

Praktikum Compilerbau

Sitzung 7 – libFirm

Lehrstuhl für Programmierparadigmen
Universität Karlsruhe (TH)

3. Juni 2009

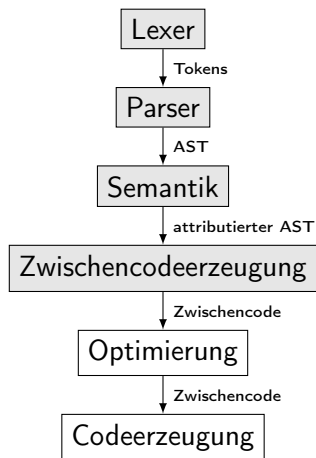
- 1 Letzte Woche
- 2 Firmgraph Optimierungen
- 3 Aufgabe
- 4 Sonstiges

Letzte Woche

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

- 1 Letzte Woche
- 2 Firmgraph Optimierungen**
- 3 Aufgabe
- 4 Sonstiges

Compilerphasen



Optimieren mit Firm

- Ersetzen von teureren Strukturen durch Billigere
 - Entfernen unnötiger Befehle
 - Gemeinsame Teilausdrücke
 - Konstantenfalten
 - Arithmetische Vereinfachung
 - ...
- Graphbasierter Zwischencode → Ablaufen des Graphen
- SSA Form → Def-Use enthalten
- Viele Optimierungen gehen einfacher (ohne Datenfluss Analyse)
- Tipp: Normalisieren als erster Schritt hilft oft

Beispiel: Konstantenfalten

```
public int foo3 () {  
    int x;  
    x = 2 + 3;  
    return x + x + 4;  
}
```



```
public int foo3 () {  
    return 14;  
}
```

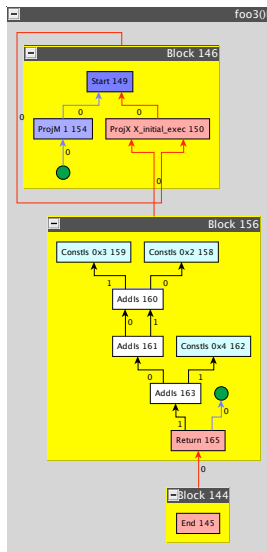
- Konstante Ausdrücke vorberechnen
- Variablen mit Konstanten Werten propagieren
- In Firm sehr einfach: keine Variablen vorhanden

Beispiel: Firmgraph

```

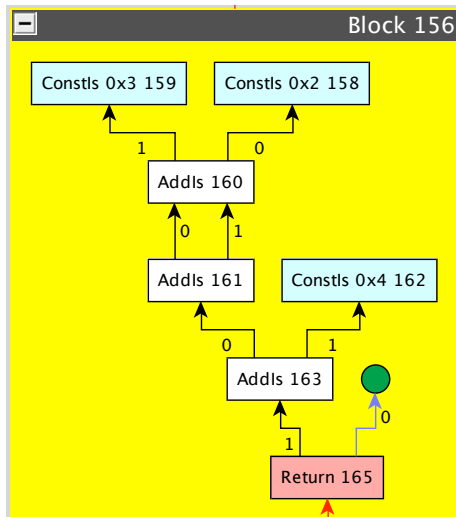
public int foo3() {
    int x;
    x = 2 + 3;
    return x + x + 4;
}

```



Beispiel: Firm Konstantenfalten

- Graph durchlaufen und konstante Ausdrücke suchen
- Wie durchlaufen? Tiefensuche, Breitensuche, Postorder, ...?



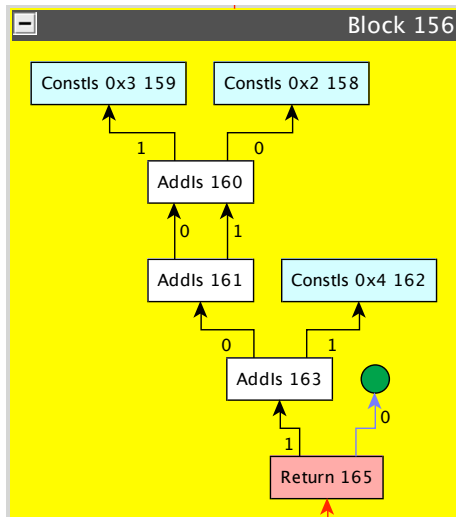
Tiefensuche

```
void preorder(Node node) {  
    act(node);  
  
    for (Node pred : node.preddecessors) {  
        preorder(pred);  
    }  
}
```

