

# Szenario für Merkmalsuche

$$(O_1, A_1) \leq (O_2, A_2)$$
$$\Leftrightarrow O_1 \subseteq O_2 \quad / \quad A_2 \subseteq A_1$$

- Ihre Aufgabe ist es, für ein System mit mehr als 10,000 Routinen
  - ein Merkmal hinzuzufügen, zu korrigieren oder zu ändern
  - oder einfach die Implementierung eines Merkmals zu verstehen.
- Wo beginnen Sie?
- Merkmalsuche:
  - welche Komponenten eines Systems implementieren ein einzelnes Merkmal bzw. eine Menge von Merkmalen?

# Merkmale und Komponenten

Ein **Merkmal** ist eine realisierte funktionale oder nicht-funktionale Eigenschaft des Systems.

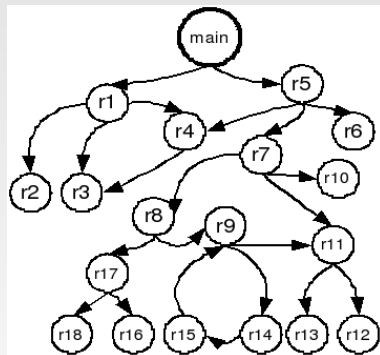
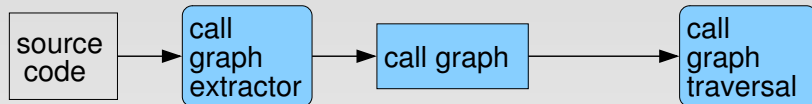
Eine **Komponente** ist eine ausführbare Einheit, z.B.:

- eine einzelne Anweisung oder ein Ausdruck,
- eine Routine (Funktion, Prozedur),
- ein Modul,
- ein Subsystem,
- eine Task, ein Thread oder Prozess.

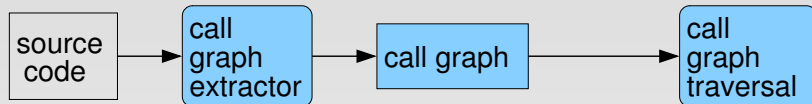
Im Folgenden:

- Merkmal = von außen anstoßbares und beobachtbares Verhalten des Systems, das auf Komponenten abgebildet werden kann.
- Komponente = Routine.

# Statische Analyse nach Chen und Rajlich (2000)

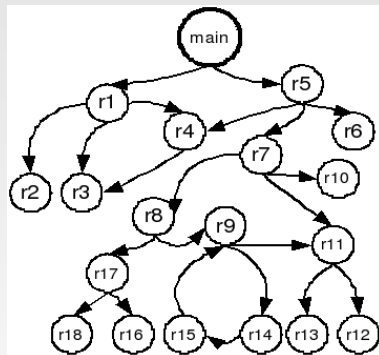


# Statische Analyse nach Chen und Rajlich (2000)

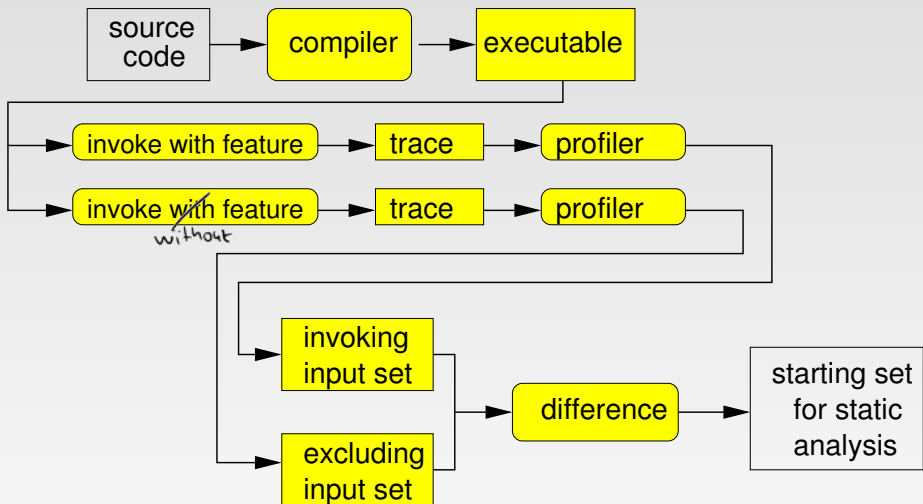


Probleme:

- Wo beginnen?
- Wo fortfahren?
- Wo aufhören?
- Präzision der statischen Extraktion?

