

## Zurück zu CM im allgemeinen

Traditionell

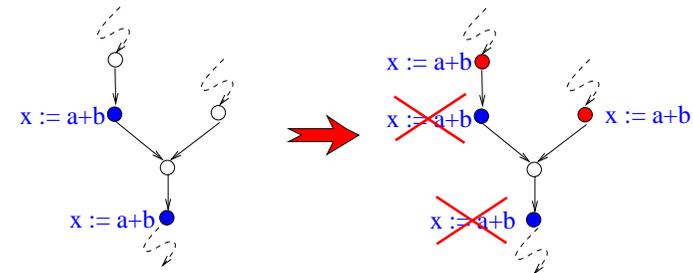
- Code (C) heißt Ausdrücke
- Motion (M) heißt vorziehen

Aber...

- CM ist mehr als vorziehen von Ausdrücken und PR(E)E!

## Zum Beispiel: Auch...

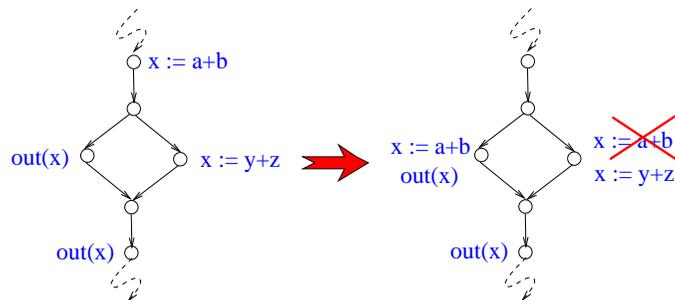
...Anweisungen sind Code.



- Hier heißt CM Elimination partiell redundanter Anweisungen (PRAE)

## Im Unterschied zu Ausdrücken, können...

...Anweisungen auch verzögert werden.



- CM heißt jetzt Elimination partiell toten Codes (PDCE)

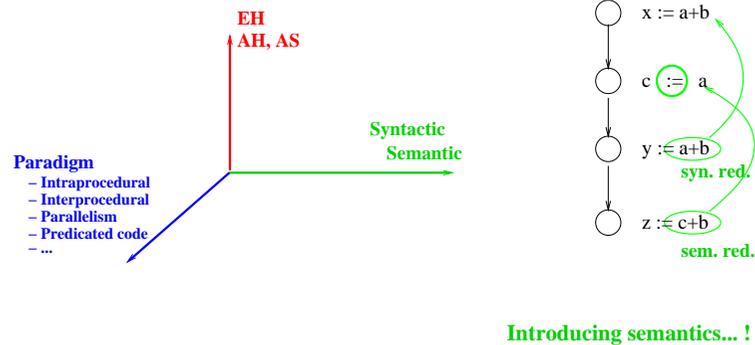
## Über den Entwurfsraum von CM-Algorithmem...

Allgemeiner...

- Code heißt Ausdrücke/Anweisungen
- Motion heißt vorziehen/verzögern

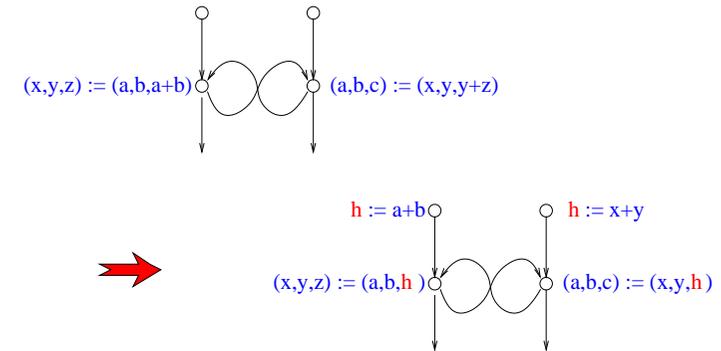
Code / Motion	Hoisting	Sinking
Expressions	EH	·/·
Assignments	AH	AS

## Weitere Verfeinerung des Entwurfsraums von CM-Algorithmen...



## Semantisches Code Motion...

erlaubt mächtigere Optimierungen!



