

# *Ausgewählte Kapitel aus dem Übersetzerbau*

*Dr. Sabine Glesner*  
Fakultät für Informatik  
Universität Karlsruhe

Sommersemester 2005

©Goos, Glesner 2005

<http://www.info.uni-karlsruhe.de/>

*Globale kontextsensitive Wertanalyse*

—

*$\chi$ -Terme*

# Vorlesungsplan

---

- Einführung (1)
- Datenflußanalyse und monotone Frameworks (1-3)
- **SSA-Darstellungen und sequentielle Optimierungen (4-7)**
  - SSA Konstruktion (1. Foliensatz)
  - Optimierungen auf SSA-Form (2. Foliensatz)  
Operatorvereinfachung, Eliminierung gemeinsamer Teilausdrücke (CSE),  
Eliminierung partieller Redundanzen (PRE)
  - Speicher SSA
  - **Globale kontextsensitive Wertanalyse auf SSA-Form**
- Cache Optimierung (8-10)
  - Caches und ihre Problematik
  - Techniken zur Cacheoptimierung (und zur Parallelisierung):  
Schleifenoptimierungen für Reihungen, Optimierungen für dynamische  
Datenstrukturen, Vorladen und Befehlsanordnung
- Weitere Optimierungen (10)
  - Nachoptimierung, Registerzuteilung, Befehlsanordnung
- Nebenläufige Sprachen (11)
  - Begriffe und Konzepte
  - Parallele Hardware-Architekturen
  - Implementierung von Parallelität
- Werkzeuge (12): Cocktail, BEG, ELI, Suif, Trimaran, (PAG, Firm)
- Evtl. Übersetzerverifikation (12)



# Inhalt – SSA

---

- Optimierungen auf SSA-Form
  - Operatorvereinfachung
  - Eliminierung gemeinsamer Teilausdrücke
  - Eliminierung partieller Redundanzen
- Speicher SSA
  - Haldenoperationen
  - Alias-Problematik
  - Points-To-Analyse
  - Namensschema
  - Iterative Analyse
- Globale kontextsensitive Wertanalyse
  - Interprozedurale Analyse
  - Nichtrealisierbare Pfade
  - $\chi$ -Terme
  - Symbolisches Rechnen mit  $\chi$ -Termen



# Interprozedurale Analyse

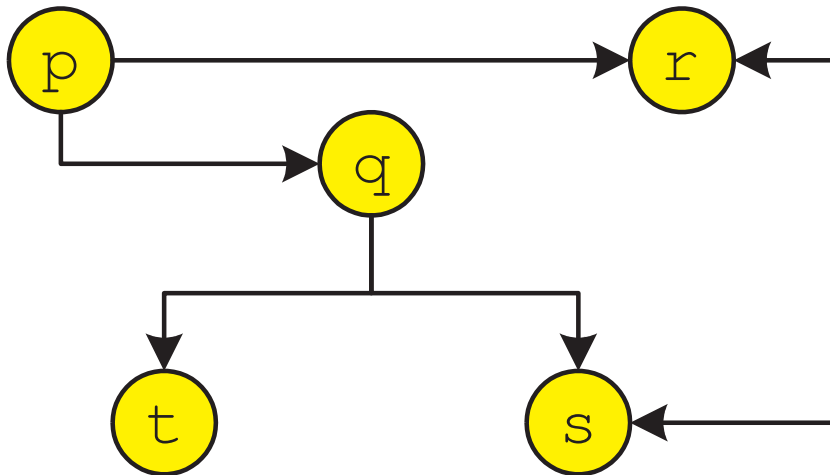
---

- Aufgaben:
  - Wirkung von Prozeduraufrufen auf den Aufrufer bestimmen:
    - \* rekursive Prozeduren feststellen
    - \* möglicherweise veränderte Werte, Nebenwirkungen
    - \* Reihenfolgebedingungen: welche Aufrufe dürfen vertauscht werden
  - Bestimmen aufgerufener Prozeduren (bei Polymorphie, Prozedurvariablen)
  - Welche Prozedurrümpfe sollen offen eingebaut werden
- Zweck:
  - Aufrufe nicht-rekursiver Prozeduren/Endrekursion vereinfachen
  - Wertnumerierung interprozedural fortsetzen
  - indirekte Prozeduraufrufe durch direkte ersetzen
  - Prozeduren offen einbauen
- Voraussetzungen:
  - vollständiges Programm + Laufzeitsystem liegt vor (keine getrennte Übersetzung)
  - Aufrufgraph
  - Kontrolle des Speicheraufwands und Methodik mit unvollständigem Wissen umzugehen



# Einfacher Aufrufgraph

- Ecken sind Prozeduren, Kanten sind Aufrufe



```
void p() {  
    r(); q(); r() };  
void q() {  
    if B then s()  
    else t() };  
void r() {  
    if B then s() };  
void s() {  
    if not B then r() };  
void t() { ... }
```

- Hauptprogramm wie Prozedur behandeln
  - Zyklen kennzeichnen potentielle Rekursionen
  - Mehrfachaufrufe nur einmal dargestellt
  - *keine Verfolgung Aufruf → Prozedur → Rückkehr möglich*
  - bei OO-Programmen: oft lange Ketten von Aufrufen
  - Anzahl Blätter im Aufrufgraph > 50%



