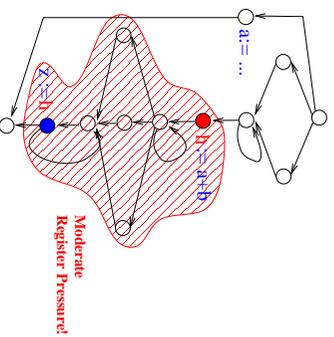


There is more than speed!

...z.B. Platz!



Der Weltmarkt für Mikroprozessoren 1999

Chip-Kategorie	Verkaufte Anzahl
Eingebettet 4-bit	2000 Millionen
Eingebettet 8-bit	4700 Millionen
Eingebettet 16-bit	700 Millionen
Eingebettet 32-bit	400 Millionen
DSP	600 Millionen
Desktop 32/64-bit	150 Millionen

... David Tennenhause (Intel Director of Research), Hauptvortrag auf dem 20th IEEE Real-Time Systems Symposium (RTSS'99), Phoenix, Arizona, Dezember 1999.

Der Weltmarkt für Mikroprozessoren 1999

Chip-Kategorie	Verkaufte Anzahl
Eingebettet 4-bit	2000 Millionen
Eingebettet 8-bit	4700 Millionen
Eingebettet 16-bit	700 Millionen
Eingebettet 32-bit	400 Millionen
DSP	600 Millionen
Desktop 32/64-bit	150 Millionen

~ 2%

... David Tennenhause (Intel Director of Research), Hauptvortrag auf dem 20th IEEE Real-Time Systems Symposium (RTSS'99), Phoenix, Arizona, Dezember 1999.

Man denke an...

... domänenspezifische Prozessoren eingesetzt in eingebetteten Systemen

- **Telekommunikation**
 - Mobiltelefone, Pager, ...
- **Heimelektronik**
 - MP3-Spieler, Kameras, Spielekonsolen, ...
- **Automobilbereich**
 - GPS-Navigation, Airbags, ...
- ...

Code für eingebettete Systeme

Anforderungen...

- Performance (oft Echtzeitanforderungen)
- Codegröße (system-on-chip, on-chip RAM/ROM)
- ...

Für eingebettete Systeme...
...Codegröße ist oft eine kritischere Größe als Geschwindigkeit!

Code für eingebettete Systeme (Fortsetzung)

Anforderungen (und wie sie häufig adressiert werden...):

- Assemblerprogrammierung
- Händische Postoptimierung

Schwächen...

- Fehlerträchtig
 - Verzögerte Marktreife/-eintritt
- ...die Probleme werden zunehmend größer mit wachsender Komplexität.

Generell beobachtet man...

...einen **Trend hin zu Hochsprachenprogrammierung (C/C++)**

Angesichts dieses Trends...

...wie unterstützen traditionelle Übersetzer- und Optimierungstechnologien das spezielle Anforderungsprofil von Code für eingebettete Systeme?



Leider nur wenig.

Unbestritten...

Traditionelle Optimierungen ...

- ...sind nahezu ausschließlich auf Performanceoptimierung getrimmt
- ...sind nicht codegrößen sensitiv und bieten i.a. keinerlei Kontrolle über ihren Einfluss auf die Codegröße

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 9

In besonderer Weise...

...gilt dies für Optimierungen, die auf Codeverschiebung beruhen

Dazu gehören insbesondere

- Partial redundancy elimination
- Partial dead-code elimination
- Partial redundant-assignment elimination
- Strength reduction
- ...

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 10

Erinnerung am Beispiel von PRE

PRE kann konzeptuell als zweistufiger Prozess gesehen werden...

1. Ausdrucksvorziehen
...vorziehen von Ausdrücken an "frühere" sichere Berechnungspunkte
2. Totale Redundanzeliminaton
...eliminieren von Berechnungen, die durch das Vorziehen total redundant geworden sind

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 11

Erinnerung am Beispiel von LCM

LCM kann konzeptuell als Ergebnis eines zweistufigen Prozesses gesehen werden...