(Beispiel von B. Steffen, TAPSOFT'87)

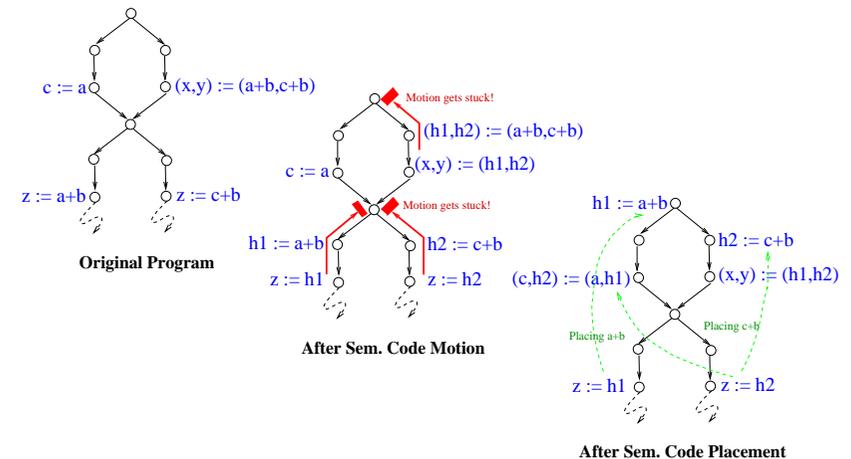
## Optimalitätsergebnisse sind sehr empfindlich!

Drei Beispiele sollen dies belegen...

- (I) Code motion vs. code placement
- (II) Abhängigkeiten elementarer Transformationen
- (III) Paradigmenabhängigkeiten

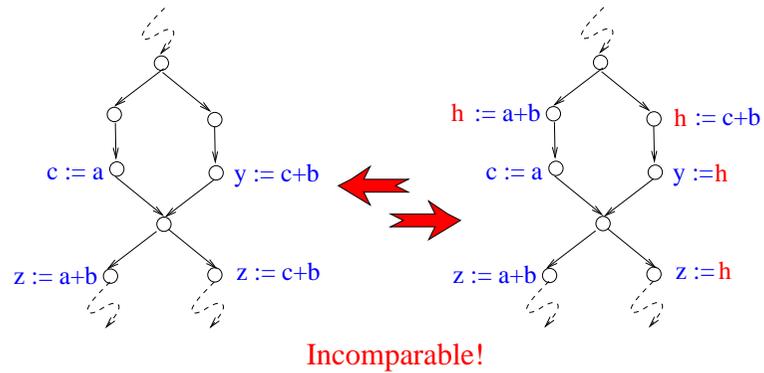
## (I) Code Motion vs. Code Placement

...sind keine Synonyme!



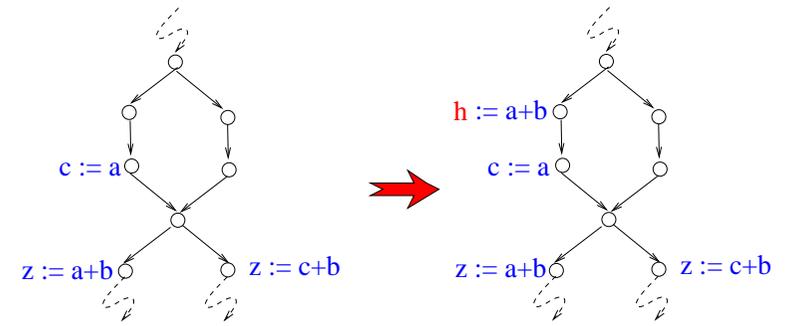
## Schlechter noch...

Optimalität ist verloren!

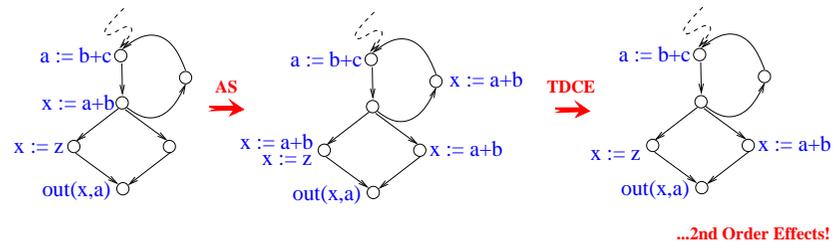


## Und sogar noch schlechter...

Performanz kann verloren gehen, wenn naiv angewandt!