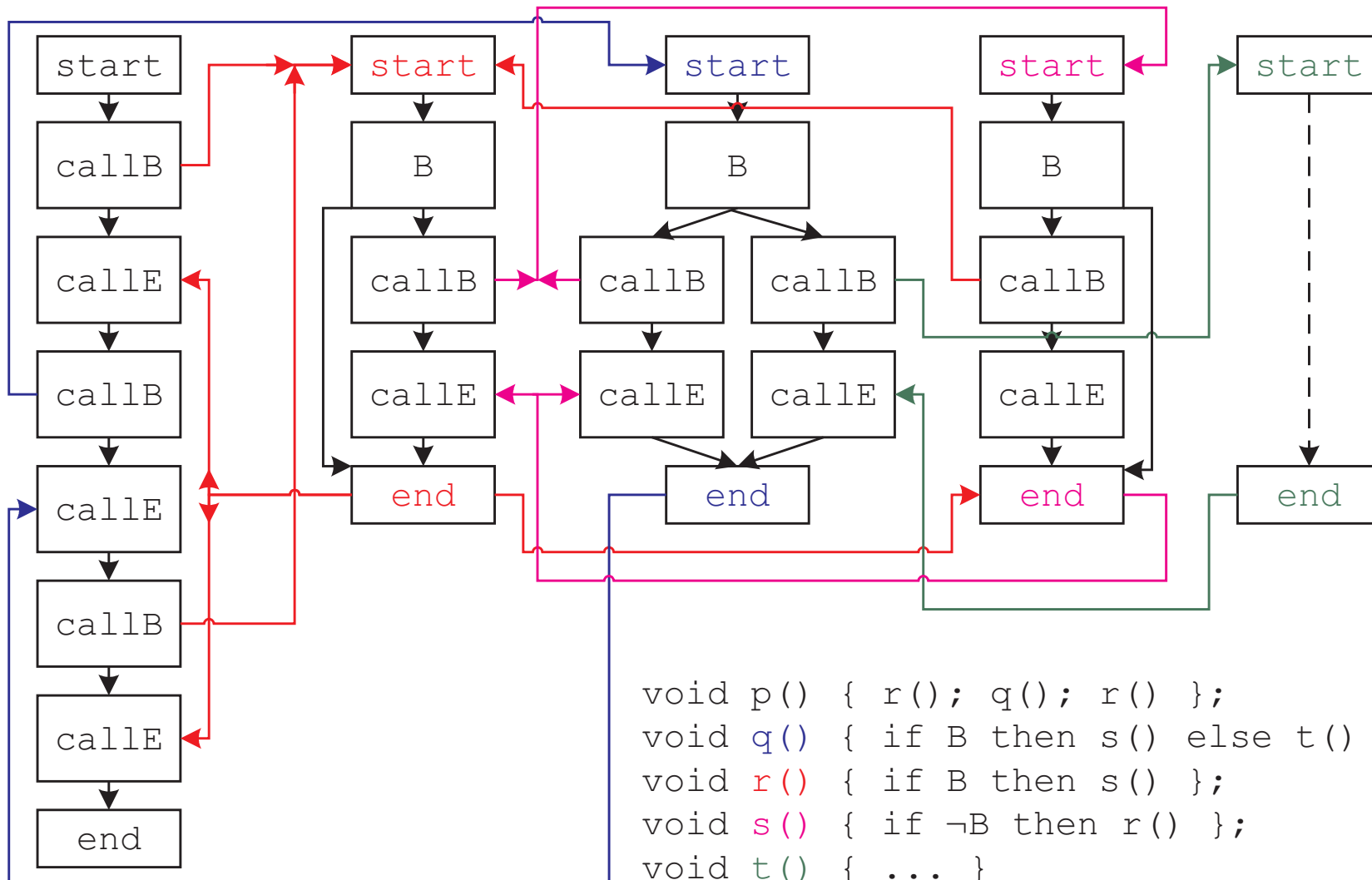
# Komplexer Aufrufgraph

---

- Neue Ecken im SSA Graph
  - `callBegin`: Aufruf einer Prozedur
  - `callEnd`: Übernahme von Ergebnissen in den Kontext des Aufrufers
- Neue Kanten im SSA Graph
  - Kante von `callBegin` Ecke zu `start` Ecke der aufgerufenen Prozedur
  - Kanten von `return` Ecke zu den `callEnd` Ecken möglicher Aufrufer
  - Kante von `callBegin` Ecke zur `callEnd` Ecke
  - Kanten von Argumenten des Aufrufers zu formalen Parametern der Prozedur ( $\phi$ -Funktion bei mehreren Aufrufen)
  - Kanten von Resultaten zu Verwendung der Resultate im Aufrufer
- Polymorphie wird (zunächst) durch explizite Verteilerprozedur aufgelöst
- Interprozedurale Datenflußanalyse jetzt möglich



# Beispiel – Komplexer Aufrufgraph





# Interprozedurale Wertanalyse

---

Alles wie bisher, jedoch:

- initialisiere Werte am Prozeduranfang nicht mit undefiniert, sondern übernahm Werte der Aufrufer
- führe nach Rückkehr keine neuen Wertnummern ein, sondern übernahm Werte der aufgerufenen Prozedur
- führe  $\phi$ -Funktionen bei unterschiedlichen Wertnummern ein
- führe offene  $\phi'$ -Funktionen ein, wenn Werte einer Prozedur noch nicht bekannt
- Fixpunktiteration, bis keine algebraischen Identitäten mehr erkennbar



