcal{D} vorkommen.
- Algorithmen:
 - *AGM, FSG*
 - *gSpan, CloseGraph*
 - *MoFa, FFSN, Gaston*
 - ...
- Grundlage aller Algorithmen:
 - Subgraphisomorphie-Problem (NP-Vollständig)



Graph Mining mit Call-Graphen

Motivation

Gliederung

Einleitung

Software Mining

Call-Graphen

Graph Mining

Transformationen

Ausblick

- 1 Voraussetzung:
 - Datenbank mit als **korrekt** oder **fehlerhaft** klassifizierten Programmläufen.
 - Diese Klassifikation von Programmläufen wird manuell von Benutzern bzw. durch automatisierte Tests vorgenommen.
- 2 Finde häufige Teilgraphen.
- 3 Identifiziere Teilgraphen, die in der Menge **fehlerhaft**, nicht aber in der Menge **korrekt** vorkommen.
→ Die Methoden dieser Teilgraphen sind mit erhöhter Wahrscheinlichkeit für Fehler verantwortlich.



Vollständige Reduktion

Motivation

Gliederung

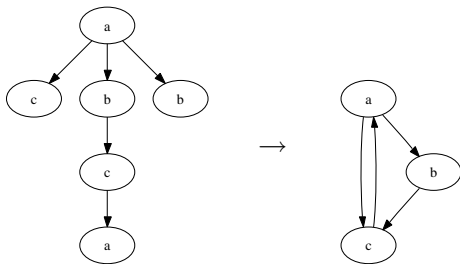
Einleitung

Software Mining
Call-Graphen
Graph Mining

Transformationen

Ausblick

- Call-Graphen können vollständig reduziert werden, indem man alle gleichartigen Knoten fusioniert.



- Beispiel-Graph: Reduktion von 743 auf 362 Knoten.
- Problem: Möglicher Verlust von wichtiger Information.
- Ansatz: Zusammenfassung von Iteration und Rekursion, Einführung von Kantengewichten.



Zusammenfassung von Iteration

Motivation

Gliederung

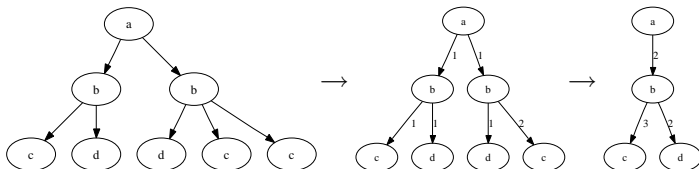
Einleitung

Software Mining
Call-Graphen
Graph Mining

Transformationen

Ausblick

- Teilgraphen mit gleicher Struktur werden zusammengefasst.
- Kantengewichte beschreiben die Auftrittshäufigkeit.



- Vorgehen:
 - Ebenenweise von Unten nach Oben.



Zusammenfassung von Rekursion (1)

Motivation

Gliederung

Einleitung

Software Mining

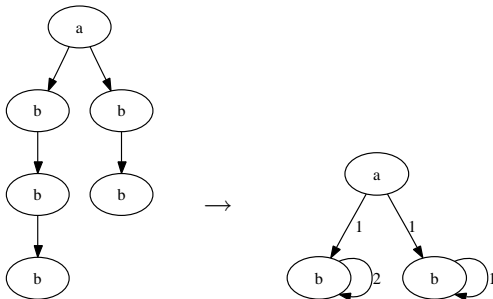
Call-Graphen

Graph Mining

Transformationen

Ausblick

- Einfache Rekursionen lassen sich durch Schlingen (“self-loops”) darstellen.
- Kantengewichte geben hier die Rekursionstiefe an.



- Anschließend kann eine Zusammenfassung der Iterationen stattfinden.



Zusammenfassung von Rekursion (2)

Motivation

Gliederung

Einleitung

Software Mining

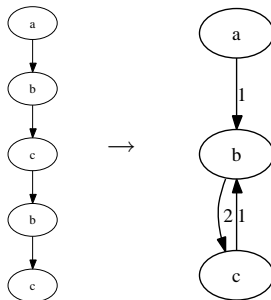
Call-Graphen

Graph Mining

Transforma-
tionen

Ausblick

- Rekursionen über mehrere Knoten lassen sich durch Zyklen darstellen.





Zusammenfassung von Rekursion (3)

Motivation

Gliederung

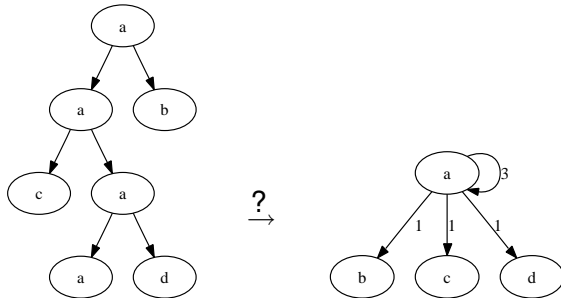
Einleitung

Software Mining
Call-Graphen
Graph Mining

Transformationen

Ausblick

- Offene Frage: Wann sollen Rekursionen nicht zusammengefasst werden?





Aktuelle und zukünftige Forschung

Motivation

Gliederung

Einleitung

Software Mining

Call-Graphen

Graph Mining

Transformationen

Ausblick

- Erweiterung von Graph Mining für weitere Graph-Klassen
 - Gerichtete Graphen
 - Multigraphen
 - Graphen mit Schlingen
 - Gewichtete Graphen
 - Eventuell durch intelligente Vor- oder Nachverarbeitung, z.B. in Kombination mit anderen Data Mining Techniken.
- **Mining von großen Graphen**
 - **Durch approximative Algorithmen**
- Evaluation der verschiedenen Call-Graph-Transformationen
- Anwendungen (nicht nur) in der Softwaretechnik



```
i40fs1:~eichi> ping karlsruhe  
ping: unknown host karlsruhe  
i40fs1:~eichi> _
```

Motivation

Gliederung

Einleitung

Software Mining

Call-Graphen

Graph Mining

Transforma-
tionen

Ausblick

Vielen Dank für Ihre Aufmerksamkeit!