1. Ausdrucksvorziehen (hoisting expressions)
...zu ihren "frühesten" sicheren Berechnungspunkten
2. Ausdrucksverzögern (sinking expressions)
...zu ihren "spätesten" sicheren und berechnungsoptimalen Berechnungspunkten

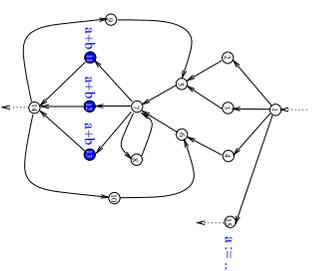
Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 12

Auf dem Weg zu codegrößen sensitiver PRE...

- Hintergrund: Klassische PRE
 - ~ Busy CM (BCM) / Lazy CM (LCM) (Knoop et al., PLDI'92)
 - Ausgezeichnet mit dem ACM SIGPLAN Most Influential PLDI Paper Award 2002 (for 1992)
 - Ausgewählt für "20 Years of the ACM SIGPLAN PLDI: A Selection" (60 Artikel aus ca. 600 Artikeln)
- Codegrößen sensitive PRE (Knoop et al., POPL'00)
 - ~ ...modulare Erweiterung von BCM/LCM
 - * Problemmodellierung und -lösung
 - * ...basiert auf graphtheoretischen Hilfsmitteln
 - * Hauptergebnisse
 - ...Korrektheit, Optimalität

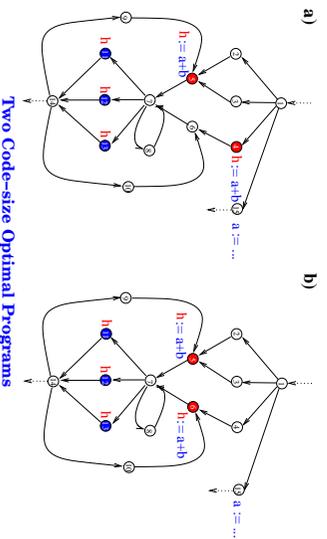
Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 13

Laufendes Beispiel



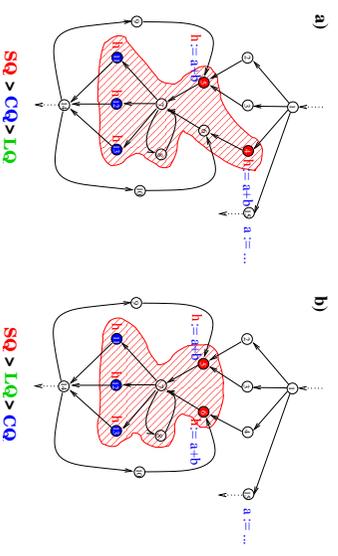
Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 14

Laufendes Beispiel (Fortsetzung)



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 15

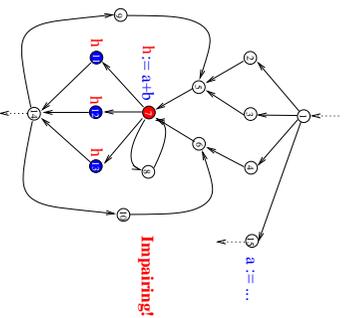
Laufendes Beispiel (Fortsetzung)



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 16

Laufendes Beispiel (Fortsetzung)

Beachte: Folgende Transformation ist unerwünscht!



Codegröbensensitive PRE

Das Problem

- Wir erhalten wir eine codegrößenminimale Platzierung der Berechnungen, d.h. eine Platzierung, die
 - zulässig (semantik- & performanzhaltend)
 - codegrößenminimal ist?

Lösung: Eine neue Sicht auf PRE

...betrachte PRE als ein Austauschproblem: Austauschen der ursprünglichen Berechnungen gegen neu eingeschaltete

Der Clou: Benutze Graphtheorie!

...führe das Austauschproblem auf die Berechnung sog. tight sets in bipartiten Graphen zurück basierend auf maximalen matchings!

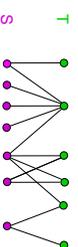
Wir verschieben, aber behalten im Gedächtnis

Wir müssen beantworten...

- Wo sind Initialisierungen vorzunehmen und wo sind ursprüngliche Berechnungen zu ersetzen?
- und beweisen
- Warum dies korrekt (semantikerhaltend) ist
- Wie sich dies auf die Codegröße auswirkt
- Warum dies "optimal" bezüglich einer vorgegebenen Priorisierung von Zielen ist?