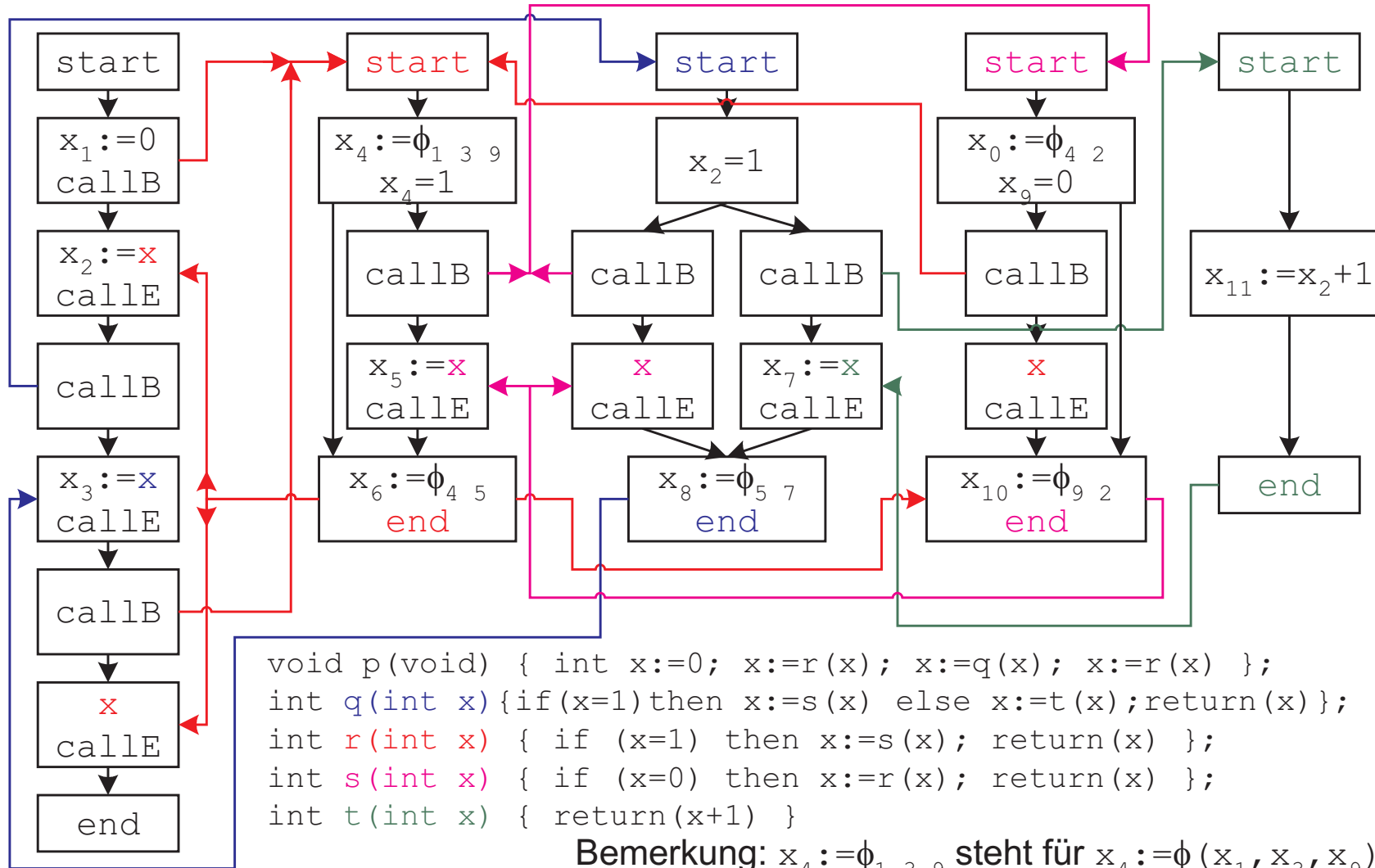
# Beispiel

---

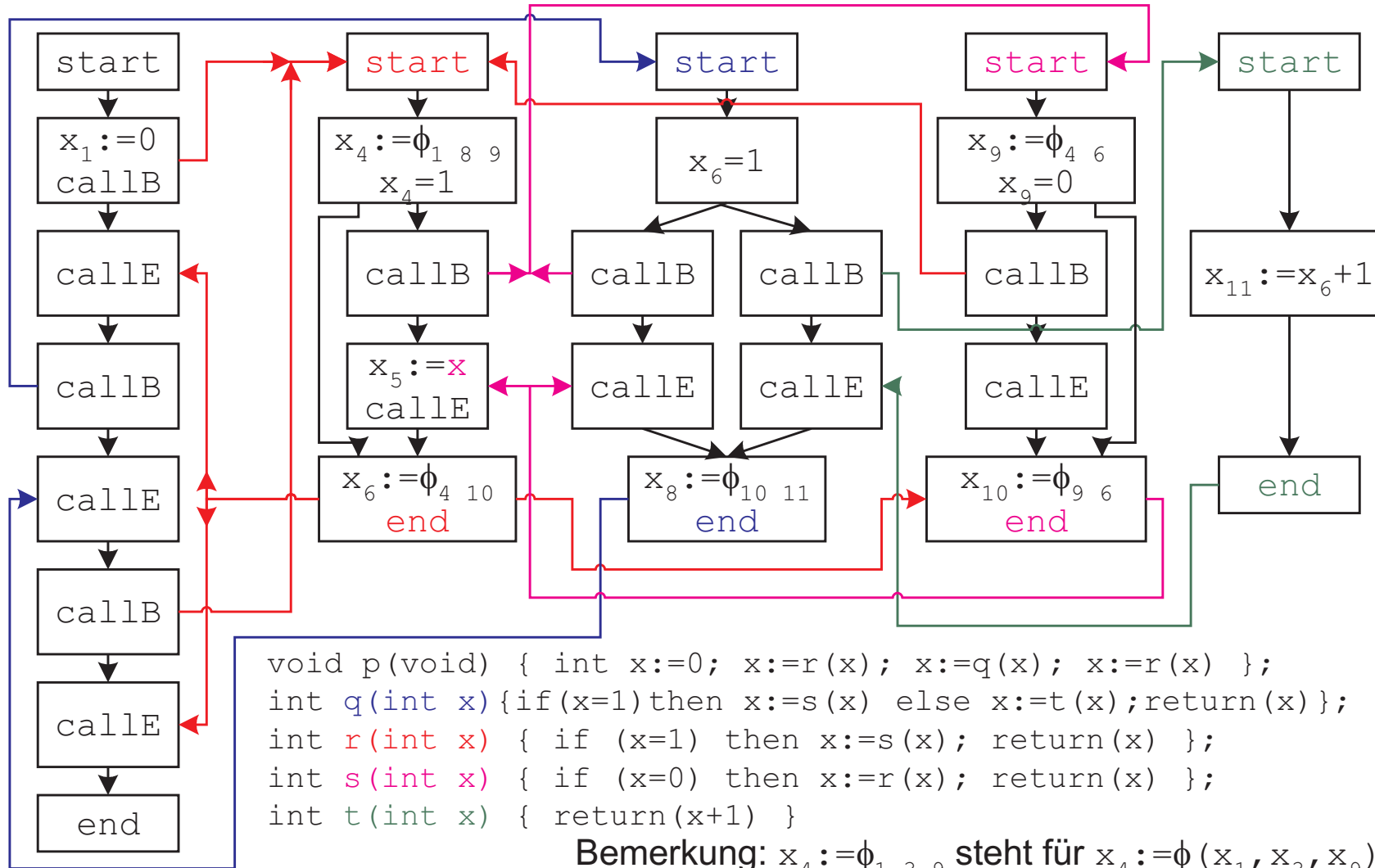
```
void p(void) { int x=0; x=r(x); x=q(x); x=r(x) };
int q(int x) { if (x==1) then x=s(x) else x=t(x); return(x) };
int r(int x) { if (x==1) then x=s(x); return(x) };
int s(int x) { if (x==0) then x=r(x); return(x) };
int t(int x) { return(x+1) }
```



# Beispielanalyse



# Beispielanalyse



## Nicht realisierbare Pfade

---

- Information wird vermischt
  - beim Aufruf einer Prozedur
  - beim Rücksprung aus einer Prozedur werden auch unmögliche Pfade betrachtet
- Keine Unterscheidung der Aufrufkontexte möglich
- Vergleich
  - intraprozedurale Analyse:
    - \* (Datenfluß-)Werte sind undefiniert am Prozedurbeginn
    - \* nach Aufruf einer Prozedur sind alle Werte undefiniert (neue Wertnummern)
  - interprozedurale Analyse:
    - \* (Datenfluß-)Werte sind Supremum der Werte der Aufrufe am Prozedurbeginn
    - \* Aufruf einer Prozedur setzt Werte auf Werte der aufgerufenen Prozedur, dabei auch unmögliche Pfade
- **Problem:** oft zu starke Abstraktion
- **Lösungsidee:** Unterscheide Kontexte (Namensschema)



# Lösung: Kontextsensitive Analyse

---

