



Universität Karlsruhe (TH)

Lehrstuhl für Programmierparadigmen

Compiler II SS 2008

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 2

Ausgabe: 5.5.2008

Besprechung: 19.5.2008

Achtung: Am 12.5. ist Pfingstmontag, die Besprechung findet deshalb eine Woche später am 19.5. statt.

Aufgabe 1: Verbände

1.1 Rechenregeln

Beweisen sie die aus der Vorlesung bekannten Rechenregeln:

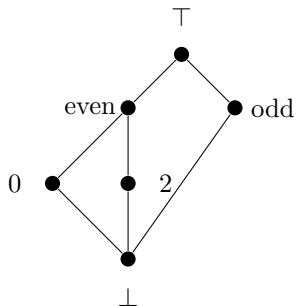
- $x \sqcup x = x$
- $x \sqcup y = y \sqcup x$
- $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
- $x \sqcap x = x$
- $x \sqcap y = y \sqcap x$
- $x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$

1.2 Distributivität

Verbände sind im allgemeinen nicht distributiv. Finden sie ein Gegenbeispiel für:

- $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$
- $x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap (x \sqcup z)$

1.3 Galois-Verbindungen



Die Abbildung zeigt den vollständigen Verband (M, \subseteq) . Gegeben sei weiterhin der Potenzmengenverband über den ganzen Zahlen \mathbb{Z} und die Funktion $\gamma : M \rightarrow \wp(\mathbb{Z})$ mit:

$$\begin{aligned} \gamma(\top) &= \mathbb{Z} & \gamma(\text{even}) &= \gamma(0) \cup \gamma(2) & \gamma(\text{odd}) &= \{z \in \mathbb{Z} \mid z \bmod 2 = 1\} \\ \gamma(0) &= \{z \in \mathbb{Z} \mid z \bmod 4 = 0\} & \gamma(2) &= \{z \in \mathbb{Z} \mid z \bmod 4 = 2\} & \gamma(\perp) &= \emptyset \end{aligned}$$

Zeigen oder widerlegen Sie, dass zwischen den gegebenen Verbänden eine Galois-Verbindung besteht.

Aufgabe 2: Lebendigkeitsanalyse

Eine Variable x heißt *lebendig* an einem Punkt p , wenn der Wert entlang eines Pfades beginnend mit p benutzt wird. Mit anderen Worten: “Welche Variablen werden vor einer Neuzuweisung noch benutzt?”

2.1 Datenflußgleichungen

Definieren Sie die entsprechenden Datenflußgleichungen.

2.2 CFG Beispiel

Betrachten Sie das Programm

```
1: c := 0;
2: while not a=0 do
3:   c:= c+b;
4:   a:= a-1;
   od;
5: return c
```

Wie sieht der Steuerflußgraph dieses Programms aus?

2.3 Lebendigkeitsanalyse

Führen Sie eine Lebendigkeitsanalyse am vorherigen Beispiel durch.

Aufgabe 3: *Programmabhängigkeitsgraph***3.1** *PDG*

Zeichnen sie den PDG für folgendes Programm:

```
1 int count = 0;
2 int k    = rand();
3 int k2   = rand();
4
5 if (k)
6     k = k + 5;
7
8 while (k < k2) {
9     count += 2;
10    if (rand())
11        count -= 5;
12    if (count % 13)
13        break;
14 }
15
16 printf("finished ");
```

3.2 *Kontrollabhängigkeit*

Zeigen oder widerlegen sie: Der Kontrollabhängigkeitsgraph ist ein Baum.