Für jede dieser Fragen werden wir ein spezielles Theorem angeben, das uns die entsprechende Antwort liefert!

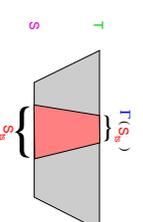
Bipartite Graphen



Tight Set

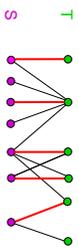
...eines bipartiten Graphen $(S \cup T, E)$: Teilmenge $S_{ts} \subseteq S$ mit

$$\forall S' \subseteq S: |S_{ts}| - |r(S_{ts})| \geq |S'| - |r(S')|$$



Zwei Varianten: (1) **Größte** Tight Sets (2) **Kleinste** Tight Sets

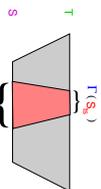
Bipartite Graphen



Tight Set

...eines bipartiten Graphen $(S \cup T, E)$: Teilmenge $S_{ts} \subseteq S$ mit

$$\forall S' \subseteq S: |S_{ts}| - |r(S_{ts})| \geq |S'| - |r(S')|$$



Zwei Varianten: (1) **Größte** Tight Sets (2) **Kleinste** Tight Sets

Offensichtlich

...können wir auf vorgefertigte Standardalgorithmen aus der Graphtheorie zurückgreifen, um

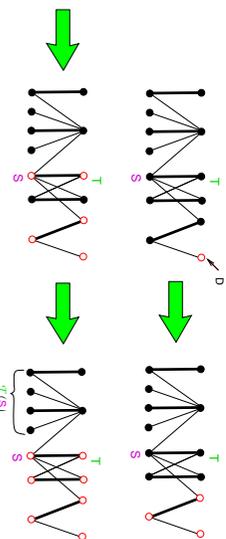
- Maximale Matchings und
- Tight sets zu berechnen.

Damit reduziert sich unser PRE-Problem auf...

...die Konstruktion des bipartiten Graphen, der das Problem modelliert!

Zur Berechnung größter/kleinster Tight Sets

...auf Grundlagede maximaler Matchings



Algorithmus LTS (Largest Tight Sets)

Eingabe: Bipartiter Graph $(S \cup T, E)$, maximales Matching M.

Ausgabe: Größte tight set $T_{LTS}(S) \subseteq S$.

```

SM := S; D := {t ∈ T | t is unmatched};
WHILE D ≠ ∅ DO
  choose some x ∈ D; D := D \ {x};
  IF x ∈ S
    THEN SM := SM \ {x};
    D := D ∪ {y | {x,y} ∈ M};
  ELSE D := D ∪ (r(x) ∩ SM);
FI
OD;
T_{LTS}(S) := SM
    
```

Algorithmus STS (Smallest Tight Sets)

Eingabe: Bipartiter Graph $(S \cup T, E)$, maximales Matching M .

Ausgabe: Kleinste tight set $T_{SmT}(S) \subseteq S$.

```

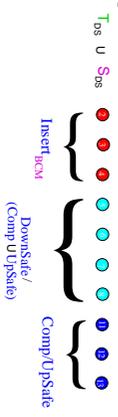
SM := ∅; A := {s ∈ S | s is unmatched};
WHILE A ≠ ∅ DO
  CHOOSE some x ∈ A; A := A \ {x};
  IF x ∈ S
    THEN SM := SM ∪ {x};
    A := A ∪ (Γ(x) \ SM)
  ELSE A := A ∪ {y | {x,y} ∈ M}
FI
OD;
TSmT(S) := SM
  
```

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

25

Modellierung des Austauschproblems

Die Menge der Knoten



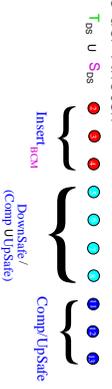
Die Menge der Kanten...