- Idee:
  - Unterscheide an Zusammenflüssen im (interprozeduralen) Steuerfluß die Werte unterschiedlicher Pfade
  - Pfad  $:=$  Kontext
  - Unterscheide lediglich Pfade mit unterschiedlichen Enden, dadurch nur endlich viele Kontexte
- Sei  $(L, sup)$  der (monotone Datenfluß-)Rahmen einer kontextinsensitiven Analyse, so ist  $(K \times L, sup')$  der Rahmen der dazugehörigen kontextsensitiven Analyse mit
  - Kontext  $K$
  - $sup'$  und Transferfunktionen für jeden einzelnen Kontext mit  $sup$  bzw. der jeweiligen kontextinsensitiven Transferfunktionen berechnet



# Wiederholung: Namensschemata

---

- Zunächst: Speicher ist ein (Werte-) Objekt
- gewünscht: feinere Zerlegung des Speichers in kleinere Objekte
- Unterscheide
  - Halde und Keller
  - lokale Reihungen unterschiedlicher Namen
  - disjunkte Indexmengen einer Reihung (gerade/ungerade etc.)
  - Objekte unterschiedlicher Typen von Haldenobjekten
  - Objekte gleicher Typen mit syntaktisch unterscheidbarer Erzeugungsstelle
  - Objekte gleicher Erzeugungsstelle mit syntaktische unterscheidbarem Weg zur Erzeugungsstelle (Ausführungskontext)
- Unterscheidungsschema heißt Namensschema



