



# Compilerbau

## 11. Übungsblatt

Abgabe: 2. Juli 2007, bis 16 Uhr in der Vorlesung oder Raum 02.07.59

### Aufgabe 32: Typinferenz

6 Punkte

Berechnen Sie den Typ der folgenden Ausdrücke der funktionalen Kernsprache der Vorlesung:

- letrec  $r = \text{fn } x \Rightarrow \text{if } x = 0 \text{ then } 0 \text{ else } x + r(x-1)$
- letrec  $f = \text{fn } t \Rightarrow \text{case } t \text{ of } [] \rightarrow [] \mid [(l,x,r)] \rightarrow [((f\ l),0,(f\ r))]$

### Aufgabe 33: Typinferenz

10 Punkte

Die Sprache der Vorlesung wird um benutzerdefinierte Datentypen erweitert. Zum Beispiel soll ein Datentyp zur Repräsentation binärer Bäume wie folgt definiert werden können:

```
datatype tree t = Leaf t | Node (tree t, tree t)
```

`tree int` ist laut dieser Definition ein Typ, welchem z.B. der Wert `Node(Leaf 1, Leaf 2)` angehört. In diesem Kontext bezeichnet man `tree` als Typ-Konstruktor, sowie `Leaf` und `Node` als Werte-Konstrukoren. Im Allgemeinen sieht eine Typ-Definition wie folgt aus:

```
datatype <Typ-Bezeichner> (t1, ..., tm) =  
  <Konstruktor-Bezeichner_1> [<Typ-Ausdruck_1> ]  
  | ...  
  | <Konstruktor-Bezeichner_n> [<Typ-Ausdruck_n> ]
```

wobei in den Typ-Ausdrücken die Typ-Variablen  $t_1, \dots, t_n$ , sowie der eingeführte Typ verwendet werden können. Der `case`-Ausdruck muss so erweitert werden, dass Pattern-Matching über Werte benutzerdefinierter Typen erlaubt ist. Die Auswertung des Ausdrucks

```
letrec count = fn tree =>  
  case tree of Leaf x -> 1  
  | Node(x,y) -> (count x)+(count y)  
in count (Node(Leaf 1, Leaf 2))
```

soll die Anzahl der Blätter der Baumes `Node(Leaf 1, Leaf 2)`, also 2, liefern. Um das Typ-System um benutzerdefinierte Datentypen anzureichern, ist folgendes Vorgehen ratsam:

- Erweitern Sie die Regeln zur Aufstellung der Term-Gleichungen zur Typinferenz.
- Erweitern Sie den Algorithmus  $\mathcal{W}$ .

### Aufgabe 34: Unifikation

4 Punkte

Unifizieren Sie die folgenden Termpaare.

- $p(f(a,Y),Z)$  und  $p(g(a,X),Z)$ ;
- $p(X,g(b,Y))$  und  $p(a,g(Z,f(X)))$ ;