

### Einführung in die Informatik II

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

Dieses Blatt behandelt Kapitel 2.5, 2.6 und 2.7 ohne 2.7.1 - 2.7.4 aus dem Buch zur Vorlesung. Lesen Sie diese Kapitel!

**Aufgabe sec2.6.b** Geben Sie die Typen der folgenden Ausdrücken an. Begründen Sie Ihre Antwort.

- a)  $1 + 2$
- b) `if 1 < 2 then "ok" else "not ok"`
- c) `#1("1.0", 2.0) = "2.0"`
- d) `fun f (x,y,z,t) =  
if x = y then z + 1 else if x > y then z else y + t;`

**Lösungsvorschlag sec2.6.b:**

- a)  $1 + 2 : \text{int}$
- b) `if 1 < 2 then "ok" else "not ok" : string`
- c) `#1("1.0", 2.0) = "2.0" : bool`
- d) `fun f (x,y,z,t) =  
if x = y then z + 1 else if x > y then z else y + t;`  
The type is  $\text{int} * \text{int} * \text{int} * \text{int} \rightarrow \text{int}$

**Aufgabe sec2.6.c** Typeregeln können die Wohlgetyptheit nachweisen, indem sie eine Typableitung bilden. Zum Beispiel, der Typ `int` des Ausdrucks `1+2` wird durch den folgenden Baum nachgewiesen.

$$\frac{\frac{}{\emptyset \vdash 1 : \text{int}} \quad \frac{}{\emptyset \vdash + : (\text{int} * \text{int}) \rightarrow \text{int}} \quad \frac{}{\emptyset \vdash 2 : \text{int}}}{\emptyset \vdash 1 + 2 : \text{int}}$$

Geben Sie Typableitungsbäume für die folgenden Ausdrücke und Programme.

- a) `(1.5, "3", 7)`
- b) `if 1 < 2 then "ok" else "not ok"`
- c) `let val x = 1 val y = 2.0 in x = 2 end`
- d) `val a = 1 fun f (x : int) = if x > 0 then x + f(x - a) else 0 val b = f 1`
- e) `fun f (x : int) = x+1`
- f) `fun o1 x y = let  
fun o2 z = x (y (z - 1) + 0) + 1  
in  
o2  
end`

**Lösungsvorschlag sec2.6.c:**

a) (1.5, "3", 7)

$$\frac{\frac{}{\emptyset \vdash 1.5 : \text{real}} \quad \frac{}{\emptyset \vdash "3" : \text{string}} \quad \frac{}{\emptyset \vdash 7 : \text{int}}}{\emptyset \vdash (1.5, "3", 7) : \text{real} * \text{string} * \text{int}}$$

b) if 1 < 2 then "ok" else "not ok" : string

$$\frac{\frac{\frac{}{\emptyset \vdash 1 : \text{int}}}{\emptyset \vdash < : ( \text{int} * \text{int} ) \rightarrow \text{bool}} \quad \frac{}{\emptyset \vdash 2 : \text{int}}}{\emptyset \vdash 1 < 2 : \text{bool}} \quad \frac{}{\emptyset \vdash "ok" : \text{string}} \quad \frac{}{\emptyset \vdash "not ok" : \text{string}}}{\emptyset \vdash \text{if } 1 < 2 \text{ then "ok" else "not ok" : string}}$$







e) `fun f (x : int) = x+1`

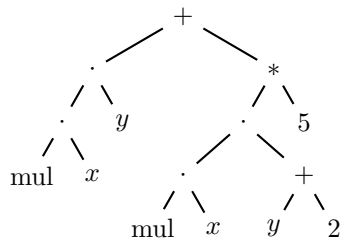
$$\frac{\frac{\frac{}{\{ f := \text{int} \rightarrow \text{int}; x := \text{int} \} \vdash x : \text{int}}}{\{ f := \text{int} \rightarrow \text{int}; x := \text{int} \} \vdash + : \text{int} * \text{int} \rightarrow \text{int}} \quad \frac{\frac{}{\{ f := \text{int} \rightarrow \text{int}; x := \text{int} \} \vdash 1 : \text{int}}}{\{ f := \text{int} \rightarrow \text{int}; x := \text{int} \} \vdash x+1 : \text{int}}}{\emptyset \vdash \text{fun } f (x : \text{int}) = x+1 : \{ f := \text{int} \rightarrow \text{int} \}}$$