# Kontextsensitive Analyse

---

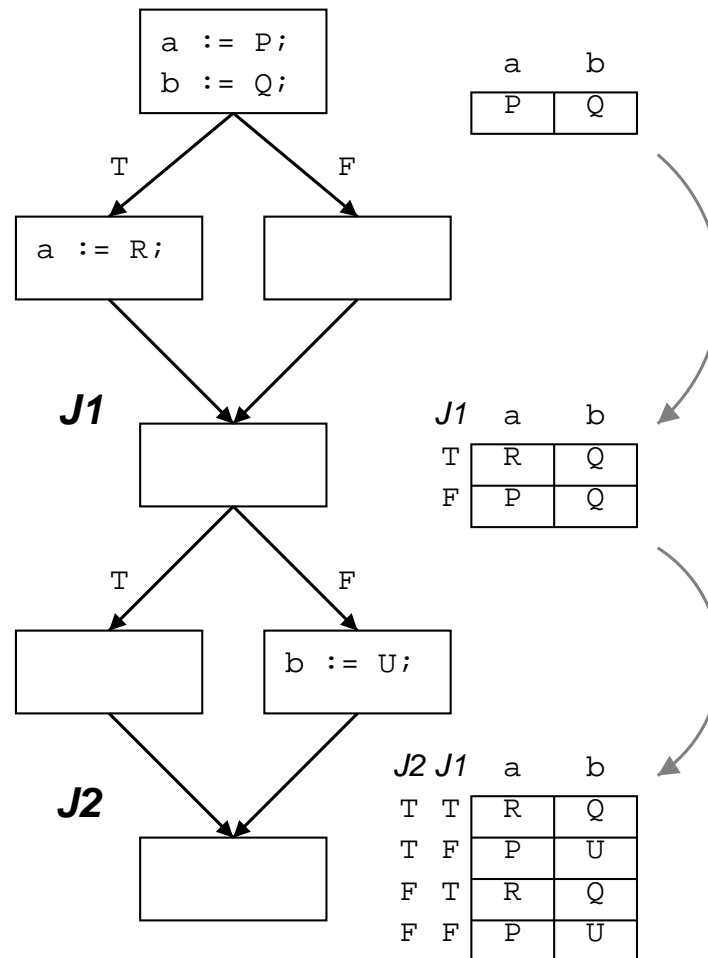
- Beim Zusammenfluß von  $n$  Ablauf-Vorgängern entstehen  $n$  *Kontexte*
- *Kontextsensitive Analyse*: Getrennte Betrachtung der Kontexte
- Anzahl der Kontexte wächst exponentiell mit der Länge des Programms
- Problem: Bei naiver Repräsentation von kontextsensitiver Analyseinformation wächst diese exponentiell
- Ausweg: Verschmelzung von Informationen verschiedener Kontexte
- Beobachtung: Die Anzahl *unterschiedlicher* Analyseinformationen wächst häufig fast nur linear
- Gesucht: Repräsentation von Analyseinformation, die mit der Anzahl unterschiedlicher Analyseinformation wächst





# Beispiel: Kontexte

Speicherung kontextsensitiver Analyseinformationen in Tabellen



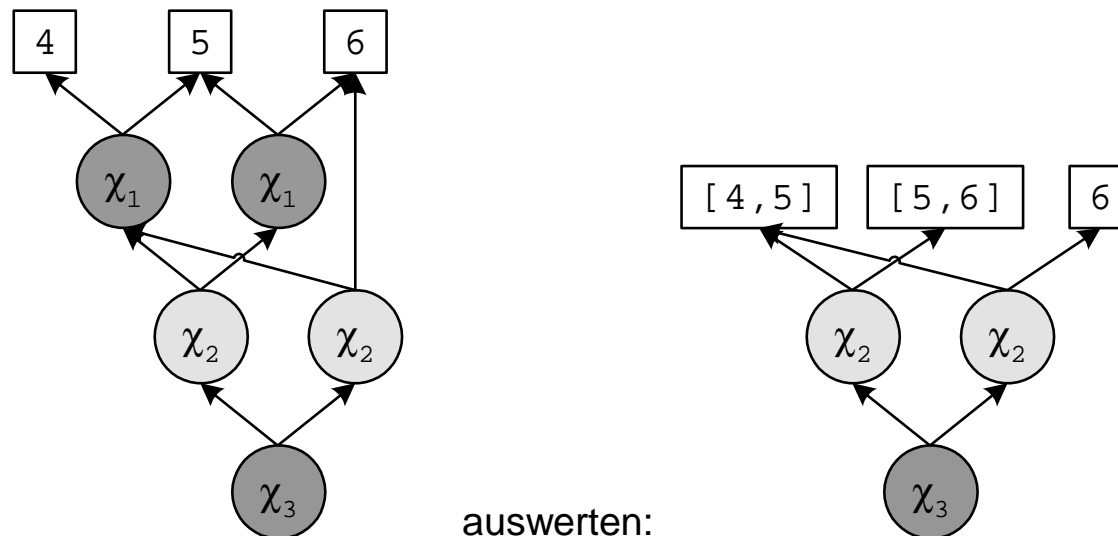
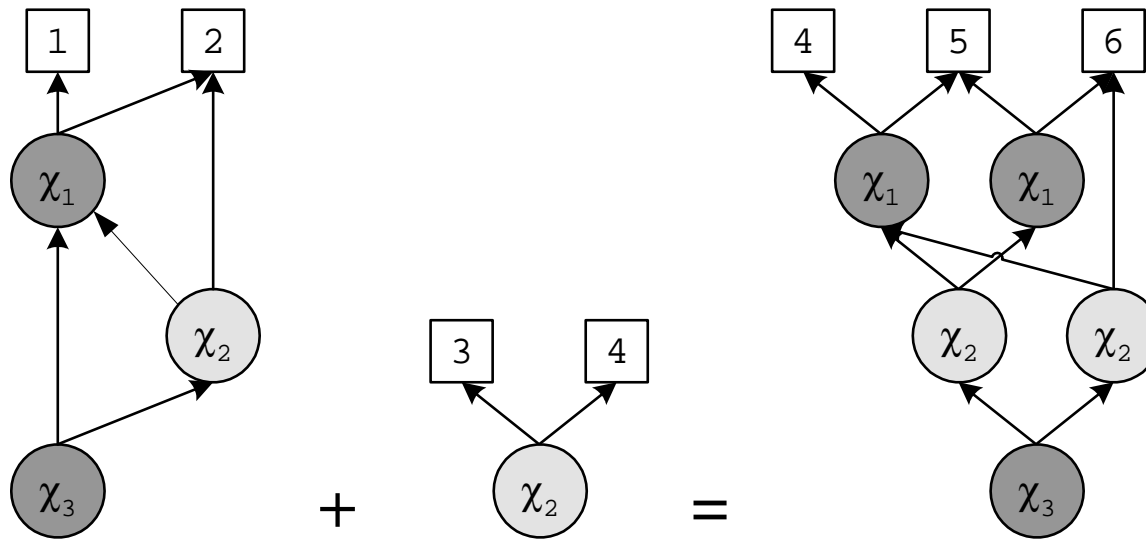
# Kontextsensitive Analyse auf SSA Graphen