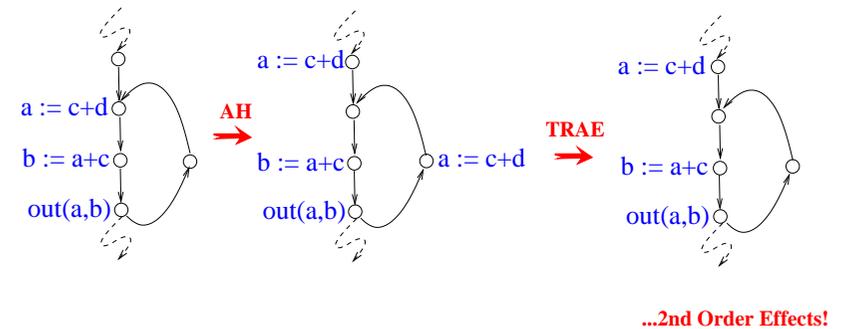


## (II) Abhängigkeiten von Transformationen



~> ...Partial Dead-Code Elimination (PDCE)

## Abhängigkeiten von Transformationen



~> ...Partially Redundant Assignment Elimination (PRAE)

## Konzeptuell

...können wir PREE, PRAE und PDCE wie folgt verstehen:

- PREE = EH ; TREE
- PRAE = (AH + TRAE)\*
- PDCE = (AS + TDCE)\*

## PRAE/PDCE – Optimalitätsergebnisse

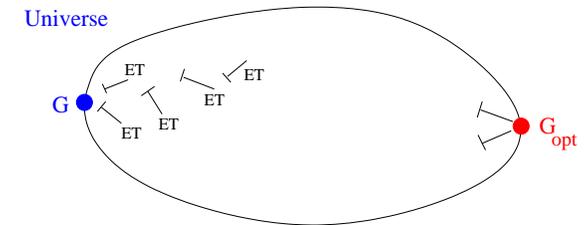
Ableitungsrelation  $\vdash \dots$

- PRAE...  $G \vdash_{AH,TRAE} G'$  (ET={AH,TRAE})
- PDCE...  $G \vdash_{AS,TDCE} G'$  (ET={AS,TDCE})

Wir können beweisen...

Optimalitätstheorem

Für PRAE und PDCE ist  $\vdash_{ET}$  konfluent und terminierend

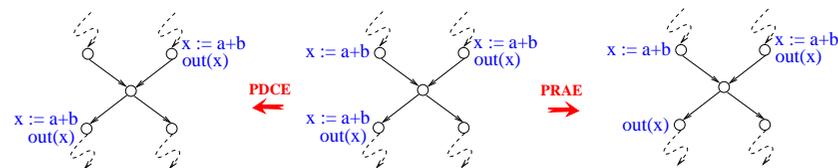


## Betrachte jetzt...

- Assignment Placement AP  
AP = (AH + TRAE + AS + TDCE)\*

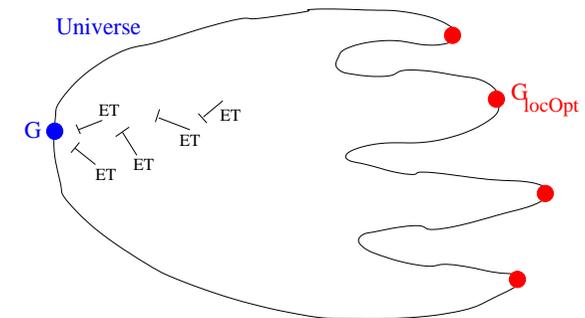
...sollte noch mächtiger sein!

In der Tat, aber...



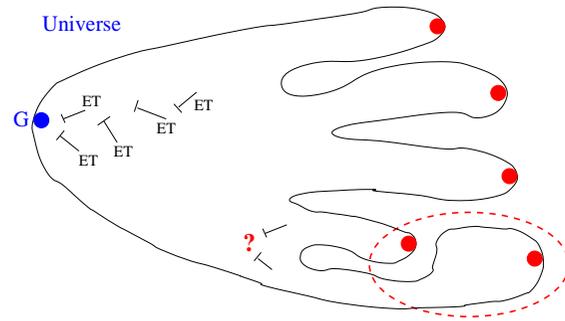
## Konfluenz...

...und folglich (globale) Optimalität ist verloren!

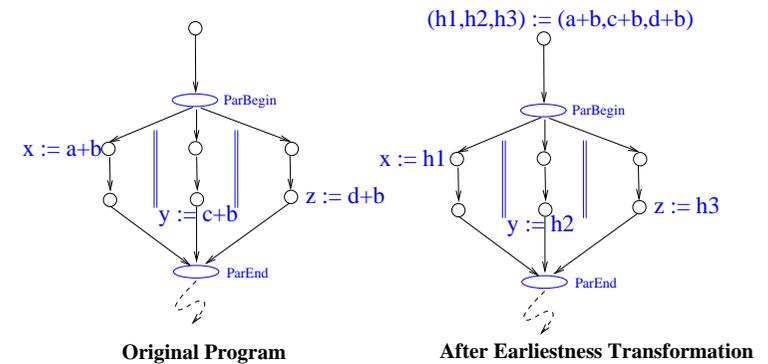


## Noch schlechter...

...es gibt Szenarien, in denen wir mit Universen wie dem folgenden enden können:



## (III) Paradigmenabhängigkeiten



...ein naiver Transfer der Transformationsstrategie führt hier zu einem im wesentlichen sequentiellen Programm!

