



Einführung in die Informatik II

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

Dieses Blatt behandelt Kapitel 2.7.1 - 3.3 aus dem Buch zur Vorlesung. Lesen Sie diese Kapitel!

Aufgabe 2.4

Welche Bindungen berechnet das folgende Programm? Geben Sie außerdem den Auswertungsbaum in der Umgebung [] für dieses Programm an.

Hinweis: Lesen Sie Kapitel 2.7.1 - 2.7.4 und das Skript von der Homepage um diese Aufgabe zu lösen.

```
fun f (x:bool) : int = if x then 1 else 0
val x = 5*7
fun g (z:int) : bool = f(z<x)<x
val x = g 5
```

Aufgabe 2.6

 Betrachten Sie das folgende Programm:

```
val x = 3+2
fun f (y:int) : int = x+y
fun g (y:int) : int = if y<x then 0 else y+g(y-1)
```

- Geben Sie die Umgebung an, die die Ausführung des Programms in der Umgebung [] liefert.
- Geben Sie die Umgebung an, in der der Rumpf der Prozedur f bei der Ausführung des Aufrufs f 3 mit Umgebung V = [x:=4] ausgeführt wird.
- Geben Sie die Umgebung an, in der der Rumpf der Prozedur g bei der Ausführung des Aufrufs g 13 mit Umgebung V = [y:=5,x:=3] ausgeführt wird.
- Geben Sie den Auswertungsbaum in der Umgebung [] für das Programm an. **Hinweis:** Lesen Sie das Skript von der Homepage um diese Aufgabe zu lösen.

Aufgabe 2.7* Geben Sie einen geschlossenen Ausdruck (d. h., keine Deklaration) an, der eine Prozedur $\text{int} \rightarrow \text{int}$ beschreibt, die zu x das Ergebnis x^2 liefert. Geben Sie die Tripeldarstellung der durch Ihren Ausdruck beschriebenen Prozedur an. Hinweis: Verwenden Sie einen Let-Ausdruck.

Aufgabe 3.1 Deklarieren Sie eine Prozedur $\text{mul}:\text{int}\rightarrow\text{int}\rightarrow\text{int}\rightarrow\text{int}$, die das Produkt dreier Zahlen liefert. Deklarieren Sie mul auf 3 Arten: Mit einer kaskadierten Prozedurdeklaration, mit einer Prozedurdeklaration und zwei Abstraktionen, und mit einer Deklaration mit val und drei Abstraktionen.

Aufgabe sec3.1.b Geben Sie der Typ von f im fun f g h x = g (h x) an.

Aufgabe 3.2 Deklarieren Sie zwei Prozeduren, die die Operation div (ganzahlige Division) kartesisch und kaskadiert darstellen.

Aufgabe 3.3 Geben Sie die Baumdarstellung des Ausdrucks

```
mul x y + mul x (y + 2) * 5
```

an. Überprüfen Sie die Richtigkeit Ihrer Darstellung mit einem Interpreter.

Aufgabe 3.5 Geben Sie zu den folgenden Abstraktionen semantisch äquivalente Ausdrücke an, die ohne die Verwendung von Abstraktionen gebildet sind.

a) `fn (x : int) => x*x`

b) `fn (x : int) => fn (y : int) => x+y`

Hilfe: Verwenden Sie Let-Ausdrücke und Prozedurdeklarationen.

Aufgabe 3.6 Geben Sie die Tripeldarstellung der Prozedur an, zu der der folgende Ausdruck ausgewertet:

```
(fn (x:int) => fn (b:bool) => if b then x else 7) (2+3)
```

Aufgabe 3.7 Geben Sie die Tripeldarstellung der Prozedur an, zu der der folgende Ausdruck ausgewertet:

```
let val a = 7
    fun f (x:int) = a + x
    fun g (x:int) (y:int) : int = g (f x) y
in
  g (f 5)
end
```

Aufgabe 3.8.old Deklarieren Sie eine Prozedur `power : int → int → int`, die zu x und $n \geq 0$ die Potenz x^n liefert, wie folgt:

a) Mit einer kaskadierten Deklaration.

b) Mit einer Deklaration mit `val` und Abstraktionen.