---

- Idee  $\chi$ -Terme (Trapp 1999):
  - Merke symbolisch Berechnungen einzelner Kontexte bei Zusammenflüssen im (interprozeduralen) Steuerflußgraphen
  - Merke symbolisch die dazugehörige Berechnung, die das Aufspalten der Kontexte bewirkt hat
    - \* Bedingte Anweisung
    - \* Schleife
    - \* Prozeduraufruf
- Symbolisches Rechnen (Vereinfachung von Termen), wenn möglich
  - verzögere Auswertung (kontextinsensitive Verschmelzung mit Informationsverlust) der Terme bis
  - kein Speicher mehr zuteilbar
  - unterschiedliche Aufrufkontexte wieder vereinfacht werden können (nach Aufruf der Prozedur)
- Dadurch: kein **unnötiger** Informationsverlust



# Symbolisches Rechnen und Auswerten



auswerten:



# Interprozedurale kontextsensitive Wertanalyse

---

Alles wie bei kontextinsensitiver interprozeduraler Wertanalyse nur:

- ersetze  $\phi$ -Funktionen durch  $\chi$ -Funktionen
- Analysiere Werte am Anfang und Ende von Aufrufen mit Unterscheidung der Kontexte
- Fixpunktiteration bis keine Vereinfachungen der  $\chi$ -Terme mehr erkennbar
- u.U. Auswerten der  $\chi$ -Terme, wenn nicht genügend Speicher vorhanden



# Beispiel

---

```
void p(void) { int x=0; x=r(x); x=q(x); x=r(x) };
int q(int x) { if (x==1) then x=s(x) else x=t(x); return(x) };
int r(int x) { if (x==1) then x=s(x); return(x) };
int s(int x) { if (x==0) then x=r(x); return(x) };
int t(int x) { return(x+1) }
```

---

p:  $x_1 := 0, x_{callE\ r_1} := x_3$   
r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$   
q:  $x_4 := \chi_{\{x_3=1\}}(x_6, x_7)$   
s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
t:  $x_7 := x_3 + 1$



## Nach 1. Aufruf von $r()$ - Aufrufkontext einsetzen

---

p:  $x_1 := 0, x_{callE\ r_1} := x_3$   
r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$   
q:  $x_4 := \chi_{\{x_3=1\}}(x_6, x_7)$   
s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
t:  $x_7 := x_3 + 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := \chi_{\{\chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5)=1\}}(x_6, \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5))$   
r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$   
q:  $x_4 := \chi_{\{x_3=1\}}(x_6, x_7)$   
s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
t:  $x_7 := x_3 + 1$

(Mit **Rot** werden Änderungen markiert, unabhängig davon, zu welcher Prozedur sie gehören. Ist nichts **Rot** markiert, hat sich auch nichts geändert. Auf der oberen Hälfte wiederholt sich stets die untere Hälfte der vorangegangenen Folie (mit minimalen Vereinfachungen), damit man eine bessere Übersicht hat.)



