

4. Übung zur Vorlesung „Informatik I“

Abgabe am Freitag, 21. November, 14.00 Uhr

Bitte geben Sie Ihre Lösungen sowohl schriftlich im Schrein der Informatik, als auch über iLearn ab!

Präsenzaufgabe 1

Ein häufig verwendeter abstrakter Datentyp (ADT) ist der *Stack* (Stapel- oder Kellerspeicher). Auf einem Stack werden Daten ausschließlich nach dem LIFO-Prinzip (*last in - first out*) abgelegt: ein Datum kann mittels der *push*-Operation oben auf den Stack gelegt werden, die *top*-Operation liefert das oberste Datum des Stacks zurück, und die *pop*-Operation entfernt das oberste Datum vom Stack.

Definieren Sie das Stack-Konzept als $ADT(\Sigma, X, E)$. Die Signatur Σ sollte enthalten:

- eine Konstante *emptystack*, die den leeren Stack repräsentiert;
- eine Funktion *isempty?*, die testet, ob ein Stack leer ist;
- die Funktionen *push*, *top* und *pop*.

Für die Gleichungen in E überlegen Sie, welche Eigenschaften die Elemente der Signatur haben müssen, um einen Stack zu definieren.

Machen Sie sich Gedanken zur Implementierung eines Stacks in Scheme. Überprüfen Sie, ob die Implementierung die Gleichungen in E erfüllt.

Aufgabe 2 (Programmieraufgabe)

2+4 Punkte

Die Zahlen, die Sie in Serie 2, Aufgabe 5 berechnet haben, heißen *Binomialkoeffizienten*. Man schreibt $\binom{n}{k}$ für die k -te Zahl in der n -ten Zeile des Pascalschen Dreiecks. Es gilt also für alle $n \in \mathbb{N}$:

$$\binom{n}{0} = 1$$

und für alle $n > 0$ und $k > 0$ mit $n \geq k$:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

Es lässt sich zeigen, dass für alle $n, k \in \mathbb{N}$ mit $n \geq k$ gilt:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \quad (1)$$

- Nutzen Sie diese Formel und die Fakultätsfunktion, um eine Funktion (`binom n k`) zu definieren, die den Wert von $\binom{n}{k}$ berechnet.
- Wie unterscheiden sich die Funktionen `pascal` und `binom` bezüglich ihrer Laufzeit und ihres Speicherbedarfs? Testen Sie beide Funktionen mit geeigneten Werten und begründen Sie Ihre Antwort.

Aufgabe 3 (Programmieraufgabe)

4+2 Punkte

In dem Bruch aus Gleichung 1 (Aufgabe 2) lassen sich einige Faktoren herauskürzen. Führen Sie sich dies für einige Beispiele vor Augen und verallgemeinern Sie Ihre Beobachtung.

- Nutzen Sie diese Verallgemeinerung, um eine Funktion (`binom n k`) zu definieren, die den Wert von $\binom{n}{k}$ in einem *iterativen* Prozess berechnet.
- Vergleichen Sie die Laufzeit und den Speicherbedarf der iterativen Variante mit Ihrer Lösung aus Aufgabe 2.

Aufgabe 4 (Programmieraufgabe)

8 Punkte

Implementieren Sie einen abstrakten Datentyp für komplexe Zahlen. Das DrScheme-System erzeugt mit der Zeile

```
(define-struct komplex (real imag))
```

automatisch einen Konstruktor (`make-komplex r i`), Selektoren (`komplex-real k`) und (`komplex-imag k`) und einen Operator (`komplex? k`) der testet, ob sein Argument eine komplexe Zahl ist. Implementieren Sie Operationen (`komplex+ k1 k2`), (`komplex- k1 k2`), (`komplex* k1 k2`) und (`komplex-abs k`), die komplexe Zahlen addieren, subtrahieren, multiplizieren bzw. deren Betrag berechnen.