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

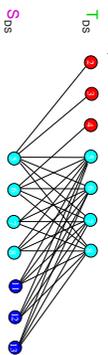
26

Modellierung des Austauschproblems

Die Menge der Knoten



Der bipartite Graph



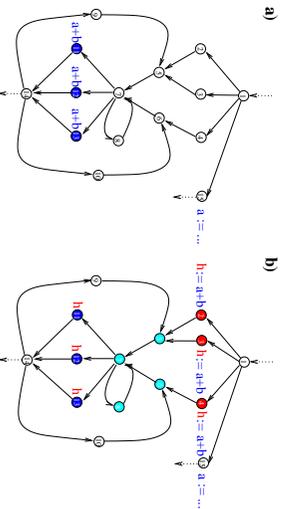
Die Menge der Kanten ... $\forall n \in S_{Ds} \forall m \in T_{Ds}$.

$$\{n, m\} \in E_{Ds} \iff \forall m \in \text{Closure}(\text{pred}(n))$$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

28

Die Menge der Knoten



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

27

DownSafety-Hüllen – Die zentrale Idee I(4)

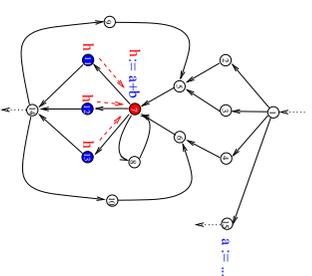
DownSafety-Hüllen

Für $n \in \text{DownSafe/UpSafe}$ ist die DownSafety-Hülle $\text{Closure}(n)$ die kleinste Menge von Knoten, so dass

1. $n \in \text{Closure}(n)$
2. $\forall m \in \text{Closure}(n) \setminus \text{Comp}_{\text{succ}}(m) \subseteq \text{Closure}(n)$
3. $\forall m \in \text{Closure}(n), \text{pred}(m) \cap \text{Closure}(n) \neq \emptyset \Rightarrow \text{pred}(m) \setminus \text{UpSafe} \subseteq \text{Closure}(n)$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

29

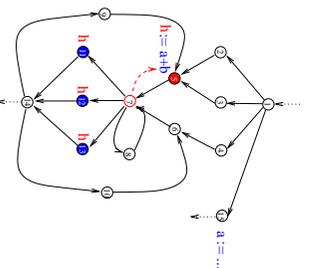


Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

30

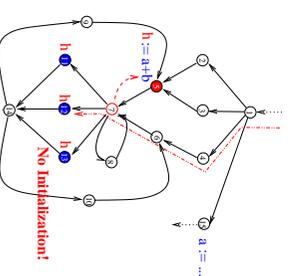
DownSafety-Hüllen – Die zentrale Idee 3(4)

DownSafety-Hüllen – Die zentrale Idee 2(4)



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

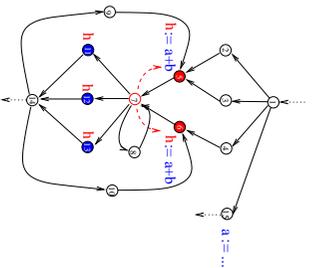
31



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

32

DownSafety-Hüllen – Die zentrale Idee 4(4)



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

33

DownSafety-Hüllen

DownSafety-Hüllen

Für $n \in \text{DownSafe}/\text{Upsafe}$ ist die DownSafety-Hülle $\text{Closure}(n)$ die kleinste Menge von Knoten, so dass

1. $n \in \text{Closure}(n)$
2. $\forall m \in \text{Closure}(n) \setminus \text{Comp}, \text{succ}(m) \subseteq \text{Closure}(n)$
3. $\forall m \in \text{Closure}(n), \text{pred}(m) \cap \text{Closure}(n) \neq \emptyset \Rightarrow \text{pred}(m) \setminus \text{Upsafe} \subseteq \text{Closure}(n)$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

34

DownSafety-Regionen

Einige Teilmengen von Knoten sind in besonderer Weise ausgezeichnet. Wir nennen diese Mengen DownSafety-Regionen...

- Eine Menge $\mathcal{R} \subseteq N$ von Knoten heißt DownSafety-Region gdw
 1. $\text{Comp} \setminus \text{Upsafe} \subseteq \mathcal{R} \subseteq \text{DownSafe} \setminus \text{Upsafe}$
 2. $\text{Closure}(\mathcal{R}) = \mathcal{R}$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

35

Die Schlüsselfragen

...bezüglich Korrektheit und Optimalität:

1. Wo Initialisierungen vornehmen, warum ist es korrekt?
 2. Wie ist der Effekt auf die Codegröße?
 3. Warum ist das Resultat optimal, d.h., codegrößenminimal?
- ...drei Theoreme werden jeweils eine dieser Fragen beantworten.