## Interprozedurale Datenflussanalyse

In zwei Teilen...

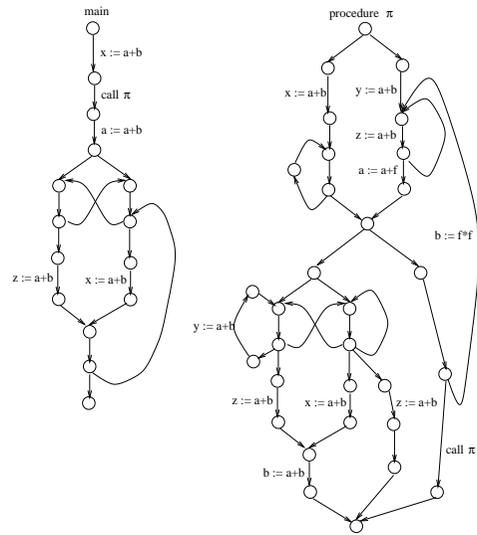
- Der Basisfall: Parameterlose Prozeduren, keine lokalen Variablen
- Erweiterungen
  - Wertparameter und lokale Variablen
  - Referenzparameter
  - Prozedurparameter

## Interprozedurale Datenflussanalyse

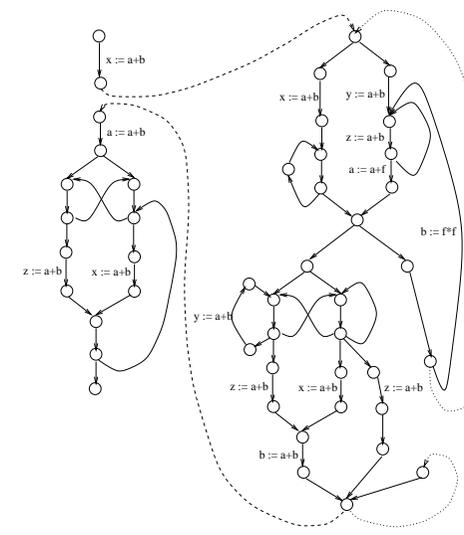
Darstellung von Programmen durch

- Flussgraphsysteme
- Interprozedurale Flussgraphen

# Interprozedurale Flussgraphensysteme



# Interprozedurale Flussgraphen



## (Lokale) abstrakte Semantik

Zwei Komponenten:

- Datenflussanalyseverband  $\hat{\mathcal{C}} = (\mathcal{C}, \sqcap, \sqcup, \sqsubseteq, \perp, \top)$
- Datenflussanalysefunktional  $\llbracket \cdot \rrbracket' : E^* \rightarrow (\mathcal{C} \rightarrow \mathcal{C})$

## Der IMOP-Ansatz

Die IMOP-Lösung:

$$\forall c_s \in \mathcal{C} \forall n \in N. IMOP_{c_s}(n) =_{df} \bigsqcap \{ \llbracket p \rrbracket'(c_s) \mid p \in IP[s, n] \}$$

wobei  $IP[s, n]$  die Menge der *interprozedural gültigen Pfade* von  $s$  nach  $n$  bezeichnet.

---

## Der *IMaxFP*-Ansatz 1(3)

Zweistufiger Ansatz

Stufe 1: Semantik von Prozeduren (2nd-Order Effects)

$\llbracket n \rrbracket =$

$$\begin{cases} Id_{\mathcal{C}} & \text{falls } n \in \{s_0, \dots, s_k\} \\ \sqcap \{ \llbracket (m, n) \rrbracket \circ \llbracket m \rrbracket \mid m \in \text{pred}_{\text{flowGraph}(n)}(n) \} & \text{sonst} \end{cases}$$

und

$$\llbracket e \rrbracket = \begin{cases} \llbracket e \rrbracket' & \text{falls } e \in E \setminus E_{\text{call}} \\ \llbracket \text{end}(\text{caller}(e)) \rrbracket & \text{sonst} \end{cases}$$

