



Einführung in die Informatik II

Univ.-Prof. Dr. Andrey Rybalchenko, M.Sc. Ruslán Ledesma Garza

WS 11/12
Übungsblatt 5
29.11.2011

Dieses Blatt behandelt Kapitel 3.4, 3.6, 3.13, 4.1 und 4.2 aus dem Buch zur Vorlesung. Lesen Sie diese Kapitel!

Aufgabe 3.24

Deklariieren Sie eine Prozedur `first : int → (int → bool) → int`, die zu x und p die kleinste Zahl $y \geq x$ mit $p\ y = \text{true}$ liefert. Verzichten Sie dabei vollständig auf Typangaben.

Aufgabe 3.25

Geben Sie die Typschemen an, mit denen die Bezeichner p und q des folgenden Programms typisiert werden.

```
fun p f (x,y) = f x y
fun q f g x = g (f x)
```

Aufgabe 3.26

Geben Sie Deklarationen an, die monomorph getypte Bezeichner wie folgt deklarieren:

- a) `int * unit * bool`
- b) `unit * (int * unit) * (real * unit)`
- c) `int → int`
- d) `int * bool → int`
- e) `int → real`
- f) `int → real → real`
- g) `(int→int)→bool`

Verzichten Sie dabei auf explizite Typangaben und verwenden Sie keine Operator- und Prozeduranwendungen. Hinweis: Für einige der Deklarationen ist die Verwendung eines Konditionals essentiell. Die Typregel für Konditionale verlangt, dass die Konsequenz und die Alternative den gleichen Typ haben (siehe § 2.6). Außerdem ist für einige der Deklarationen die Verwendung von Tupeln und Projektionen erforderlich, um Werte vergessen zu können, die nur zur Steuerung der Typinferenz konstruiert wurden.

Aufgabe 3.27 **

Im Zusammenhang mit fehlenden Typangaben kann die Verwendung von Projektionen problematisch sein. Beispielsweise kann Typinferenz das Programm `fun f x = #1x` nicht typisieren. Können Sie erklären, warum das so ist? **Hinweis:** Lesen Sie das Kapitel 2.8.

Aufgabe 3.36

Bitte lesen Sie §3.13. Deklarieren Sie mithilfe der Prozedur `iterup` eine Prozedur

- a) `power`, die zu x und n die Potenz x^n liefert.

- b) `fac`, die zu $n \geq 0$ die n -te Fakultät $n!$ liefert.
- c) `sum`, die zu f und n die Summe $0 + f 1 + \dots + f n$ liefert.
- d) `iter'`, die zu n , s und f dasselbe Ergebnis liefert wie `iter n s f`.

Aufgabe 3.37 Bitte lesen Sie §3.13. Deklarieren Sie mithilfe der Prozedur `iter` eine Prozedur

- a) `iterup'`, die zu m , n , s und f dasselbe Ergebnis wie `iterup m n s f` liefert.
- b) `iterdn'`, die zu n , m , s und f dasselbe Ergebnis wie `iterdn n m s f` liefert.

Aufgabe 4.1 Geben Sie einen Ausdruck an, der die Liste `[7, 2, 4]` klammerfrei mit `Cons` und `nil` beschreibt. Geben Sie die Baumdarstellung ihres Ausdrucks an. Unterscheidet sich die Baumdarstellung des Ausdrucks von der Baumdarstellung der Liste?

Aufgabe 4.2 Betrachten Sie den Ausdruck `1::2::nil@3::4::nil`.

- a) Geben Sie die Baumdarstellung des Ausdrucks an.
- b) Geben Sie die Baumdarstellung der beschriebenen Liste an.
- c) Geben Sie die beschriebene Liste mit “[...]” an.

Aufgabe 4.3 Macht es für die dargestellten Listen einen Unterschied, wie die folgenden Ausdrücke geklammert sind?

- a) `(e1::e2)@e3` oder `e1::(e2@e3)`.
- b) `(e1@e2)@e3` oder `e1@(e2@e3)`.
- c) `(e1::e2)::e3` oder `e1::(e2::e3)`.

Aufgabe 4.4 (Enum) Schreiben Sie mithilfe der Prozedur `iterdn` (§3.13) eine Prozedur `enum:int→int→int list`, die zu zwei Zahlen $m \leq n$ die Liste `[m, ..., n]` liefert. Beispielsweise soll `enum 3 6 = [3, 4, 5, 6]` gelten. Für $m > n$ soll `enum` die leere Liste liefern.