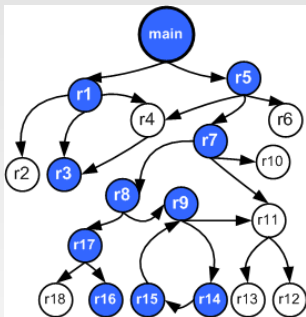


# Zwei dynamische Ausführungs-Traces nach Wilde u. a. (1992)



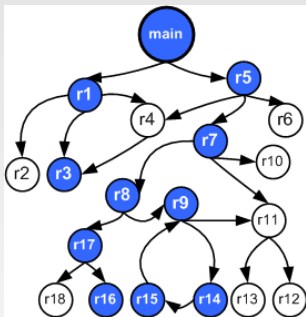
# Dynamische Analyse

Dynamischer  
Aufrufgraph unter  
Verwendung des  
Merkmals

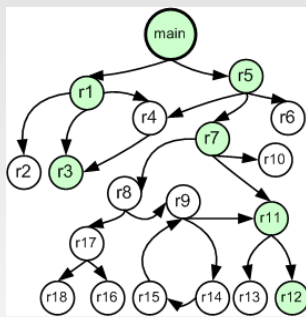


# Dynamische Analyse

Dynamischer  
Aufrufgraph unter  
Verwendung des  
Merkmals

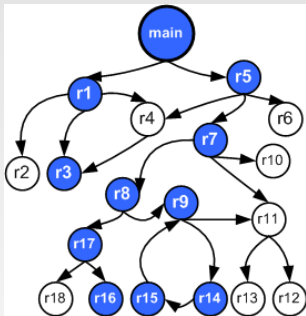


Dynamischer  
Aufrufgraph **ohne**  
Verwendung des  
Merkmals

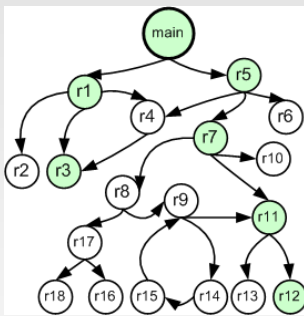


# Dynamische Analyse

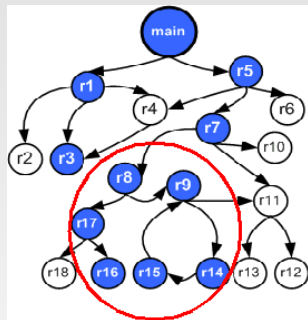
Dynamischer  
Aufrufgraph unter  
Verwendung des  
Merkmals



Dynamischer  
Aufrufgraph **ohne**  
Verwendung des  
Merkmals



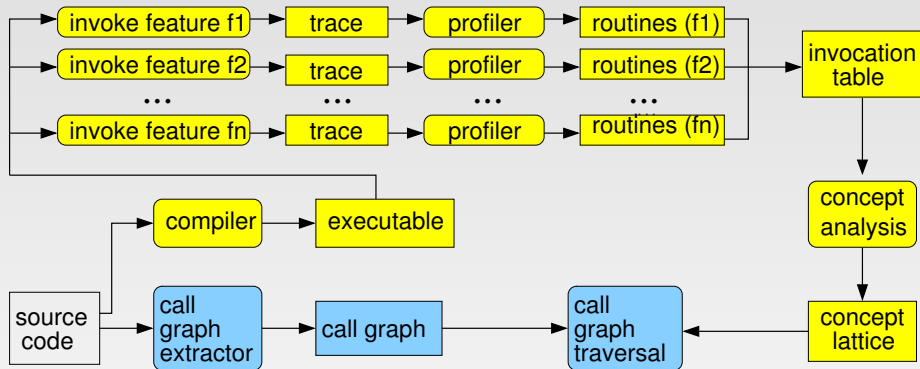
Resultat





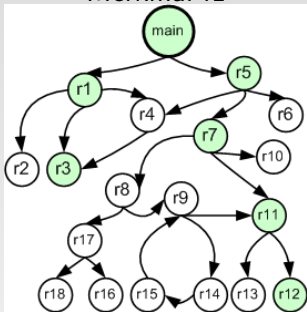
- Invoking-Input-Set – Excluding-Input-Set kann
  - immer noch sehr viele nicht-merkmalspezifische Routinen
  - sowie nicht wirklich merkmalspezifische Routinen enthalten.
- Resultate hängen von der Eingabe ab;
  - d.h. Resultate bilden nur einen Startpunkt für weitere statische Analysen.
- Wo soll mit der statischen Analyse begonnen werden?
  - Das Resultat differenziert die identifizierten Routinen nicht weiter.