## Wert weiterreichen an $q()$ und $t()$

---

p:  $x_1 := 0, x_{callE\ r_1} := \chi_{\{\chi_{\{p_1,p_2,s\}}(x_1,x_4,x_5)=1\}}(x_6, \chi_{\{p_1,p_2,s\}}(x_1, x_4, x_5))$

r:  $x_2 := \chi_{\{p_1,p_2,s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$

q:  $x_4 := \chi_{\{x_3=1\}}(x_6, x_7)$

s:  $x_5 := \chi_{\{r,q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := x_3 + 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := \chi_{\{x_1=1\}}(x_6, x_1) := 0$

r:  $x_2 := \chi_{\{p_1,p_2,s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$

q:  $x_4 := \chi_{\{x_{callE\ r_1}=1\}}(x_6, x_7)$

s:  $x_5 := \chi_{\{r,q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := x_{callE\ r_1} + 1$



# Vereinfachung mit weitergereichtem Wert

---

p:  $x_1 := 0, x_{callE\ r_1} := 0$   
r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$   
q:  $x_4 := \chi_{\{0=1\}}(x_6, x_7)$   
s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
t:  $x_7 := 0 + 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0$   
r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$   
q:  $x_4 := x_7$   
s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
t:  $x_7 := 1$





# Vereinfachung mit weitergereichtem Wert

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := x_4$   
r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$   
q:  $x_4 := 1$   
s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1$   
r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$   
q:  $x_4 := 1$   
s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
t:  $x_7 := 1$



## Nach 2. Aufruf von $r()$ - Aufrufkontext einsetzen

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_3$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$

q:  $x_4 := 1$

s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1,$

$x_{callE\ r_2} := \chi_{\{\chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5)=1\}}(x_6, \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5))$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$

q:  $x_4 := 1$

s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := 1$



# Vereinfachen des Aufrufkontextes

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := \chi_{\{x_4=1\}}(x_6, x_4)$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$

q:  $x_4 := 1$

s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$

q:  $x_4 := 1$

s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := 1$



## Nach Aufruf von $s()$ - Aufrufkontext einsetzen

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_{callE\ r_2} := x_6, x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$

q:  $x_4 := 1$

s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5),$

$x_{callE\ s} := \chi_{\{\chi_{\{r, q\}}(x_2, x_3)=0\}}(\chi_{\{r, q\}}(x_2, x_3), x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$

q:  $x_4 := 1$

s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := 1$



# Vereinfachen des Aufrufkontextes

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5),$

$x_{callE\ s} := \chi_{\{\chi_{\{r, q\}}(x_2, x_3)=0\}}(\chi_{\{r, q\}}(x_2, x_3), x_5), x_3 := \chi_{\{x_2=1\}}(x_6, x_2)$

q:  $x_4 := 1$

s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$

$x_3 := \chi_{\{x_2=1\}}(x_{callE\ s}, x_2)$

q:  $x_4 := 1$

s:  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$

t:  $x_7 := 1$



## Wert weiterreichen in $r()$

---

**p:**  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
**r:**  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$   
 $x_3 := \chi_{\{x_2=1\}}(x_{callE\ s}, x_2)$   
**q:**  $x_4 := 1$   
**s:**  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
**t:**  $x_7 := 1$

---

**p:**  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
**r:**  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$   
 $x_3 := \chi_{\{x_2=1\}}(\chi_{\{x_2=0\}}(x_2, x_5), x_2)$   
**q:**  $x_4 := 1$   
**s:**  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
**t:**  $x_7 := 1$

# Vereinfachen

---

**p:**  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
**r:**  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$   
 $x_3 := \chi_{\{x_2=1\}}(\chi_{\{x_2=0\}}(x_2, x_5), x_2)$   
**q:**  $x_4 := 1$   
**s:**  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
**t:**  $x_7 := 1$

---

**p:**  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
**r:**  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$   
 $x_3 := \chi_{\{x_2=1\}}(x_5, x_2)$   
**q:**  $x_4 := 1$   
**s:**  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
**t:**  $x_7 := 1$



## $s()$ nur von $r()$ nicht von $q()$ aufgerufen - $\chi$ entfällt

---

**p:**  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
**r:**  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$   
 $x_3 := \chi_{\{x_2=1\}}(x_5, x_2)$   
**q:**  $x_4 := 1$   
**s:**  $x_5 := \chi_{\{r, q\}}(x_2, x_3), x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
**t:**  $x_7 := 1$

---

**p:**  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
**r:**  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_5), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$   
 $x_3 := \chi_{\{x_2=1\}}(x_5, x_2)$   
**q:**  $x_4 := 1$   
**s:**  $x_5 := x_2, x_6 := \chi_{\{x_5=0\}}(x_3, x_5)$   
**t:**  $x_7 := 1$



## Nach Aufruf von $r()$ - Aufrufkontext einsetzen

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$

$x_3 := \chi_{\{x_2=1\}}(x_5, x_2)$

q:  $x_4 := 1$

s:  $x_{callE\ r} := x_3, x_6 := \chi_{\{x_2=0\}}(x_3, x_2)$

t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$

$x_3 := \chi_{\{x_2=1\}}(x_5, x_2)$

q:  $x_4 := 1$

s:  $x_{callE\ r} := \chi_{\{\chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2)=1\}}(x_5, \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2)),$

$x_6 := \chi_{\{x_2=0\}}(x_3, x_2)$

t:  $x_7 := 1$



# Vereinfachen und Weiterreichen

---

**p:**  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
**r:**  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5),$   
 $x_3 := \chi_{\{x_2=1\}}(x_5, x_2)$   
**q:**  $x_4 := 1$   
**s:**  $x_{callE\ r} := \chi_{\{\chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2)=1\}}(x_5, \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2)),$   
 $x_6 := \chi_{\{x_2=0\}}(x_3, x_2)$   
**t:**  $x_7 := 1$

