



Universität Karlsruhe (TH)

Lehrstuhl für Programmierparadigmen

Sprachtechnologie und Compiler II SS 2009

Dozent: Prof. Dr.-Ing. G. Snelting

Übungsleiter: Matthias Braun

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

snelting@ipd.info.uni-karlsruhe.de

braun@ipd.info.uni-karlsruhe.de

Übungsblatt 4

Ausgabe: 3.6.2009

Besprechung: 9.6.2009

Aufgabe 1: Datenflussanalyse

1.1 Reaching Definitions Analysis

Gegeben ist das folgende Programm in der While-Sprache:

```
[z := 0]1; while [x ≠ y]2 do ([z := z + 1]3; if [x > 4]4 then [x := x - y]5 else [y := y - x]6); [a := x]7
```

Auf welcher Menge operiert die Reaching Definitions Analysis?

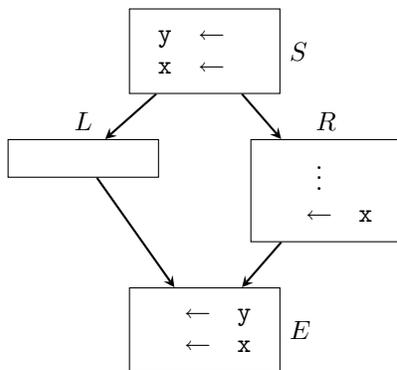
Führe eine Reaching Definitions Analysis für das Programm durch. Erstelle dafür zunächst eine Tabelle mit den Gen- und Kill-Mengen der einzelnen Blöcke und führe danach eine Fixpunktiteration durch.

Wie sehen Ergebnisse einer Reaching Definitions Analysis für Programme in SSA-Form aus?

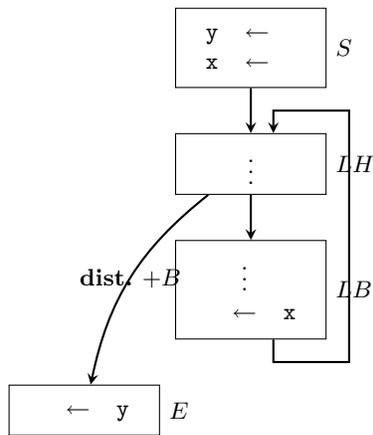
1.2 Next-Use Analyse

Sie wollen einen Algorithmus entwerfen der Programmvariablen abhängig von der Entfernung zu ihrer nächsten Verwendung auswählt. Um die Entfernung für allgemeine Ablaufgraphen zu berechnen sollten einige Regeln beachtet werden:

- Gibt es an einer Gabelung im Ablaufgraph mehrere nächste Verwendungen, so soll der kleinste Wert übernommen werden. Im folgenden Beispiel liegt die nächste Benutzung von x also vor der von y .



- Da Schleifen typischerweise erst einmal im Kreis laufen, bevor Sie verlassen werden, wird beim Verlassen der Schleife die Anzahl der Befehle innerhalb der Schleife addiert. Die nächste Benutzung von y innerhalb der Schleife ist also der eigentliche Next-Use Wert plus die Anzahl der Befehle in der Schleife (hier als B bezeichnet):



Entwerfen Sie eine geeignete Datenflußanalyse um Entfernungen bis zur nächsten Benutzung zu bestimmen!

- Auf welchen Mengen wird operiert?
- Sind wir an der kleinsten oder der größten Lösung interessiert?
- Wie stehen die Next-Use Analyse und Liveness Analyse miteinander in Beziehung?
- Führen Sie eine Next-Use Analyse auf dem folgenden Programm durch:

```

int foo(void)
{
    int a = 10;
    int b = rand();

    while(a < 100) {
        int c = a * 2;
        if (a > 80) {
            printf("%d\n", c);
        }
        a = a+2;
    }

    return b;
}
  
```