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

37

Hauptergebnisse / Zweite Frage

2. Wie ist der Effekt auf die Codegröße?

Intuitiv: Die Differenz aus eingesetzten und ersetzten Berechnungen...

Theorem 2 [DownSafety-Regionen: Platzgewinn]

Sei \mathcal{R} eine DownSafety-Region
mit $\text{Body}_{\mathcal{R}} =_{df} \mathcal{R} \setminus \text{EarliestFrontier}_{\mathcal{R}}$

Dann

- **Platzgewinn** aufgrund Einsetzens an **EarliestFrontier \mathcal{R}** :
- $$|\text{Comp} \setminus \text{Upsafe}| - |\text{EarliestFrontier}_{\mathcal{R}}| = |\text{Body}_{\mathcal{R}}| - |\Gamma(\text{Body}_{\mathcal{R}})| \quad_{df} = \text{defic}(\text{Body}_{\mathcal{R}})$$

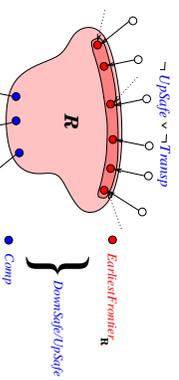
Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

39

Fundamental...

Initialisierungstheorem

Initialisierungen zulässiger PRE-Transformationen erfolgen stets am "frühesten Rand" von DownSafety-Regionen.



...charakterisiert erstmals alle semantikerhaltenden PRE-Transformationen.

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

36

Hauptergebnisse / Erste Frage

1. Wo Initialisierungen vornehmen, warum ist es korrekt?

Intuitiv: am frühesten Rand der von der tight set induzierten DS-Region...

Theorem 1 [Tight Sets: Initialisierungspunkte]

Sei $TS \subseteq S_{DS}$ eine tight set.

Dann ist $\mathcal{R}_{TS} =_{df} \Gamma(TS) \cup (\text{Comp} \setminus \text{Upsafe})$ eine DownSafety-Region mit $\text{Body}_{\mathcal{R}_{TS}} = TS$

Korrektheit

...unmittelbares Korollar aus Theorem 1 und dem Initialisierungstheorem

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

38

Hauptergebnisse / Dritte Frage

3. Warum ist das Resultat optimal, d.h., codegrößenminimal?

Aufgrund einer inhärenten Eigenschaft von tight sets (non-negative deficiency)...

Optimalitätstheorem [Die Transformation]

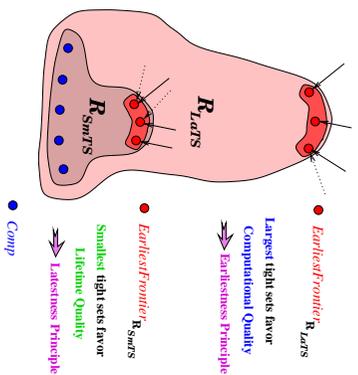
Sei $TS \subseteq S_{DS}$ eine tight set.

- **Initialisierungspunkte:**
 $\text{Insert}_{\text{SICM}} =_{df} \text{EarliestFrontier}_{\mathcal{R}_{TS}} = \mathcal{R}_{TS} \setminus TS$
- **Platzgewinn:**
 $\text{defic}(TS) =_{df} |TS| - |\Gamma(TS)| \geq 0 \text{ max.}$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

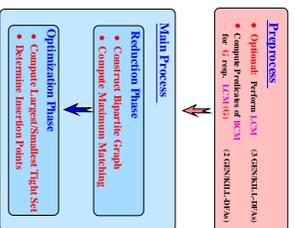
40

Größte vs. kleinste Tight Sets: Der Einfluss



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 41

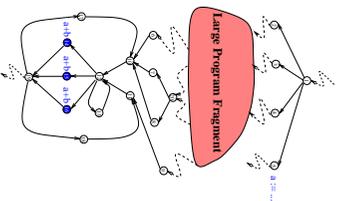
Codegrößensensitive PRE auf einen Blick 1(2)



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 43

Flexibilität (1)

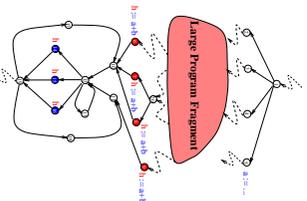
Das Ausgangsprogramm ...



