Übungen zu Computergrundlagen WS 2011/2012

Übungsblatt 5: Python und Boole'sche Algebra

15. November 2011

Allgemeine Hinweise

Abgabetermin für die Lösungen ist

- **Dienstag, 22.11., 13:00** für die Übungsgruppen am Mittwoch und Donnerstag
- **Donnerstag, 24.11., 13:00** für die Übungsgruppen am Montag und Dienstag

Die Lösungen solltest Du in eine Kopie der Datei `/share/Courses/CG2011/05/vorlage05.txt` einfügen. Zur Abgabe schickst Du die Lösungsdatei im Anhang einer Email an Deinen Tutor.

Aufgabe 5.1: Python: Funktionen, Rekursionen und Schleifen (6 Punkte)

- 5.1.1 (1 Punkt) Die Sequenz der Fibonacci-Zahlen ist wie folgt definiert:

$$\text{fib}(n) = \begin{cases} 1 & \text{falls } n \leq 1 \\ \text{fib}(n-1) + \text{fib}(n-2) & \text{sonst} \end{cases} \tag{1}$$

  Schreibe eine Python-Funktion `fib1(n)`, die die n-te Fibonacci-Zahl wie in Gleichung 1 berechnet. Füge die Funktion in die Lösungsdatei ein.

  #### Hinweis

  Das Programm `/share/Courses/CG2011/05/pow.py` kannst Du als Vorlage für Dein Programm benutzen. Es enthält eine Funktion `pow`, das die Exponentialfunktion x^n berechnet, die mathematisch wie folgt definiert werden kann:

$$\text{pow}(x, n) = \begin{cases} 1 & \text{falls } n = 0 \\ x \cdot \text{pow}(x, n-1) & \text{sonst} \end{cases} \tag{2}$$

- 5.1.2 (2 Punkte) Schreibe eine Funktion `fib2`, die die Fibonacci-Zahlen mit Hilfe einer Schleife berechnet, also ohne den rekursiven Aufruf der Funktion, und füge sie in die Lösungsdatei ein.

  #### Hinweis

  Berechne zunächst auf dem Papier von Hand die ersten paar Fibonacci-Zahlen. Das macht es einfacher, zu verstehen, wie man die Schleife implementieren kann.

- 5.1.3 (2 Punkte) Eine andere Definition der Fibonacci-Zahlen sieht so aus:

$$\text{fib}_{a,b}(n) = \begin{cases} a & \text{falls } n = 0 \\ b & \text{falls } n = 1 \\ \text{fib}_{b,(a+b)}(n-1) & \text{sonst} \end{cases} \quad (3)$$

$$\text{fib}(n) = \text{fib}_{1,1}(n) \quad (4)$$

Schreibe eine Python-Funktion `fib3`, die die Fibonacci-Zahlen wie in Gleichung 4 berechnet. Füge die Funktion in die Lösungsdatei ein.

Hinweis

Am besten definierst Du die Indizes a und b in Python als optionale Parameter:

```
def fib3(n, a=1, b=1):
  # Hier kommt der Code!
```

- 5.1.4 (1 Punkt) Berechne mit Hilfe der Funktionen aus 5.1.1, 5.1.2 und 5.1.3 die Fibonacci-Zahl für $n = 35$. Was fällt Dir bei den Laufzeiten auf? Wieso sind sie so unterschiedlich?

Hinweis

Zur Laufzeitmessung kannst Du den Unixbefehl `time` wie folgt verwenden:

```
> time python fib.py
121393

real    0m0.080s
user    0m0.070s
sys     0m0.000s
```

Aufgabe 5.2: Boole'sche Algebra (4 Punkte)

Gegeben sei der folgende Boole'schen Ausdruck:

$$F = \neg\Big\{\neg\Big[\neg(a \wedge b) \wedge a\Big] \wedge \neg\Big[\neg(a \wedge b) \wedge b\Big]\Big\}$$

oder in Python-Notation

```
F = not (not (not (a and b) and a) and not (not (a and b) and b))
```

- 5.2.1 (1 Punkt) Stelle für F eine Wertetafel mit jeweils allen Belegungen der Variablen a und b auf.
- 5.2.2 (1 Punkt) Vereinfache den Ausdruck F so lange, bis keine der folgenden Gesetze mehr anwendbar ist: Distributivität, Adsorption, Komplemente, Neutralität, Idempotenz, Extremalgesetze, Doppelnegation, De Morgansche Gesetze.
- 5.2.3 (2 Punkte) Zeige mit Hilfe der Axiome der Boole'schen Algebra:

$$a \vee 0 = a \qquad \text{(Neutralität)}$$
$$a \vee a = a \qquad \text{(Idempotenz)}$$