---

**p:**  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
**r:**  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5)$   
**q:**  $x_4 := 1$   
**s:**  $x_{callE\ r} := \chi_{\{x_2=1\}}(x_5, x_2), x_6 := \chi_{\{x_2=0\}}(x_{callE\ r}, x_2)$   
**t:**  $x_7 := 1$



# Einsetzen

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5)$   
q:  $x_4 := 1$   
s:  $x_{callE\ r} := \chi_{\{x_2=1\}}(x_5, x_2), x_6 := \chi_{\{x_2=0\}}(x_{callE\ r}, x_2)$   
t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$   
r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5)$   
q:  $x_4 := 1$   
s:  $x_{callE\ r} := \chi_{\{x_2=1\}}(x_5, x_2), x_6 := \chi_{\{x_2=0\}}(\chi_{\{x_2=1\}}(x_5, x_2), x_2)$   
t:  $x_7 := 1$



# Vereinfachen

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5)$

q:  $x_4 := 1$

s:  $x_{callE\ r} := \chi_{\{x_2=1\}}(x_5, x_2), x_6 := \chi_{\{x_2=0\}}(\chi_{\{x_2=1\}}(x_5, x_2), x_2)$

t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5)$

q:  $x_4 := 1$

s:  $x_{callE\ r} := \chi_{\{x_2=1\}}(x_5, x_2), x_6 := x_2$

t:  $x_7 := 1$



## Nach 2. Aufruf von $r()$ - Aufrufkontext einsetzen

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_6 := x_2$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5)$

q:  $x_4 := 1$

s:  $x_{callE\ r} := \chi_{\{x_2=1\}}(x_5, x_2)$

t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2)$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5)$

q:  $x_4 := 1$

s:  $x_{callE\ r} := \chi_{\{x_2=1\}}(x_5, x_2)$

t:  $x_7 := 1$



# Programm liefert immer 1

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2)$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5)$

q:  $x_4 := 1$

s:  $x_{callE\ r} := \chi_{\{x_2=1\}}(x_5, x_2)$

t:  $x_7 := 1$

---

p:  $x_1 := 0, x_{callE\ r_1} := 0, x_{callE\ q} := 1, x_{callE\ r_2} := x_4 := 1$

r:  $x_2 := \chi_{\{p_1, p_2, s\}}(x_1, x_4, x_2), x_{callE\ s} := \chi_{\{x_2=0\}}(x_2, x_5)$

q:  $x_4 := 1$

s:  $x_{callE\ r} := \chi_{\{x_2=1\}}(x_5, x_2)$

t:  $x_7 := 1$



# Spezialaufgaben

---

- Vermeidung von Polymorphie
  - Aufgabe: welche Aufrufe an den Sprungverteiler werden garantiert zu einer (monomorphen) Prozedur weitergeleitet
  - Identisch mit der Fragestellung, was ist die Typkennung von Objekten, in denen die Prozedur aufgerufen wird
- Unterscheidung von Objekten (Verweisanalyse)
  - Aufgabe: welche Objekte werden durch einen Zugriffspfad adressiert
  - Für Vermeidung von Polymorphie notwendig und umgekehrt
- Beide Probleme sind unentscheidbar!  
Wir suchen bestmögliche Abschätzungen.



# Geeignetes Namensschema

---

- Adressen sind Werte
- Gleiche Analysetechnik
- Gesucht: geeignete Abstraktion von Adressen
- Namensschemata dafür angeben
- Verband für Objektadressenwertanalyse
  - Werte  $\in \langle \text{Objekt}, \text{Merkmalsname}, \text{Index} \rangle$
  - Abstraktion der Werte  $\in \langle \mathfrak{P}(NS(\text{Objekt})), \text{Merkmalsname}, \text{Intervallanalyseverband} \rangle$
  - $\mathfrak{P}(NS(\text{Objekt}))$  - Potenzmengenverband der durch Namensschema unterscheidbaren Objekte
- Verband für Funktionsadressenwertanalyse
  - Werte  $\in \text{Funktionname}$  (statisch nach Typanalyse unterscheidbar)
  - Abstraktion der Werte  $\in \mathfrak{P}(\text{Funktionname})$  (Potenzmengenverband der unterschiedlichen Funktionsnamen)





# Analyse

---

- einfache Wertanalyse wie bislang
- Sind Adreßwertterme eindeutig
  - so ist Funktionsaufruf monomorph
  - Zugriffspfad eindeutig
- Sind Adreßwertterme zweier Zugriffspfade disjunkt, so werden unterschiedliche Objekte adressiert



# Inhalt – SSA

---

- Optimierungen auf SSA-Form
  - Operatorvereinfachung
  - Eliminierung gemeinsamer Teilausdrücke
  - Eliminierung partieller Redundanzen
- Speicher SSA
  - Haldenoperationen
  - Alias-Problematik
  - Points-To-Analyse
  - Namensschema
  - Iterative Analyse
- Globale kontextsensitive Wertanalyse
  - Interprozedurale Analyse
  - Nichtrealisierbare Pfade
  - $\chi$ -Terme
  - Symbolisches Rechnen mit  $\chi$ -Termen



# Zusammenfassung

---

- Globale, kontextsensitive Wertanalyse
- Interprozeduraler Datenfluß mit  $\Phi$ -Termen
- Repräsentation von Analyseinformationen mit  $\chi$ -Termen:
  - Effiziente Speicherung
  - Effizientes symbolisches Rechnen
  - Effiziente Auswertung