# Statisch und dynamisch nach Eisenbarth u. a. (2003)



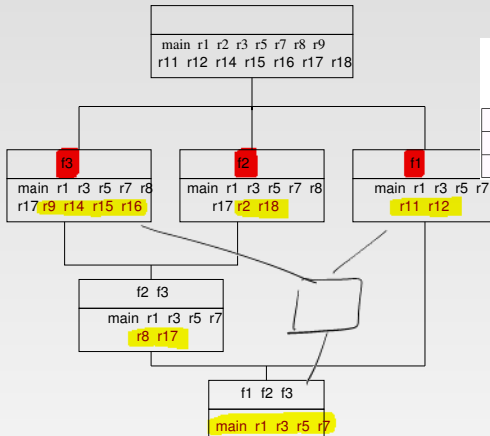
# Begriffsanalyse I

Merkmal f1



	main	r1	r2	r3	r5	r7	r8	r9	r11	r12	r14	r15	r16	r17	r18
Merkmal f1	●	●		●	●	●			●	●					
Merkmal f2	●	●	●	●	●	●	●							●	●
Merkmal f3	●	●		●	●	●	●	●			●	●	●	●	

# Begriffsanalyse II



	main	r1	r2	r3	r5	r7	r8	r9	r11	r12	r14	r15	r16	r17	r18
Merkmal f1	●	●		●	●	●			●	●					
Merkmal f2	●	●	●	●	●	●	●							●	●
Merkmal f3	●	●		●	●	●	●	●			●	●	●	●	

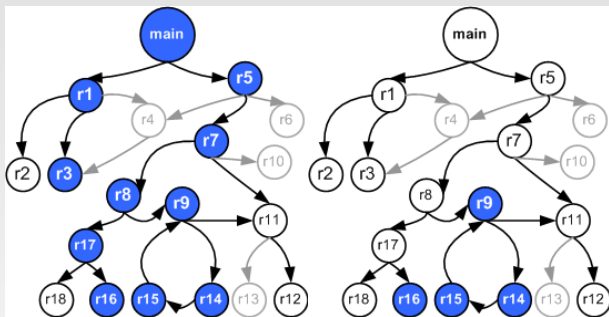
Kontext:

- Routinen → Objekte
- Merkmale → Attribute
- $r$  wird aufgerufen für  $m \rightarrow$  Relation

Begriffsverband

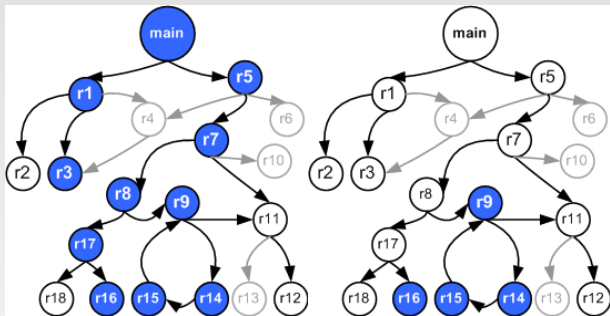
# Kombination

Aufrufgraphen **ohne** Information des Begriffsverbands



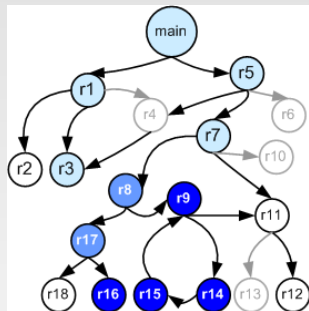
# Kombination

Aufrufgraphen **ohne** Information des Begriffsverbands



Top-down / Bottom-up-Traversierung anhand der Merkmalspezifizität der Nachbarn.