---

## Der *IMaxFP*-Ansatz 2(3)

Stufe 2: Die "eigentliche" interprozedurale DFA

Das *IMaxFP*-Gleichungssystem

$\mathbf{inf}(n) =$

$$\begin{cases} c_s & \text{falls } n = s_0 \\ \sqcap \{ \mathbf{inf}(\text{src}(e)) \mid e \in \text{caller}(\text{flowGraph}(n)) \} & \text{falls } n \in \{s_1, \dots, s_k\} \\ \sqcap \{ \llbracket (m, n) \rrbracket(\mathbf{inf}(m)) \mid m \in \text{pred}_{\text{flowGraph}(n)}(n) \} & \text{sonst} \end{cases}$$

---

## Der *IMaxFP*-Ansatz 3(3)

Die *IMaxFP*-Lösung

$$\forall c_s \in \mathcal{C} \forall n \in N. \text{IMaxFP}_{c_s}(n) =_{df} \mathbf{inf}_{c_s}^*(n)$$

---

## Hauptergebnisse 1(2)

Safety and Coincidence of the First Stage:

**Theorem** [2nd-Order]

For all  $e \in E_{\text{call}}$ , we have:

- $\llbracket e \rrbracket \sqsubseteq \sqcap \{ \llbracket p \rrbracket' \mid p \in \mathbf{CIP}[\text{src}(e), \text{dst}(e)] \}$ , if the data-flow functional  $\llbracket \rrbracket'$  is monotonic,
- $\llbracket e \rrbracket = \sqcap \{ \llbracket p \rrbracket' \mid p \in \mathbf{CIP}[\text{src}(e), \text{dst}(e)] \}$ , if the data-flow functional  $\llbracket \rrbracket'$  is distributive.

## Hauptergebnisse 2(2): Sicherheit und Koinzidenz

### Theorem [Interprocedural Safety]

The *IMaxFP*-solution is a lower approximation of the *IMOP*-solution, i.e.,

$$\forall c_s \in \mathcal{C} \forall n \in \mathbb{N}. \text{IMaxFP}_{c_s}(n) \sqsubseteq \text{IMOP}_{c_s}(n)$$

if the data-flow functional  $\llbracket \cdot \rrbracket'$  is monotonic.

### Theorem [Interprocedural Coincidence]

The *IMaxFP*-solution coincides with the *IMOP*-solution, i.e.,

$$\forall c_s \in \mathcal{C} \forall n \in \mathbb{N}. \text{IMaxFP}_{c_s}(n) = \text{IMOP}_{c_s}(n)$$

if the data-flow functional  $\llbracket \cdot \rrbracket'$  is distributive.

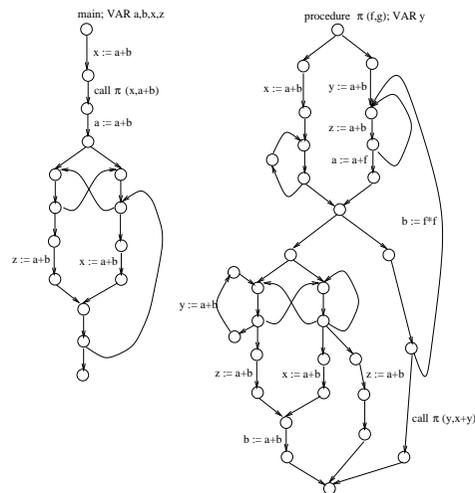
## Interprozedurale Erweiterungen

Zunächst...

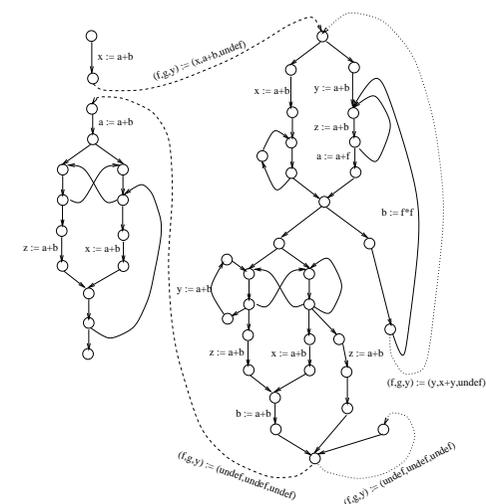
Hinzunahme von

- Wertparametern und lokalen Variablen

## Interprozedurale Flussgraphensysteme



## Interprozedurale Flussgraphen



---

## Für das Weitere

Siehe ausgeteilten Umdruck!