



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

Lösung zu Übungsblatt 6

Ausgabe: 30.6.2009

Besprechung: 7.7.2009

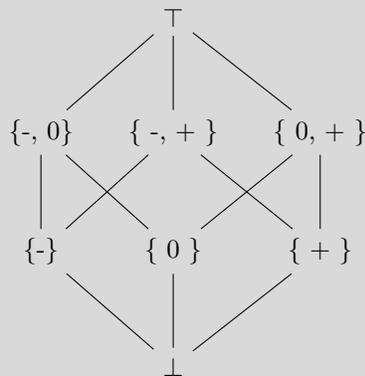
Aufgabe 1: Abstrakte Interpretation - Sign Analysis

In dieser Aufgabe soll eine einfache Analyse entworfen werden, die Werte von Variablen danach einteilt, ob sie Positiv, Negativ oder 0 sind.

1.1 Galois Verbindung

- Geben Sie einen passenden Verband für ihre Analyse an.

Lösung:



- Stellen Sie eine Galois-Verbindung von der Potenzmenge der ganzen Zahlen zu ihrem Verband auf.

Lösung:

Menge: $\mathcal{P}(\{-, 0, +\})$

$\gamma : \mathcal{P}(\{-, 0, +\}) \rightarrow \mathcal{P}(\mathbb{Z})$

$\gamma(x) = \{z \mid z < 0 \text{ if } - \in x; z = 0 \text{ if } 0 \in x; z > 0 \text{ if } + \in x\}$

$\alpha : \mathcal{P}(\mathbb{Z}) \rightarrow \mathcal{P}(\{-, 0, +\})$

$\alpha(x) = \{z \mid - \text{ if } \exists z \in x : z < 0; 0 \text{ if } 0 \in x; + \text{ if } \exists z \in x : z > 0\}$

1.2 Semantik

Stellen Sie eine (operationale) Semantik für alle Zeilen des folgenden Programms auf. Ein Zustand besteht dabei aus einem Paar von Programmpunkt (Zeile) und Wert der Variablen k .

```
1 while (k > 0) {
2   k = k - 2;
3   if (k == 0)
4     k = k + 7;
5 }
6 if (k < 0)
7   k = 0;
8 ;
```

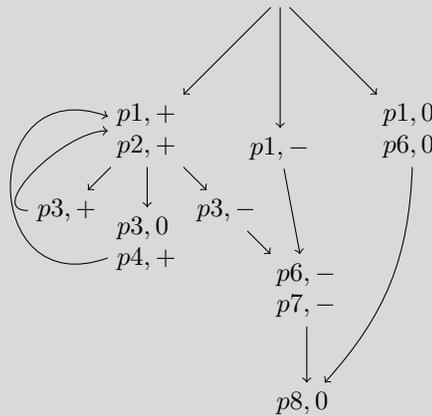
Lösung:

$$\begin{array}{cccccc} \frac{1, k \quad k > 0}{2, k} & \frac{1, k \quad k \leq 0}{6, k} & \frac{2, k}{3, (k-2)} & \frac{3, k \quad k = 0}{4, k} & \frac{3, k \quad k \neq 0}{1, k} & \frac{4, k}{1, (k+7)} \\ & & \frac{6, k \quad k < 0}{7, k} & \frac{6, k \quad k \geq 0}{8, k} & \frac{7, k}{9, 0} & \end{array}$$

1.3 Abstrakte Interpretation/Trace-Trees

Führen Sie eine abstrakte Interpretation durch und geben sie alle möglichen Trace-Trees an.

Lösung:



Aufgabe 2: Abstrakte Interpretation

- „A static analysis of a program is a sound, finite and approximate calculation of the program’s executions”. Was bedeuten die Begriffe **sound**, **finite** und **approximate** in diesem Zusammenhang?

Lösung:

- Eine Analyse wird als *sound* (fehlerfrei, gründlich) bezeichnet, wenn alle Möglichen Effekte und Eigenschaften eines Programms in der Analyse beachtet werden sofern diese Auswirkungen auf das Ergebnis der Analyse haben.
- Die Analyse ist *finite* (endlich) da nur mit endlich vielen Zwischenzuständen und endlichen Verbänden operiert wird. Ansonsten kann die Termination der Analyse nicht immer sichergestellt werden.
- Die Berechnung ist *approximate*, da sie den eigentlichen Wertebereich der Berechnung durch Rechnung auf einem größeren Wertebereich annähert.

- Wie stehen abstrakte Interpretation und Datenflussanalyse miteinander in Beziehung?

Lösung:

Abstrakte Interpretation ist eine allgemeinere Technik auf die praktisch alle bekannten Programmanalysen abgebildet werden können. Insbesondere für Datenflussanalysen die nur Aussagen über einzelne Werte machen (und nicht über Beziehungen zwischen Werte) ist es trivial die Datenflussanalyse als abstrakte Interpretation anzugeben.