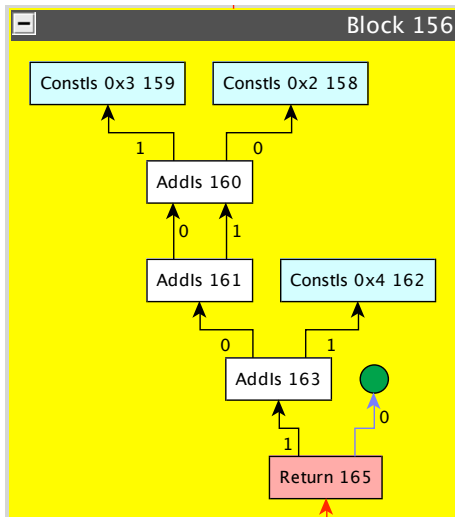


```
void postorder(Node node) {  
    for (Node pred : node.preddecessors) {  
        postorder(pred);  
    }  
  
    act(node);  
}
```

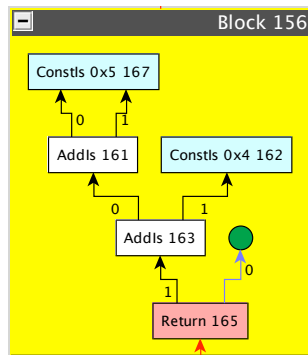
Beispiel: Tiefensuche



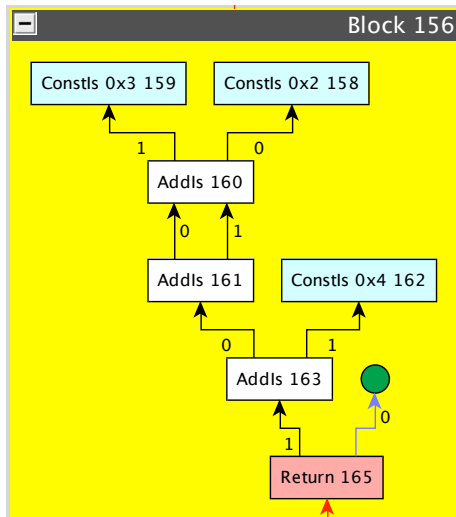
Beispiel: Tiefensuche



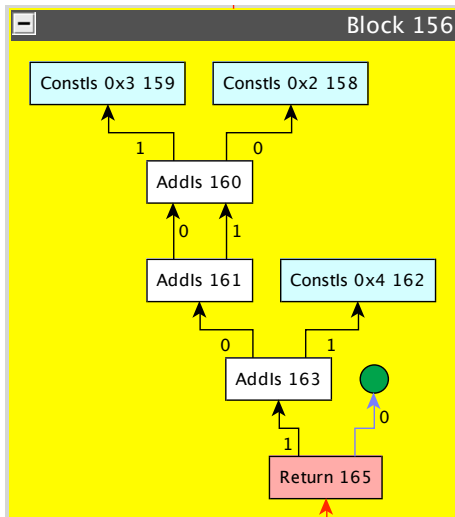
Preorder



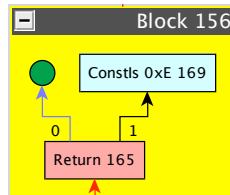
Beispiel: Tiefensuche



Beispiel: Tiefensuche



Postorder



Graph traversieren in Firm

- Preorder Tiefensuche im Firmgraph:
walk(NodeVisitor visitor)
- Postorder Tiefensuche im Firmgraph:
walkPostorder(NodeVisitor visitor)

```

public interface NodeVisitor {
    void visit(NoMem node);
    void visit(End node);
    ...

    static abstract class Default implements NodeVisitor {
        public abstract void defaultVisit(Node n);

        public void visit(NoMem node) {
            defaultVisit(node);
        }
    }
}

```

Beispiel: Konstantenfalten - Initialisierung

```
public class FoldConstants extends NodeVisitor.Default {  
  
    private final Graph graph;  
  
    public FoldConstants(Graph method) {  
        this.graph = method;  
    }  
  
    public void optimize() {  
        graph.walkPostorder(this);  
        //graph.walk(this);  
    }  
  
    public void defaultVisit(Node n) {  
        // Leave nodes untouched per default.  
    }  
  
}
```


Beispiel: Konstantenfalten - Add

```
public void visit(Add add) {  
    Node left = add.getLeft();  
    Node right = add.getRight();  
  
    if (left instanceof Const && right instanceof Const) {  
        Const c1 = (Const) left;  
        Const c2 = (Const) right;  
  
        TargetValue val1 = c1.getTarval();  
        TargetValue val2 = c2.getTarval();  
  
        TargetValue valSum = val1.add(val2);  
        Node sum = graph.newConst(valSum);  
  
        Graph.exchange(add, sum);  
    }  
}
```

Demo

Demo

- 1 Letzte Woche
- 2 Firmgraph Optimierungen
- 3 Aufgabe**
- 4 Sonstiges

Optimierungen

- Implementiere Konstantenfaltung.
- Entwirf eine sinnvolle Normalisierung und begründe sie.
- Implementiere die Normalisierung.

- 1 Letzte Woche
- 2 Firmgraph Optimierungen
- 3 Aufgabe
- 4 Sonstiges**

Feedback! Fragen? Probleme?

- Anmerkungen?
- Probleme?
- Fragen?