**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.

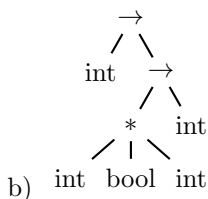
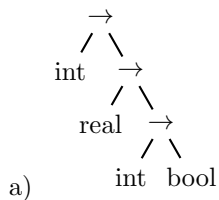
**Lösungsvorschlag 3.3:**



**Aufgabe 3.4** Geben Sie die Baumdarstellungen der folgenden Typen an:

- a) `int → real → int → bool`
- b) `int → int * bool * int → int`

**Lösungsvorschlag 3.4:**



**Aufgabe Knapsack!**

- a) Erstellen Sie eine Prozedur `enumerate : unit -> unit`, die durch Rekursion die Binärdarstellungen der Zahlen von 0 bis 31 ausgibt.
- b) Angenommen Sie haben einen Rucksack und eine Menge von Boxen, die Sie in den Rucksack packen wollen. Dabei haben die Boxen jeweils einen Wert und ein Gewicht, wie aus folgender Liste ersichtlich ist:
  - i) Box 1: 1 kg und 2€
  - ii) Box 2: 12 kg und 4€

- iii) Box 3: 2 kg und 2€
- iv) Box 4: 1 kg und 1€
- v) Box 5: 4 kg und 10€

Sie möchten nun den größtmöglichen Gesamtwert in diesem Rucksack transportieren, doch erlaubt er nur ein Gesamtmaximalgewicht von 15kg.

Schreiben Sie eine Prozedur `solve : unit -> unit`, die ein Tupel von 5 Bits ausgibt, welches die mitzunehmenden Boxen kennzeichnet. (Beispiel: (1,0,0,1,0) gäbe an, dass durch Mitnahme der Boxen 1 und 4 der maximale Wert für ein Gewicht  $\leq 15$ kg erreicht wird.)

### Lösungsvorschlag Knapsack!: Part 1:

```

fun decimal_to_binary state =
  (state mod 2,
   (state div 2) mod 2,
   (state div 4) mod 2,
   (state div 8) mod 2,
   (state div 16) mod 2);

fun print_binary n =
  let
  val t = decimal_to_binary n
  in
  print (Int.toString (#1t) ^
         Int.toString (#2t) ^
         Int.toString (#3t) ^
         Int.toString (#4t) ^
         Int.toString (#5t) ^ "\n")
  end;

fun enumerate_ n =
  if n < 32 then (
  print_binary n;
  enumerate_ (n + 1)
  ) else
  ();

fun enumerate () = enumerate_ 0;

```

### Part 2:

```

fun weight_and_val n =
  let
  val t = decimal_to_binary n
  in
  (#1t + #2t * 12 + #3t * 2 + #4t + #5t * 4,
   #1t * 2 + #2t * 4 + #3t * 2 + #4t + #5t * 10)
  end;

fun solve_ (best_n, best_val, curr_n) =
  let
  val maxw = 15
  val (curr_w, curr_val) = weight_and_val curr_n
  in
  if curr_n < 32 then
  if curr_w <= 15 andalso curr_val > best_val then
  solve_ (curr_n, curr_val, curr_n + 1)
  else
  solve_ (best_n, best_val, curr_n + 1)
  else
  print_binary best_n
  end;

```

```
fun solve () = solve_ (0, 0, 0)
```