Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik (20)
Johann Wolfgang Goethe-Universität Frankfurt am Main

Praktische Informatik 1:

Wintersemester 1997/98

Lösungshinweise zum Übungsblatt Nr. 4

Aufgabe 1

- $a \geq b$, weil $a^{b^n} = b^{log_b(a)*b^n}$ und $b^{log_b(a)*b^n} < b^{a^n}$ gdw. $log_b(a) * b^n < a^n$ gdw. $log_b(a) < (\frac{a}{b})^n$
- $1/k < i$
- $O(e^n)$

Aufgabe 2

1. Berechne 3^{3^n} zähle dann runter bis 0
2. man muß nur mit den Operationen `+`, `-` Funktionen schreiben, mit denen man z.B. 3^{3^n} oder 2^{2^n} berechnen kann, dann zählt man bis 0. Hier eine Lösung von Urs

```
 -- Produkt mit Plus und Minus
prod :: Integer -> Integer -> Integer
prod x y
|(x==0) || (y ==0) = 0
|y == 1 = x
|otherwise = x + prod x (y-1)

 -- Expo mit Minus und `prod`
aHochb :: Integer -> Integer -> Integer
aHochb a b
|(b == 0) = 1
|(b == 1) = a
|otherwise = a `prod` aHochb a (b-1)

 -- Aufruf fuer 3 hoch 3 hoch n
 -- aHochb 3 (aHochb 3 n) where n = 2 (zum Beispiel)

countToZero :: Integer -> Integer
countToZero n
|(n == 0) = 0
|otherwise = countToZero (n-1)

 -- countToZero (3 (aHochb 3 n)) where n = 3 (zum Bspl.)
```

Aufgabe 3

```
potenz_log b n = if n == 0 then 1
                 else  if geradeq n
                       then quadrat (potenz_log b (n ‘div‘ 2))
                                    else b * (potenz_log b (n - 1))

geradeq n = n ‘rem‘ 2 == 0
quadrat x = x*x
```

Anzahl Transformationen:

$$
\begin{array}{lcl}
red(0) & = & 2 \\
red(1) & & \beta, ==,\ \text{Case}, \beta, rem, ==,\ \text{Case}, \beta, red(0) \\
red(1) & = & 10 \\
\multicolumn{3}{l}{\text{für } n \text{ gerade}, n > 0:} \\
red(n) & & \beta, ==,\ \text{Case}, \beta, rem, ==,\ \text{Case}, \beta, red(n/2) + 1, (div), * \\
red(n) & = & red(n/2) + 10 \\
\multicolumn{3}{l}{\text{für } n \text{ ungerade}, n > 0:} \\
red(n) & & \beta, ==,\ \text{Case}, \beta, rem, ==,\ \text{Case}, red(n-1) + 1, (-), * \\
red(n) & = & red(n-1) + 9
\end{array}
$$

Zusammen: Betrachte Binärdarstellung von n mit führender 1. sei $w(n)$ Anzahl der Einsen und $z(n)$ Anzahl der Nullen. Dann ist $red(n) = (w(n) - 1) * 9 + z(n) * 10 + 10$.

$$
\begin{array}{lcl}
mul(n) & = & w(n) + z(n) - 1 \\
div(n) & = & z(n) \\
min(n) & = & w(n) \\
rem(n) & = & w(n) + z(n)
\end{array}
$$

Offenbar ist $red(n) \leq 10 * (w(n) + z(n)) + 1$. Da die Anazhl der Binärstellen einer Zahl n kleiner ist als $\log_2(n) + 1$, ist die Größenordnung $O(\log(n))$.

Aufgabe 4 (10 Punkte)

1.
```
  (\x, y -> x*y) (y + 2) 3
→ (\x’, y’ -> x’*y’) (y + 2) 3
→ (y+2) * 3
```

2.
```
  (let y = z*z in \f -> f y) (\x -> x*y)
→ (let y = z*z in \f -> f y) (\x -> x*y)
→ let y’ = z*z in (\f -> f y’) (\x -> x*y)
→ let y’ = z*z in (\x -> x*y) y’
→ let y’ = z*z in y’*y
```

3.
```
  let y = x*x, x = a*b in a*b*x*y
→ let y = x*x, x = a*b in a*b*(a*b)*(x*x)
→ let y = x*x, x = a*b in a*b*(a*b)*((a*b)*(a*b))
```

4.

```
let z = a*b, b = b in let z = b*a in z
→ let z = a*b, b = b in let z = b*a in b*a
→ let z = a*b, b = b in let z = b*a in b*a
⋮
```