

Universität Potsdam  
Institut für Informatik  
Algorithmen und Datenstrukturen

02. Aufgabenblatt

**Lernziele (zum Abhaken):** Die Student:innen können...

- die formale Spezifikation der Schnittstelle eines gegebenen abstrakten Datentyps angeben.
- die Interpretation der formalen Spezifikation eines abstrakten Datentyps angeben.
- die Funktionen einer gegebenen formalen Spezifikation eines abstrakten Datentyps in Python implementieren.
- die Axiome für eine spezifische Eigenschaft eines gegebenen abstrakten Datentyps definieren.

### 3 ADT Pair

*Viele Datentypen verwenden den Typ Tupel oder, spezieller, geordnetes Paar. Z.B. sind Links- und Rechtssequenzen (geschachtelte) geordnete Paare. Dieses Beispiel zeigt, dass die beiden Komponenten eines Paares Werte unterschiedlicher Grundtypen sein können. Bitte beachten Sie das.*

1. Geben Sie eine geeignete formale Spezifikation der Schnittstelle des ADT Pair an.
2. Geben Sie eine geeignete Interpretation (durch Definition der Funktionen) an.
3. Implementieren Sie die Funktionen  $\text{insert}(S, x, p)$  und  $\text{delete}(S, p)$  des ADT Sequenz so, dass  $\text{insert}(S, x, p)$  das Einfügen eines Elements  $x$  als neues Element an die Position  $p$  einer Linkssequenz  $S$  und  $\text{delete}(S, p)$  das Löschen des Elements an der Position  $p$  einer Linkssequenz  $S$  realisiert. **Verwenden Sie dabei den ADT Pair.**

### 4 ADT Menge

1. Geben Sie eine geeignete formale Spezifikation der Schnittstelle des ADT Menge an.
2. Geben Sie eine geeignete semi-formale Interpretation in Form von Vor- und Nachbedingungen an.
3. Formulieren Sie drei Axiome (Gesetze), die die Leerheit von Mengen betreffen und alle Ihre Funktionen berücksichtigen.

### Zusatzaufgabe: Links- und Rechtssequenzen

1. Schreiben Sie die Liste  $[2, 4, 6]$ 
  - (a) als Linkssequenz,
  - (b) als Rechtssequenz.
2. Zur Erinnerung: Für eine Rechtssequenz  $xs$  geben die Funktionen  $\text{last}(xs)$  und  $\text{rest}(xs)$  das letzte Element bzw. die Rechtssequenz mit allen Elementen außer dem letzten Element zurück. Schreiben Sie den Python-Code mit der Definition einer Funktion  $\text{rfirst}$ , die folgenden Algorithmus im funktionalen Stil realisiert:

**Eingabe:** Rechtssequenz  $xs$

**Ausgabe:** erstes Element in  $xs$

3. Sei `fib` die Linkssequenz der ersten Elementen der Fibonacci-Folge, nämlich 1, 1, 2, 3, 5, 8. Welche Linkssequenzen bzw. Werte geben die folgenden Aufrufe zurück?
- (a) `filter(lambda x: x<0, fib)`
  - (b) `filter(lambda x: x%2, fib)`
  - (c) `filter(lambda x: x%2 == 0, fib)`
  - (d) `map(lambda x: x**2, fib)`
  - (e) `reduce(lambda x,y: x*y, fib, 1)`
4. Sei `xs` eine Linkssequenz. Welche Funktion wird durch den folgenden Aufruf von `reduce` berechnet?
- `reduce(lambda x,y: x+1, xs, 0)`