

Vorsemesterkurs Informatik

Sommersemester 2013

Aufgabenblatt Nr. I.1

Aufgabe 1 (Verzeichnisse)

- a) Finden Sie heraus, wie Ihr Homeverzeichnis auf ihrem Rechner, Ihr Username und der Name des Rechners, auf dem Sie arbeiten, heißen.
- b) Legen Sie in Ihrem Homeverzeichnis die im folgenden Baum dargestellten Unterverzeichnisse ausschließlich unter Verwendung der Shell-Befehle `cd` und `mkdir` an, ohne dabei ein `/` zu benutzen.

```
vorkurs
|-- verzeichnis-1
|   |-- unterverzeichnis-1-1
|   |-- unterverzeichnis-1-2
|   '-- unterverzeichnis-1-3
|-- verzeichnis-2
|   '-- .verstecktes_unterverzeichnis-2-1
'-- verzeichnis-3
    |-- unterverzeichnis-3-1
    '-- unterverzeichnis-3-2
```

Welche Kommandos haben Sie eingegeben?

- c) Wechseln Sie in Ihr Homeverzeichnis und führen Sie den Befehl `ls` mit den unten genannten Optionen (auch mit allen Kombinationen) und dem Parameter `vorkurs` aus. Erläutern Sie die Effekte der unterschiedlichen Optionen (schauen Sie dazu in die Man Page von `ls` (mittels `man ls`)).
- `-a`
 - `-l`
 - `-R`
- d) Verwenden Sie den Befehl `tree`, um sich den angelegten Verzeichnisbaum anzeigen zu lassen. Studieren Sie die Man Page zu `tree` und finden Sie heraus, wie man versteckte Verzeichnisse mit anzeigen kann.
- e) Informieren Sie sich über die Verwendung des Befehls `rm`, indem Sie die Man Page dazu durchlesen (mittels `man rm`) und/oder im Internet danach suchen. Löschen Sie die angelegten Unterverzeichnisse von `vorkurs` mithilfe des Befehls `rm`.
- f) Erstellen Sie Unterverzeichnisse von `vorkurs` erneut jedoch unter Verwendung des Kommandos `mkdirhier` (studieren Sie vorher die dazugehörige Man Page). Löschen Sie anschließend die Unterverzeichnisse von `vorkurs` erneut.

Aufgabe 2 (Textdateien)

- a) Legen Sie mithilfe eines Editors eine Datei namens `IrgendeinText.txt` im Verzeichnis `vorkurs` an, und schreiben Sie irgendeinen Text der mindestens 100 Zeilen besitzt und das Wort "Informatik" enthält in die Datei. Sichern Sie die Datei.
- b) Verwenden Sie eine Shell und wechseln Sie in das Verzeichnis `vorkurs` und führen Sie die Kommandos
- ```
cat -n IrgendeinText.txt
more IrgendeinText.txt
less IrgendeinText.txt
head IrgendeinText.txt
tail IrgendeinText.txt
```
- aus. Erklären Sie kurz (evtl. unter Verwendung der Man Pages oder Suchen im Internet), was die einzelnen Kommandos machen.
- c) Führen Sie das Kommando
- ```
tail -n 20 IrgendeinText.txt | head > IET.txt
```
- aus und finden Sie heraus, was dieses Kommando anstellt. Hinweis: Lesen Sie z.B. die Webseite http://www.selflinux.de/selflinux/html/bash_basic03.html.
- d) Verwenden Sie das Kommando `mv`, um die Datei `IrgendeinText.txt` umzubenennen in `MeinText.txt`
- e) Verwenden Sie das Kommando `cp`, um eine Kopie der Datei `MeinText.txt` namens `NochmalMeinText.txt` zu erstellen
- f) Wechseln Sie in Ihr Homeverzeichnis und erstellen Sie eine Kopie des Verzeichnisses `vorkurs` namens `vorkurs2`. Welche Option benötigt `cp` hierfür?
- g) Benennen Sie das Verzeichnis `vorkurs2` in `vorkurs-2` um.
- h) Führen Sie im Homeverzeichnis das Kommando
- ```
grep -R Informatik *
```
- aus. Welche Ausgabe erhalten Sie? Finden Sie anhand der Man Page zu `grep` heraus, was das Kommando macht.

## Aufgabe 3 (Haskell Interpreter: GHCi)

Starten Sie den Haskell Interpreter `GHCi` aus Ihrem Homeverzeichnis.