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 45

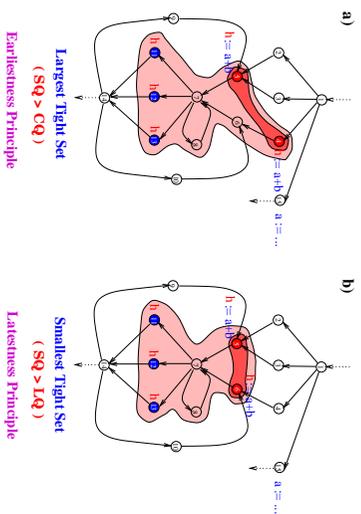
Flexibilität (3)

LCM ... A Computationally & Lifetime Opt. Program ($CQ > LQ$)



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 47

Effekt illustriert am laufenden Beispiel



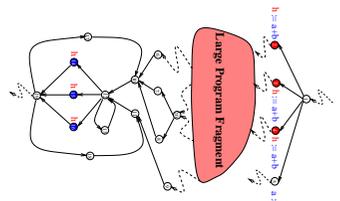
Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 42

...auf einen Blick 2(2)

Choice of Priority	Apply	To	Using	Yields	Additional Information Required
LQ			Not meaningful: 'The identity, i.e., G itself is optimal!'		
SQ				Subsumed by $SQ > CQ$ and $SQ > LQ$!	
CQ	BCM	G			$update(G)$, $bestset(G)$
$CQ > LQ$	LCM	G			$update(G)$, $bestset(G)$, $bestset(G)$
$SQ > CQ$	SpCM	G		Largest right set	$update(G)$, $bestset(G)$
$SQ > LQ$	SpCM	G		Smallest right set	$update(G)$, $bestset(G)$
$CQ > SQ$	SpCM	LCM(G)		Largest right set	$update(G)$, $bestset(G)$, $bestset(G)$, $bestset(LCM(G))$, $bestset(LCM(G))$
$CQ > SQ > LQ$	SpCM	LCM(G)		Smallest right set	$update(LCM(G))$, $bestset(LCM(G))$, $bestset(LCM(G))$, $bestset(LCM(G))$
$SQ > CQ > LQ$	SpCM	DL/SpCM/LTS(G)		Smallest right set	$update(G)$, $bestset(G)$, $bestset(SpCM/LTS(G))$, $bestset(DL/SpCM/LTS(G))$, $bestset(DL/SpCM/LTS(G))$

Flexibilität (2)

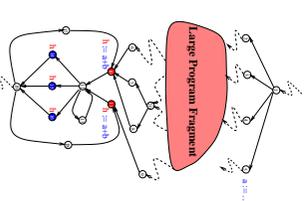
BCM ... Ein berechnungsoptimales Programm (CQ)



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 46

Flexibilität (4)

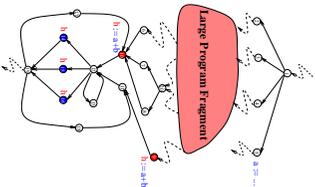
SpCM ... A Code-Size & Lifetime Optimal Program ($SQ > LQ$)



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007) 48

Flexibilität (5)

SpCM ... A Computationally and Lifetime Best Code-Size Optimal Program ($SQ > CQ > LQ$)

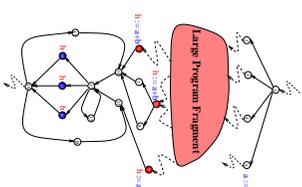


Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

49

Flexibilität (6)

SpCM ... A Code-Size and Lifetime Best Computationally Optimal Program ($CQ > SQ > LQ$)



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

50

Ein Rückblick

...auf die Entwicklung von PRE:

- 1958: ... *first glimpse of PRE*
 - ~ Ershov's work on *On Programming of Arithmetic Operations*.
- < 1979 ... Special Techniques
 - ~ Total Redundancy Elimination, Loop Invariant Code Motion
- 1979: ... *origin of contemporary PRE*
 - ~ Morel/Rennoise's seminal work on PRE
- 1992: ... LCM [Knoop et al., PLDI'92]
 - ~ ... first to achieve comp. optimality with minimum register pressure
 - ~ ... first to rigorously be proven correct and optimal

