

Vorsemesterkurs Informatik
Sommersemester 2020

Aufgabenblatt Nr. 4b

Aufgabe 1 (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.

Lösung

`ghci` (oder falls nicht im PATH auf den RBI-Rechnern: `/opt/rbi/bin/ghci`) im Homeverzeichnis aufrufen und `:?` für die Hilfe im Interpreter.

- b) Lassen Sie sich im GHCi das Verzeichnis anzeigen, in dem Sie sich befinden. Das Kommando `:!` ist hierbei hilfreich. Wechseln Sie anschließend in das Verzeichnis `~/vorkurs` ohne den GHCi zu verlassen.

Lösung

`:!pwd` und `:cd vorkurs`.

- 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.

Lösung

$1 + 3 + 5 + 7 + 9$	25
$(15 - 6) * 3 + 12 * 2$	51
$1/2 + 1/4$	0.75
$(1/3 + 1/2) + 1/6$	0.9999999999999999
$1/3 + (1/2 + 1/6)$	1
$1/0$	<i>Infinity</i>
$-1 * 2$	-2
$2 * (-1)$	-2
$2^2^2^2$	65536

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

Lösung

fun1 4) 7)
 fun2 1) 7) 10) 11)
 fun3 3) 6)
 fun4 8) 9) 11)
 fun5 2) 4) 5)

Aufgabe 3 (Boolesche Ausdrücke und Rätsel lösen)

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

Lösung

Im Skript, Abschnitt 3.2.1 finden sich z.B. Wahrheitstabellen und Erläuterungen.

- \neg wahr
ergibt falsch
Haskell:

```
Prelude> not True  
False
```

- wahr \vee falsch
ergibt wahr
Haskell:

```
Prelude> True || False  
True
```

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

```
Prelude> not((False || True) && (not False))  
False
```

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

```
Prelude> not((False || (not False)) && (not True))  
True
```

- $\neg(\neg((\neg\text{wahr}) \vee \text{falsch}))$ *ergibt* $\neg(\neg(\text{falsch} \vee \text{falsch}))$
ergibt $\neg(\neg\text{falsch})$
ergibt \neg wahr
ergibt falsch
Haskell:

```
Prelude> not (not ((not True) || False))  
False
```

- b) Lügenbolde lügen immer, während Wahrsager immer die Wahrheit sagen. Jeder der drei Brüder Knasi, Knesi und Knosi ist entweder ein Lügenbold oder ein Wahrsager. Knasi sagt: "Wir Brüder sind alle Lügenbolde". Knesi sagt: "Genauer einer von uns sagt die Wahrheit".

Formalisieren Sie das Rätsel mit Aussagenlogik in Haskell, indem Sie zunächst definieren

```
knasi_ist_wahrsager = undefined  
knesi_ist_wahrsager = undefined  
knosi_ist_wahrsager = undefined
```

und anschließend die beiden Aussagen in Haskell formulieren und verknüpfen:

```
knasis_aussage = ...
knasis_aussage = ...

beide_aussagen = ...
```

Probieren Sie anschließend durch alle Belegungen (True oder False) für die Variablen knasi_ist_wahrsager, knesi_ist_wahrsager, knosi_ist_wahrsager aus (indem Sie undefined durch die Wahrheitswerte ersetzen) und lassen Sie den Wert von beide_aussagen berechnen.

Für welche Belegung(en) sind beide Aussagen wahr, d.h. wer ist Lügenbold und wer ist Wahrsager?

Lösung

```
-- Knasi sagt: "Wir Brüder sind alle Lügenbolde".
knasis_aussage =
  -- 1. Möglichkeit: Knasi sagt die Wahrheit und alle drei sind L"ügenbolde
  (knasi_ist_wahrsager && not knasi_ist_wahrsager && not
   knesi_ist_wahrsager && not knosi_ist_wahrsager)
  -- oder 2. Möglichkeit: Knasi l"ugt, und nicht alle sind L"ügenbolde
  || (not knasi_ist_wahrsager && not (not knasi_ist_wahrsager && not
   knesi_ist_wahrsager && not knosi_ist_wahrsager))
  -- Knesi sagt: "Genauer einer von uns sagt die Wahrheit".
knasis_aussage =
  -- 1.Möglichkeit: Knesi sagt die Wahrheit und es gibt genau einen
  (knesi_ist_wahrsager && (knasi_ist_wahrsager || knesi_ist_wahrsager
   || knosi_ist_wahrsager)
   && not (knasi_ist_wahrsager && knesi_ist_wahrsager)
   && not (knasi_ist_wahrsager && knosi_ist_wahrsager)
   && not (knesi_ist_wahrsager && knosi_ist_wahrsager))
  ||
  -- 2.Möglichkeit: Knesi lügt und es gibt nicht genau einen
  (not knesi_ist_wahrsager && not ((knasi_ist_wahrsager ||
   knesi_ist_wahrsager || knosi_ist_wahrsager)
   && not (knasi_ist_wahrsager && knesi_ist_wahrsager)
   && not (knasi_ist_wahrsager && knosi_ist_wahrsager)
   && not (knesi_ist_wahrsager && knosi_ist_wahrsager)))

beide_aussagen = knasis_aussage && knesis_aussage

-- nur diese Belegung liefert wahr f"ur beide_aussagen
knasi_ist_wahrsager = False
knesi_ist_wahrsager = True
knosi_ist_wahrsager = False

-- eher lange L"osung, man kann optimieren, z.B. sieht man
-- bei knesis_aussage, dass wenn er die Wahrheit sagt, nur er der wahrsager ist usw.
```