

2. Übung zur Vorlesung „Prinzipien von Programmiersprachen“
 Wintersemester 2008/2009

Abgabe: 18. November 2008 in der Vorlesung

Aufgabe 4

(Präsenzaufgabe)

In dieser Aufgabe soll als Beispiel für ein Inferenzsystem der Gentzenkalkül der Aussagenlogik betrachtet werden. Eine *Sequenz* (Γ, Δ) ist ein Paar von endlichen (möglicherweise leeren) Mengen von aussagenlogischen Formeln. Diese Formeln sind aufgebaut aus aussagenlogischen Variablen und den Konnektoren \wedge , \vee , \Rightarrow und \neg . Für eine Sequenz $(\{A_1, \dots, A_n\}, \{B_1, \dots, B_m\})$ schreibt man auch $A_1, \dots, A_n \rightarrow B_1, \dots, B_m$. Dies wird als $A_1 \wedge \dots \wedge A_n \Rightarrow B_1 \vee \dots \vee B_m$ interpretiert. Ist Γ in (Γ, Δ) die leere Menge so wird die Sequenz auch mit $\rightarrow \Delta$ bezeichnet; ist Δ die leere Menge, so wird die Sequenz mit $\Gamma \rightarrow$ bezeichnet. Das Regelsystem des Gentzenkalküls besteht aus acht Inferenzschemata:

$$\frac{\Gamma, A, B \rightarrow \Delta}{\Gamma, A \wedge B \rightarrow \Delta} (\wedge : left)$$

$$\frac{\Gamma \rightarrow A, \Delta \quad \Gamma \rightarrow B, \Delta}{\Gamma \rightarrow A \wedge B, \Delta} (\wedge : right)$$

$$\frac{\Gamma, A \rightarrow \Delta \quad \Gamma, B \rightarrow \Delta}{\Gamma, A \vee B \rightarrow \Delta} (\vee : left)$$

$$\frac{\Gamma \rightarrow A, B, \Delta}{\Gamma \rightarrow A \vee B, \Delta} (\vee : right)$$

$$\frac{\Gamma \rightarrow A, \Delta \quad \Gamma, B \rightarrow \Delta}{\Gamma, A \Rightarrow B \rightarrow \Delta} (\Rightarrow : left)$$

$$\frac{\Gamma, A \rightarrow B, \Delta}{\Gamma \rightarrow A \Rightarrow B, \Delta} (\Rightarrow : right)$$

$$\frac{\Gamma \rightarrow A, \Delta}{\Gamma, \neg A \rightarrow \Delta} (\neg : left)$$

$$\frac{\Gamma, A \rightarrow \Delta}{\Gamma \rightarrow \neg A, \Delta} (\neg : right)$$

und einem Axiomenschema:

$$\Gamma \rightarrow \Delta \quad \text{falls } \Gamma \text{ und } \Delta \text{ mindestens eine gemeinsame Formel enthalten}$$

Geben Sie jeweils eine Ableitung für folgende Sequenzen an:

a) $P \wedge \neg Q, P \Rightarrow Q, T \Rightarrow R, P \wedge S \rightarrow T$

b) $\neg Q \Rightarrow P, Q \vee T, T \Rightarrow Q, R \Rightarrow S \rightarrow T \Rightarrow \neg R, S \vee Q$

Aufgabe 5

In dieser Aufgabe soll ein Beispiel für die strukturierte operationelle Semantik aus der Vorlesung (Kapitel 2.2) betrachtet werden. Sei π folgendes Programm:

```
n:= 3; j:= 1; u:= 1; v:= 1; w:= 1;
while j<n do
  w:= u+v; u:= v; v:= w; j:= j+1
```

Geben Sie eine terminierende Berechnung des Zustands $\langle \pi; \text{skip}, \sigma_{init} \rangle$ an. Die initiale Abbildung σ_{init} ordne jeder Variablen den Wert 0 zu. Die Addition wird wie üblich über den ganzen Zahlen interpretiert. Mehrere aufeinanderfolgende Zuweisungen dürfen zu einem Schritt zusammengefasst werden.

Aufgabe 6

Geben Sie eine Ableitung von

$$\{x \mapsto 5, y \mapsto 3\} \vdash x * (\text{let } z = 4 + y \text{ in } (z \text{ mod } 5)) : 10$$

in der natürlichen Semantik an.

Aufgabe 7

In der Vorlesung wurde ein operationelles Modell für imperative Sprachen vorgestellt. Der Zustand einer Programmabarbeitung wird hierbei mit Hilfe eines Paares $\langle E, M \rangle$ modelliert. Die erste Komponente E ist die Umgebung und die zweite Komponente M ist der Speicher.

- a) Sei $E = x : (\text{ref1}, \text{float}); x : (\text{ref42}, \text{int}); y : (\text{ref32}, \text{float})$.
Geben sie eine Ableitung von

$$E \stackrel{\text{lookup}}{\vdash} x : (\text{ref42}, \text{int})$$

an.

- b) Geben Sie eine Ableitung von

$$\langle x : (\text{ref42}, \text{int}); y : (\text{ref32}, \text{int}), \{\text{ref42} \mapsto 2, \text{ref32} \mapsto 5\} \rangle \stackrel{R}{\vdash} 3 * y + x : 17$$

an.