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

51

Ein Rückblick (fortgesetzt)

- 2000: ... *origin of code-size sensitive PRE* [Knoop et al., POPL 2000]
 - ~ ... first to allow prioritization of goals
 - ~ ... rigorously be proven correct and optimal
 - ~ ... first to bridge the gap between traditional compilation and compilation for embedded systems
- ca. since 1997: ... *a new strand of research on PRE*
 - ~ Speculative PRE: Gupta, Horspool, Sofia, Xue, Scholz, Knoop, ...
 - 2005: ... *another fresh look at PRE (as maximum flow problem)*
 - ~ Unifying PRE and Speculative PRE [Jingling Xue and J. Knoop]

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

52

Namen sind Nachrichten

Ein anderer Rückblick...

- < 1979 ... Special Techniques
 - ~ Total Redundancy Elimination, Loop Invariant Code Motion
- 1979 ... Partial Redundancy Elimination
 - ~ **Pioneering** ... Morel/Rennoise's bidirectional algorithm [1979]
 - ~ **Heuristic Improvements** ... Dhamdhere [1988, 1991], Drechsler/Stadel [1988], Sorkin [1989], Dhamdhere/Rosen/Zadeck [1992], ...
 - 1992 ... BCM & LCM [Knoop et al., PLDI'92]
 - ~ BCM ... first to achieve Computational Optimality: Earliestness Principle
 - ~ LCM ... first to achieve Comp. & Lifetime Optimality: Latestness Principle
- ... first to be purely unidirectional, however, not yet code-size sensitive.
- 2000/2004: Code-Size Sensitive PRE [Knoop et al., POPL 2000, LCTES 2004]

Warum lohnt es sich, PRE zu betrachten? (2)

Zu guter letzt, PRE ist...

- Wahlich klassisch ...blickt auf eine lange Geschichte zuruck
 - Morel, E. and Rennoise, C. *Global Optimization by Suppression of Partial Redundancies*. CACM 22 (2), 96 - 103, 1979.
 - Ershov, A. P. *On Programming of Arithmetic Operations*. CACM 1 (8), 3 - 6, 1958.

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

55

Warum lohnt es sich, PRE zu betrachten? (1)

Es ist...

- Relevant ... weit verbreitet in der Praxis
- Generell ... eine Familie von Optimierungen denn eine einzelne Optimierung
- Wohlverstanden ... bewiesen korrekt und *optimal*
- Herausfordernd ... konzeptuell einfach, aber weist eine Reihe kopffussaufgebender Phänomene auf

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

54

Exkurs: Die PRE-Formulierung von Morel/Rennoise

PRE ist untrennbar mit den Namen von E. Morel und C. Rennoise verknüpft. Ihr 1979 vorgestelltes Verfahren kann als "Ur-vater" aller CM-Verfahren angesehen werden und war bis in die 90er-Jahre hinein das "state of the art"-Verfahren für PRE.

Kennzeichnend für dieses Verfahren sind:

- 3 unidirektionale Bitvektoranalysen (AV, ANT, PAV)
- 1 bidirektionales Bitvektoranalyse (PP)

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

56

Die bahnbrechende PRE-Formulierung von Morel/Renvoise (1)

- Verfügbarkeit (Availability):

$$\text{AVIN}(n) = \begin{cases} \text{false} & \text{falls } n = s \\ \prod_{m \in \text{pred}(n)} \text{AVOUT}(m) & \text{sonst} \end{cases}$$

$$\text{AVOUT}(n) = \text{TRANSP}(n) * (\text{COMP}(n) + \text{AVIN}(n))$$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

57

Die bahnbrechende PRE-Formulierung von Morel/Renvoise (2)

- Vorziehbarkeit (Anticipability):

$$\text{ANTIN}(n) = \text{COMP}(n) + \text{TRANSP}(n) * \text{ANTOUT}(n)$$

$$\text{ANTOUT}(n) = \begin{cases} \text{false} & \text{falls } n = e \\ \prod_{m \in \text{succ}(n)} \text{ANTIN}(m) & \text{sonst} \end{cases}$$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