- a) Verschaffen Sie sich einen Überblick über die Bedienung des `GHCi`, indem Sie sich die Hilfe im `GHCi` anzeigen lassen.
- b) Lassen Sie sich im `GHCi` das Verzeichnis anzeigen, in dem Sie sich befinden. Das Kommando `:!` ist hierbei hilfreich. Wechseln Sie in anschließend in das Verzeichnis `~/vorkurs` ohne den `GHCi` zu verlassen.
- c) Geben Sie zu jedem der folgenden arithmetischen Ausdrücke den entsprechenden Haskell-Ausdruck an und lassen Sie den `GHCi` jeweils dessen Wert berechnen. Füllen Sie dabei die folgende Tabelle aus.

| Arithmetischer Ausdruck                     | Ausdruck in Haskell | Ergebnis im GHCi |
|---------------------------------------------|---------------------|------------------|
| $1 + 3 + 5 + 7 + 9$                         |                     |                  |
| $(15 - 6) \cdot 3 + 12 \cdot 2$             |                     |                  |
| $\frac{1}{2} + \frac{1}{4}$                 |                     |                  |
| $(\frac{1}{3} + \frac{1}{2}) + \frac{1}{6}$ |                     |                  |
| $\frac{1}{3} + (\frac{1}{2} + \frac{1}{6})$ |                     |                  |
| $\frac{1}{0}$                               |                     |                  |
| $-1 \cdot 2$                                |                     |                  |
| $2 \cdot -1$                                |                     |                  |
| $2^{2^2}$                                   |                     |                  |

Hinweis: Verwenden Sie die Operatoren  $*$ ,  $/$  und  $^$  zur Multiplikation, Division und Potenzierung.

## Aufgabe 4 (Funktionalität testen)

Auf der Webseite zum Vorkurs (<http://vorkurs.informatik.uni-frankfurt.de/>) finden Sie eine Datei `magic.hs`. Laden Sie diese herunter und laden Sie sie anschließend in den GHCi. Die Datei stellt die Funktionen `fun1`, `fun2`, `fun3`, `fun4` und `fun5` bereit. Diese erwarten eine Zeichenkette als Eingabe und liefern eine veränderte Zeichenkette. Ein Test ist z.B. `fun1 "Hallo"`.

Finden Sie durch *Testen* der Funktionen `fun1` bis `fun5` heraus, welche der folgenden Änderungen diese Funktionen auf Zeichenketten durchführen (Mehrfachantworten pro Funktion sind möglich) und kreuzen Sie die entsprechenden Antworten an. Beachten Sie, dass der Quellcode der Datei `magic.hs` mit Absicht nahezu unverständlich ist.

| Wirkung                                                                          | fun1                                                         | fun2                                                         | fun3                                                         | fun4                                                         | fun5                                                         |
|----------------------------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|
| 1) macht aus Kleinbuchstaben Grossbuchstaben                                     | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 2) macht aus Grossbuchstaben Kleinbuchstaben                                     | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 3) macht aus dem i-ten Buchstaben des Alphabets die Zahl i                       | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 4) erniedrigt alle Ziffern (ausser der 0) um 1                                   | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 5) macht aus jedem Fragezeichen ein Ausrufezeichen                               | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 6) ersetzt alle Ziffern (außer 0), durch ihre Darstellung als römische Zahl      | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 7) entfernt alle runden Klammern                                                 | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 8) verdoppelt alle Vorkommen der Buchstaben x,y,z,X,Y,Z                          | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 9) vertauscht in Sätzen manche Worte                                             | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 10) entfernt Worte, die mehrfach im Text auftauchen                              | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |
| 11) löscht Leerzeichen, falls Worte mit mehr als einem Leerzeichen getrennt sind | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein | <input type="checkbox"/> Ja<br><input type="checkbox"/> Nein |

## Aufgabe 5 (Boolesche Ausdrücke)

- a) Überlegen Sie sich für die folgenden Booleschen Ausdrücke, *welchen* Wahrheitswert diese jeweils darstellen. Überprüfen Sie *anschließend* Ihr Ergebnis, indem Sie die Ausdrücke im GHCi auswerten lassen. Hierfür müssen Sie die Ausdrücke in Haskell-Notation überführen (d.h. `True` und `False` für die Wahrheitswerte und `not`, `&&`, `||` für  $\neg$ ,  $\wedge$ ,  $\vee$  verwenden).

- $\neg \text{wahr}$
- $\text{wahr} \vee \text{falsch}$
- $\neg((\text{falsch} \vee \text{wahr}) \wedge (\neg \text{falsch}))$
- $\neg((\text{falsch} \vee (\neg \text{falsch})) \wedge (\neg \text{wahr}))$
- $\neg(\neg((\neg \text{wahr}) \vee \text{falsch}))$

- b) Finden Sie durch geeignetes Testen heraus, welcher der beiden Operatoren (`&&`) und (`||`) die höhere *Präzedenz* in Haskell hat, d.h. klären Sie die Frage, ob z.B. `a || b && c` als `((a || b) && c)` oder als `(a || (b && c))` vom GHCi interpretiert wird. Hinweis: Finden Sie einen Testfall, der beide Möglichkeiten eindeutig unterscheidet.
- c) In Haskell gibt es den vordefinierten Ausdruck `undefined`, der für einen Ausdruck steht, welcher nicht in einen Wert überführt werden kann, d.h. die Auswertung von `undefined` führt stets zu einem Laufzeitfehler.

Prüfen Sie, welche Ergebnisse der GHCi für die Ausdrücke

- `False && undefined`
- `True && undefined`
- `False || undefined`
- `True || undefined`

liefert. Stellen Sie eine Vermutung auf, *warum* der GHCi die entsprechenden Ergebnisse liefert.