Aufgabe 3.8 Deklarieren Sie eine Prozedur `prod : (int→int)→int→int`, die für $n \geq 0$ die Gleichung $\text{prod } f \ n = 1 \cdot (f \ 1) \cdot \dots \cdot (f \ n)$ erfüllt. Deklarieren Sie außerdem mithilfe von `prod` eine Prozedur `fac : int→int`, die für $n \geq 0$ die Fakultät $n!$ berechnet (siehe Aufgabe 1.26 auf S. 22). Die Prozedur `fac` soll nicht rekursiv sein.

Aufgabe 3.9 Deklarieren Sie mithilfe der höherstufigen Prozedur `sum` eine Prozedur `sum' : (int→int)→int→int→int`, die für $k \geq 0$ die Gleichung $\text{sum}' \ f \ m \ k = 0 + f(m+1) + \dots + f(m+k)$ erfüllt. Die Prozedur `sum'` soll nicht rekursiv sein.

Aufgabe 3.10 Geben Sie die Baumdarstellungen der folgenden Typen an:

a) `(int→int)→int`

b) `int→(int*bool→int)→int`

c) `(int→bool)→(bool→real)→int→real`

Aufgabe 3.11 Geben Sie geschlossene Abstraktionen an, die die folgenden Typen haben:

- a) $(\text{int} * \text{int} \rightarrow \text{bool}) \rightarrow \text{int} \rightarrow \text{bool}$
- b) $(\text{int} \rightarrow \text{bool} \rightarrow \text{real}) \rightarrow \text{int} \rightarrow \text{bool} \rightarrow \text{real}$
- c) $(\text{int} \rightarrow \text{bool}) \rightarrow (\text{bool} \rightarrow \text{real}) \rightarrow \text{int} \rightarrow \text{real}$
- d) $(\text{int} \rightarrow \text{real}) \rightarrow (\text{bool} \rightarrow \text{int}) \rightarrow \text{int} * \text{bool} \rightarrow \text{real} * \text{int}$

Die Abstraktionen sollen nur mit Prozeduranwendungen, Tupeln und Bezeichnern gebildet werden. Konstanten und Operatoren sollen nicht verwendet werden.

Aufgabe 3.12 Schreiben Sie zwei Prozeduren

- a) `cas : (int*int→int)→int→int→int`
- b) `car : (int→int→int)→int*int→int`

sodass `cas` zur kartesischen Darstellung einer zweistelligen Operation die kaskadierte Darstellung und `car` zur kaskadierten Darstellung die kartesische Darstellung liefert. Erproben Sie `cas` und `car` mit Prozeduren, die das Maximum zweier Zahlen liefern:

```
fun maxCas (x:int) (y:int) = if x<y then y else x
fun maxCar (x:int, y:int) = if x<y then y else x
val maxCas' = cas maxCar
val maxCar' = car maxCas
```

Wenn Sie `cas` und `car` richtig geschrieben haben, verhält sich `maxCas'` genauso wie `maxCas` und `maxCar'` genauso wie `maxCar`. Hinweis: Die Aufgabe hat eine sehr einfache Lösung.

Aufgabe 3.13 Deklarieren Sie eine Prozedur `mul : int→int→int`, die das Produkt zweier Zahlen x und $n \geq 0$ gemäß der Gleichung

$$x \cdot n = 0 + \underbrace{x \cdots + x}_{n\text{-mal}}$$

durch Addieren berechnet. Die Prozedur `mul` soll mithilfe der Prozedur `iter` formuliert werden und nicht rekursiv sein.

Aufgabe 3.14 Geben Sie eine Abstraktion `e` an, sodass die Ausführung des Ausdrucks `first x e` für alle x divergiert.

Aufgabe 3.15 Deklarieren Sie mit `first` eine Prozedur `divi : int→int→int`, die zu $x \geq 0$ und $m > 0$ dasselbe Ergebnis liefert wie Division mit dem Operator `div`.

Hinweis: Für $x \geq 0$ und $m > 0$ liefert `div` die größte ganze Zahl k mit $k \cdot m \leq x$.

Aufgabe 3.16 Deklarieren Sie mit `iter` eine Prozedur `fac`, die zu $n \geq 0$ die n -te Fakultät $n!$ liefert.

Aufgabe 3.17 Deklarieren Sie mit `iter` eine Prozedur `sum`, die für $n \geq 0$ die Gleichung `sum n f = 0 + f 1 ... + f n` erfüllt.