

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

Abgabe: 25. November 2008 in der Vorlesung

Aufgabe 8

(Präsenzaufgabe)

Gegeben seien die abstrakten Datentypen `Bool` (vgl. Vorlesung) und `Nat`:

```
datatype Nat
  0    :          -> Nat
  s    : Nat      -> Nat
  plus : Nat, Nat -> Nat
  leq  : Nat, Nat -> Bool
equations a, b : Nat
  0 plus b    = b
  s(a) plus b = s(a plus b)
  0 leq a     = true
  s(a) leq 0  = false
  s(a) leq s(b) = a leq b
end
```

- Bestimmen Sie die Konstruktoren von `Nat`.
- Rechnen Sie den Ausdruck `(s(0) plus s(0)) leq s(0)` aus. Verwenden Sie hierbei die in der Vorlesung angegebenen Gleichungsinferenzen.

Aufgabe 9

(Präsenzaufgabe)

Wir betrachten folgendes imperative Programm:

```
x : Integer;
procedure p1
begin
  x := x - 6;
  if x > 0 then p2 else print(x);
end;
procedure p2
  x : Integer;
begin
  x := 5; p1;
end;
begin
  x := 10; p1;
end
```

Welche Ausgabe liefert dieses Programm bei statischer bzw. bei dynamischer Bindung der Variablenbezeichner? Begründen Sie Ihre Antwort!

Aufgabe 10

Das operationelle Modell für imperative Sprachen wurde in der Vorlesung um Zeiger erweitert. Seien

$$E := a : \text{array}(\text{ref}32, [6], \text{int}); x : (\text{ref}62, \text{pointer}); z : (\text{ref}59, \text{pointer}); \\ p : \text{array}(\text{ref}98, [8], \text{pointer}); y : (\text{ref}75, \text{pointer})$$

und

$$M := \{ \text{ref}32 \mapsto 1, \text{ref}33 \mapsto 2, \text{ref}34 \mapsto 3, \text{ref}35 \mapsto 4, \text{ref}36 \mapsto 5, \text{ref}37 \mapsto 6, \\ \text{ref}59 \mapsto \text{ref}98, \text{ref}62 \mapsto \text{ref}35, \text{ref}75 \mapsto \text{ref}32, \text{ref}98 \mapsto \text{ref}32, \\ \text{ref}99 \mapsto \text{ref}75, \text{ref}100 \mapsto \text{null}, \text{ref}101 \mapsto \text{ref}62, \text{ref}102 \mapsto \text{ref}98, \\ \text{ref}103 \mapsto \text{ref}34, \text{ref}104 \mapsto \text{ref}75, \text{ref}105 \mapsto \text{ref}37 \}.$$

Geben Sie eine Ableitung von

$$\langle E, M \rangle \vdash^R *(*(z + 7)) * (*(* *(z + 4) + 1)) + *(*(p[3]) + 1) : 17$$

an.

Aufgabe 11

In der Vorlesung wurden Inferenzsysteme zur Herleitung der R- und L-Werte von Ausdrücken vorgestellt. Ein Ausdruck sei aufgebaut aus Namen, Zahlen und den Operatoren + und *.

- Geben Sie ein Inferenzsystem zur Herleitung des Typs eines Ausdruckes an. Es stehen die drei Typen `int`, `float` und `double` zur Verfügung. Die Anwendung von + und * sei dabei nur dann zulässig, falls beide Argumente vom gleichen Typ sind.
- Erweitern Sie das System aus Teil a) so, dass Typkonversionen implizit erlaubt sind, falls dies ohne Informationsverlust möglich ist. Zum Beispiel kann der Typ `int` in den Typ `float` umgewandelt werden, aber umgekehrt nicht (da z.B. `1.3` nicht ohne Informationsverlust in einen Wert vom Typ `int` konvertierbar ist).
- Geben Sie eine Inferenzregel an, die die Semantik einer Zuweisung mit Typumwandlung spezifiziert.

Aufgabe 12

In der Vorlesung wurden Inferenzregeln für die Deklaration, Erzeugung und den Zugriff auf Arrays vorgestellt. Geben Sie analog dazu Inferenzregeln für die Deklaration eines Verbundes und den Zugriff auf einzelne Werte (L- und R-Werte) an. Falls `class V` $\{\tau_1 x_1; \dots; \tau_n x_n\}$ der Verbund ist, so soll bei der Deklaration „`V x;`“ ein Eintrag $x : \text{record}(\tau_1 x_1, \dots, \tau_n x_n, l)$ in die Umgebung eingefügt werden. Es wird hierbei davon ausgegangen, dass jede Recordkomponente genau eine Speicherzelle benötigt und dass die Komponenten eines Records in aufeinanderfolgenden Zellen abgelegt werden.