Karlsruher Institut für Technologie

Praktikum Compilerbau
Sitzung 8 – Optimierung: Konstantenfaltung

Prof. Dr.-Ing. Gregor Snelting
Andreas Zwinkau und Sebastian Buchwald

IPD Snelting, Lehrstuhl für Programmierparadigmen

Letzte Woche

- Was waren die Probleme?
- Hat soweit alles geklappt?

1. Letzte Woche
2. Motivation
3. Datenflussanalyse
4. Konstantenfaltung mit libFirm
5. Sonstiges

Wozu Optimierungen?

- Programme sollen besser werden
- Mögliche Optimierungsziele:
  - Laufzeit
  - Codegröße
  - Energieverbrauch

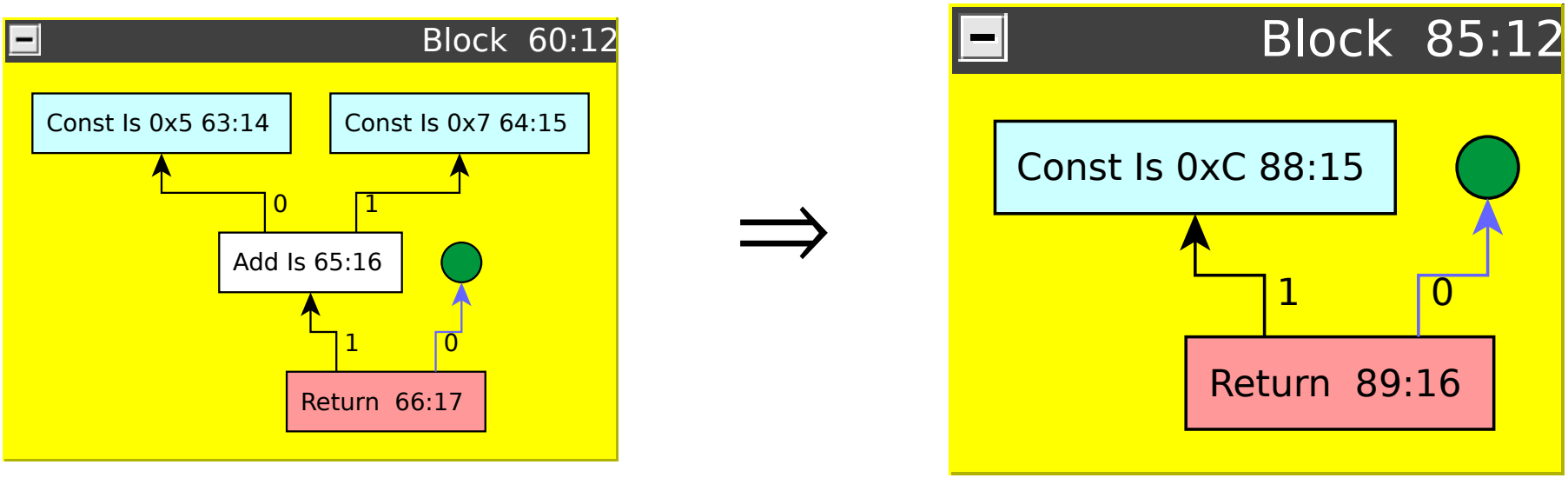
Anforderungen an Optimierungen

- Korrektheit: Optimierungen sollen Semantik erhalten
- Effizienz: Optimierung muss auch große Programme in angemessener Zeit verarbeiten können

Optimierungen im Praktikum

- Konstantenfaltung
- Lokale Optimierungen
- . . . (optional)

Konstantenfaltung



1. Letzte Woche
2. Motivation
3. Datenflussanalyse
4. Konstantenfaltung mit libFirm
5. Sonstiges

Wozu Datenflussanalyse?

Karlsruher Institut für Technologie

Datenflussanalyse ermöglicht es den Rückgabewert des folgenden Programms zu bestimmen:

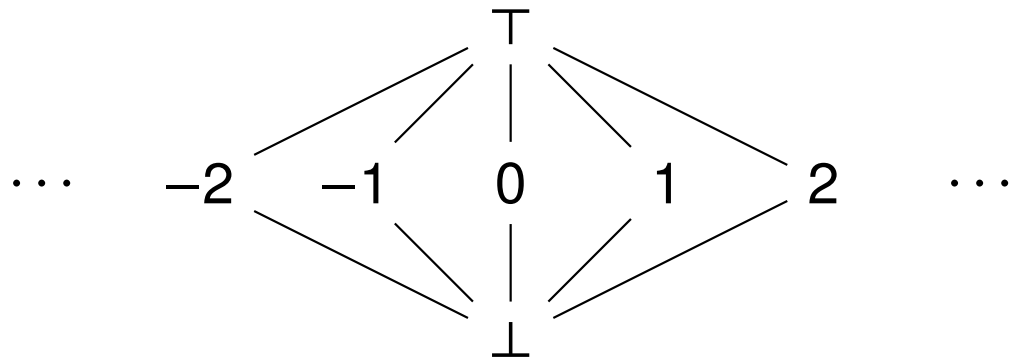
```
int optimizeMe() {
  int x = 1;

  while (...)
    x = 2 – x;

  return x;
}
```

Datenflussanalyse

- Verband $\mathcal{L} = (L; \leq, \sqcap, \sqcup)$ endlicher Höhe
  - $\top$ ist größtes Element des Verbandes
  - $\bot$ ist kleinstes Element des Verbandes
- Menge von monotonen Transferfunktionen $f : L \rightarrow L$
- Theorie: Es gibt ein i so dass $f^i(\bot) = f^{i+1}(\bot)$

Beispiel: Konstantenfaltung

- Verband besteht aus n-Tupeln über dem folgenden Verband

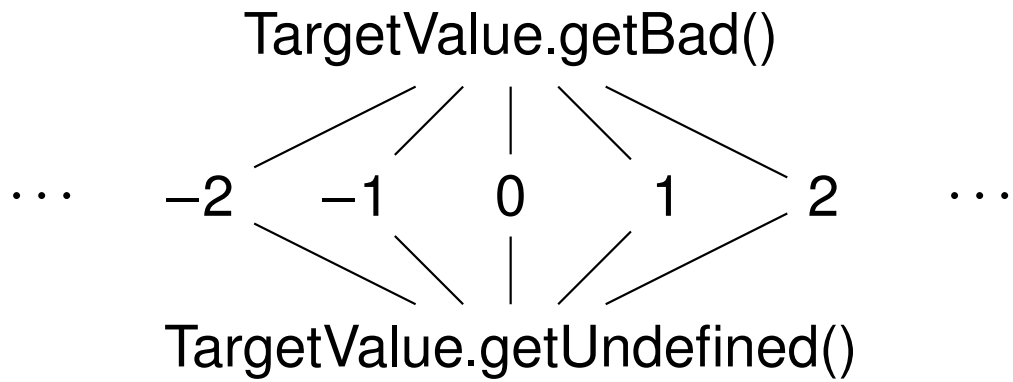


- Punktweise Anwendung von $\sqcup$ bei mehreren Vorgängern
- Transferfunktionen
  - $f_{x_2=\phi(x_0,x_1)}(l) = l[l_{x_2} \leftarrow l_{x_0} \sqcup l_{x_1}]$
  - $f_{z=x+y}(l) = \begin{cases} l[l_z \leftarrow \bot] & l_x = \bot \vee l_y = \bot \\ l[l_z \leftarrow c_0 + c_1] & l_x = c_0 \wedge l_y = c_1 \\ l[l_z \leftarrow \top] & \text{sonst} \end{cases}$
  - ...

1. Letzte Woche
2. Motivation
3. Datenflussanalyse
4. Konstantenfaltung mit libFirm
5. Sonstiges

Konstantenfaltung mit libFirm

- TargetValue als Verband



- SSA-Form vereinfacht die Datenflussanalyse
  - Zu jedem Wert muss (global) nur ein TargetValue gespeichert werden

Arbeitslisten-Algorithmus

- Arbeitsliste zur effizienten Implementierung verwenden
  - Initial mit allen Knoten füllen
  - Wenn sich die Analyse-Information an einem Knoten geändert hat: Alle Verwender des Knotens in die Arbeitsliste hängen

```
NodeCollector collector = new NodeCollector(worklist);
graph.walkTopological(collector);

while (!worklist.isEmpty()) {
   Node n = worklist.remove();
   n.accept(this);
}
```

Transformation des Graphen

```
for (Node node : latticeMap.keySet()) {
   TargetValue tv = latticeMap.get(node);

   if (tv.isConstant()) {
      Node constant = graph.newConst(tv);
      Graph.exchange(node, constant);
   }
}
```

1. Letzte Woche
2. Motivation
3. Datenflussanalyse
4. Konstantenfaltung mit libFirm
5. Sonstiges

Feedback! Fragen? Probleme?

Karlsruher Institut für Technologie

- Anmerkungen?
- Probleme?
- Fragen?