58

Die bahnbrechende PRE-Formulierung von Morel/Renvoise (3)

- Partielle Verfügbarkeit (Partial Availability):

$$\text{PAVIN}(n) = \begin{cases} \text{false} & \text{falls } n = s \\ \sum_{m \in \text{pred}(n)} \text{PAVOUT}(m) & \text{sonst} \end{cases}$$

$$\text{PAVOUT}(n) = \text{TRANSP}(n) * (\text{COMP}(n) + \text{PAVIN}(n))$$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

59

Die bahnbrechende PRE-Formulierung von Morel/Renvoise (4)

- Platzierung möglich (Placement Possible):

$$\text{PPIN}(n) = \begin{cases} \text{false} & \text{falls } n = s \\ \text{CONST}(n) * \left(\prod_{m \in \text{succ}(n)} (\text{PPOUT}(m) + \text{AVOUT}(m)) * (\text{COMP}(n) + \text{TRANSP}(n) * \text{PPOUT}(n)) \right) & \text{sonst} \end{cases}$$

$$\text{PPOUT}(n) = \begin{cases} \text{false} & \text{falls } n = e \\ \prod_{m \in \text{succ}(n)} \text{PPIN}(m) & \text{sonst} \end{cases}$$

wobei $\text{CONST}(n) =_{df} \text{ANTIN}(n) * (\text{PAVIN}(n) + \text{COMP}(n)) * \text{TRANSP}(n)$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

60

Die bahnbrechende PRE-Formulierung von Morel/Renvoise (5)

- Initialisierung:

$$\text{INSIN}(n) =_{df} \text{false}$$

$$\text{INSOUT}(n) =_{df} \text{PPOUT}(n) * \text{AVOUT}(n) * (-\text{PPIN}(n) + \text{TRANSP}(n))$$

- Ersetzung:
 $\text{REPLACE}(n) =_{df} \text{COMP}(n) * \text{PPIN}(n)$

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

61

Schwächen der PRE-Formulierung

... von Morel/Renvoise:

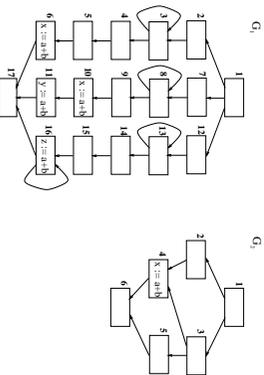
- Konzeptuell
 - Fehlende Berechnungsoptimalität
 - ~> nur aufgrund nicht gespaltener kritischer Kanten
 - Fehlende Lebenszeitoptimalität
 - ~> Heuristische Behandlung
 - Technisch
 - Bidirektionalität
 - ~> konzeptuell und berechnungsmäßig komplexer
- ... das Transformationsergebnis liegt (nicht vorhersagbar) zwischen denen von BCM- und LCM-Transformation.

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

62

Lehrreich

... folgende zwei Beispiele mithilfe des PRE-Verfahrens von Morel/Renvoise zu optimieren:



Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

63

Vorschau auf die weiteren Vorlesungstermine...

- Mo, 17.12./24.12./31.12.2007: Keine Vorlesung(en)! (Ferienzeit)
- Mo, 14.01.2008: Vorlesung von 16:15 Uhr bis 17:45 Uhr im Hörsaal 14, T-U-Hauptgebäude
- Mo, 21.01.2008: Vorlesung von 16:15 Uhr bis 17:45 Uhr im Hörsaal 14, T-U-Hauptgebäude

Analyse und Verifikation (WS 2007/2008) / 9. Teil (10.12.2007)

64

Einladung zum Kolloquiumsvortrag

Die Complang-Gruppe lädt ein zu folgendem Vortrag...

Automatic Verification of Combined Specifications

Prof. Dr. Ernst-Rüdiger Olderog
Carl v. Ossietzky Universität Oldenburg, Deutschland

ZEIT: Freitag, 13. Dezember 2007, 14:00 Uhr c.t.

ORT: TU Wien, Elektrotechnik, EI 5 Hochenegg-Hörsaal, Gußhausstr. 25-29 (Altbau), 2. Stock

MEHR INFO: <http://www.complang.tuwien.ac.at/~ralks/Oldero2007-12-13>

Alle Interessenten sind herzlich willkommen!