INFORMATIK I

Tutorium 4 — 08. November 2024

```
haskell.hs
```

Name Last
Universität Münster

Übungsblatt 3

1

SPEZIFIKATIONEN

Aufgabe 1: a) ‘Spezifikation’ erfüllt?

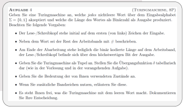
Lies eine **Zahl** von der Tastatur ein, **berechne die Quadratwurzel**, und **gib das Ergebnis am Bildschirm aus**.

Lösung 1: a)

1. Vollständig: Welche Zahlendarstellung? Negative Zahlen? ✖
2. Detailliert: Welche Grundoperationen sind erlaubt? ✖
3. Unzweideutig: Was heißt „ausgeben”? ✖

Aufgabe 1: b) ‘Spezifikation’ erfüllt?

Berechne die Länge des Wortes als Binärzahl, und zwar in der Formulierung aus Aufgabe 4 von Übungsblatt 2.



Aufgabe 4 (Turingmaschine, 8P)

Geben Sie eine Turingmaschine an, welche *jedes* nichtleere Wort über dem Eingabealphabet $\Sigma = \{0, 1\}$ akzeptiert und welche die Länge des Wortes als Binärzahl als Ausgabe produziert. Beachten Sie folgende Vorgaben:

- Der Lese-/Schreibkopf stehe initial auf dem ersten (von links) Zeichen der Eingabe.
- Neben dem Wort sei der Rest des Arbeitsbands mit $\#$ beschrieben.
- Am Ende der Abarbeitung stehe lediglich die binär kodierte Länge auf dem Arbeitsband, der Lese-/Schreibkopf befinde sich über dem höchstwertigen Bit der Ausgabe.
- Geben Sie die Turingmaschine als Tupel an. Stellen Sie die Übergangsfunktion δ tabellarisch dar (wie in der Vorlesung und in der vorangehenden Aufgabe).
- Geben Sie die Bedeutung der von Ihnen verwendeten Zustände an.
- Wenn Sie zusätzliche Bandzeichen nutzen, erläutern Sie diese.
- Es steht Ihnen frei, was die Turingmaschine mit dem leeren Wort macht. Dokumentieren Sie Ihre Entscheidung.

Lösung 1: b)

1. Vollständig: ✔
2. Detailliert: mathematisch formales Maschinenmodell ✔
3. Unzweideutig: Turingmaschine ✔

EUKLIDISCHER ALGORITHMUS

Aufgabe 2: a)

Führen Sie den Algorithmus *schrittweise* für $m = 27$ und $n = 12$ aus.

Lösung 2: a)

$\triangleright\ m = 27, n = 12$

if $m = 0$ **then**
  result $\leftarrow n$
else
  while $n \neq 0$ **do**
    if $m > n$ **then**
      $m \leftarrow m - n$
    else
      $n \leftarrow n - m$
    end if
  end while
  result $\leftarrow m$ $\triangleright$ result $\leftarrow 3$
end if

EUKLIDISCHER ALGORITHMUS

Aufgabe 2: b)

Welches Ergebnis produziert der Algorithmus? Warum führen die Subtraktionen zum gewünschten Ergebnis?

Lösung 2: b)

- Der Algorithmus berechnet den größten gemeinsamen Teiler (ggT) von m und n. ($m = 0 \implies \texttt{return}\ n$; $n = 0 \implies \texttt{return}\ m$)
- Jeder gemeinsame Teiler von $m > n$ muss auch Teiler von $m - n$ sein: $m = t \cdot k_1, n = t \cdot k_2 \implies m - n = t \cdot k_1 - t \cdot k_2 = t \cdot (k_1 - k_2)$

EUKLIDISCHER ALGORITHMUS

Aufgabe 2: c)

Geben Sie das Ergebnis in Form einer Nachbedingung an.

Lösung 2: c)

$\texttt{result} = ggT(m, n)$: $\texttt{result}$ ist Teiler von m und n und für jede Zahl $z : z \mid m \wedge z \mid n$ gilt $z \leq \texttt{result}$.

Lösung 3: b) Arbeit mit der Kommandozeile

```
mkdir Studium
cd Studium
mkdir 2024-WiSe
mkdir 2024-WiSe/Informatik-I
cd 2024-WiSe/Informatik-I
mkdir Folien
mkdir Übungen
```

KOMMANDOZEILE

Lösung 3: b) Arbeit mit der Kommandozeile

```
cd ~/Downloads
mv 00_Organisatorisches_v2.pdf ~/Studium/2024-WiSe/↩
   Informatik-I/Folien
cd ~/Studium/2024-WiSe/Informatik-I/Übungen
ghc -o hello hello.hs
ls                         #> hello hello.hi hello.hs hello.o
/hello                     #> Hello World!
```

Aufgabe 4

Programmieren Sie folgende Funktionen in Haskell:

a) inchesToCentimeters
b) footToCentimeters
c) circleArea

Lösung 4: a) inchesToCentimeters

```haskell
-- 1 Zoll = 2,54 cm
centimetersPerInch :: Float
centimetersPerInch = 2.54

-- Einfach nur multiplizieren :D
inchesToCentimeters :: Float -> Float
inchesToCentimeters inches = inches * centimetersPerInch
```

Lösung 4: b) footToCentimeters

```
-- 1 Fuß = 12 Zoll
inchesPerFoot :: Float
inchesPerFoot = 12

-- Multiplizieren und Umwandeln
footToCentimeters :: Float -> Float
footToCentimeters foot = inchesToCentimeters (foot * ↩
    inchesPerFoot)
```

Lösung 4: c) circleArea

```haskell
-- Radius -> cm -> pi * r^2
circleArea :: Float -> Float
circleArea radiusFoot = pi * (footToCentimeters radiusFoot)^
   2
```

Die Tutoriumsfolien sind jetzt auch auf GitHub !

Name Last
Münster, 13. Januar 2025
name.last@uni-muenster.de