Aufrufgraph **mit** Information des Begriffsverbands



# Fallstudie zweier Web-Browser

Kombination statischer und dynamischer Information angewandt auf zwei Web-Browser:

- Mosaic (51 KLOC C-Code, ohne Bibliotheken)
- Chimera (38 KLOC C-Code; ohne Bibliotheken)

Zwei Experimente:

- Partielle Architektur der Browser-History
  - Merkmale: Browsing, Back-Button, Forward-Button
- Partielle Architektur der Bookmarks
  - Merkmale: Öffnen/Schließen des Bookmark-Fensters, Bookmark hinzufügen/löschen/selektieren

# Reduktion des Suchraumes

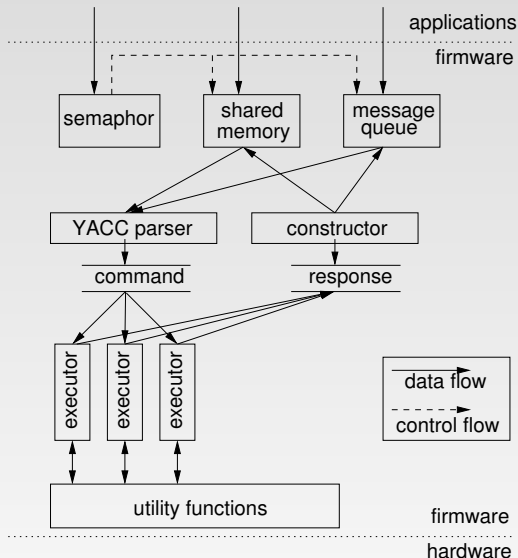




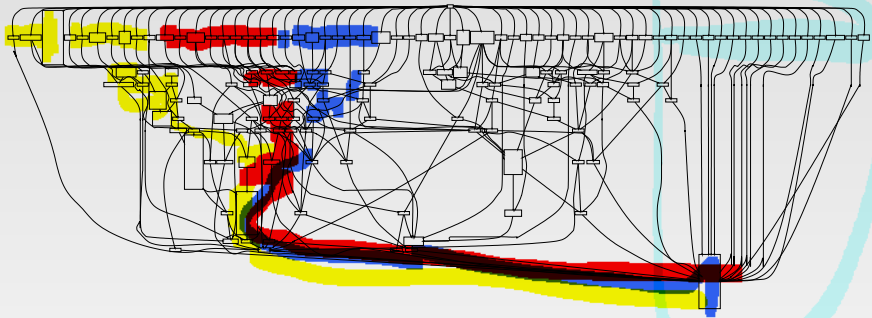
# Fallstudie: Architektur der Firmware eines Chiptesters



# Fallstudie: Architektur der Firmware eines Chiptesters



# Begriffsverband für die Firmware



# Inkrementelle Begriffsanalyse I

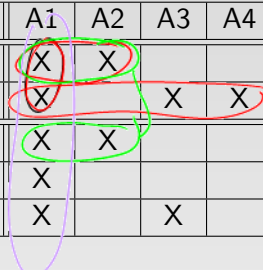
	A1	A2	A3	A4	A5
O1	X	X			
O2	X		X	X	
O3	X	X			
O4	X				X
O5	X		X		

Sei  $K = (\mathcal{O}, \mathcal{A}, \mathcal{I})$  ein Kontext,  $\mathcal{O}' \subseteq \mathcal{O}$ , und  $\mathcal{A}' \subseteq \mathcal{A}$ . Dann ist  $K' = (\mathcal{O}', \mathcal{A}', \mathcal{I} \cap (\mathcal{O}' \times \mathcal{A}'))$  ein **Subkontext** von  $K$  und  $K$  ist ein **Superkontext** von  $K'$ . □

Annahme: Wir fügen nur Zeilen dazu (für Spalten ist es analog).

# Inkrementelle Begriffsanalyse II

	A1	A2	A3	A4	A5
O1	X	X			
O2	X		X	X	
O3	X	X			
O4	X				X
O5	X		X		



**Satz.** Sei  $K = (\mathcal{O}, \mathcal{A}, \mathcal{I})$  und  $K' = (\mathcal{O}', \mathcal{A}, \mathcal{I}')$ , wobei  $\mathcal{O}' \subseteq \mathcal{O}$  und  $\mathcal{I}' = (\mathcal{I} \cap (\mathcal{O}' \times \mathcal{A}))$ . Dann ist jeder *Inhalt* von  $K'$  ein *Inhalt* von  $K$ . □

BEWEIS. Siehe Ganter und Wille (1996).

**Folgerung.** Gemäß dieses Satzes erscheint jeder Intent des Subkontexts im Superkontext. Daraus resultiert die folgende Abbildung:

$$(O, A) \mapsto (\sigma(A), A)$$

BEWEIS. Siehe Ganter und Wille (1996).