

8. Übung zur Vorlesung „Informatik I“ keine Abgabe

Aufgrund der Übungsklausur am Mittwoch den 17. Dezember, 16.15 Uhr in CAP.3-R.2 gibt es dieses Mal keine Programmieraufgaben.

Die Präsenzaufgaben 1 und 2 sollen nicht beide ausführlich vorbereitet werden. Vielmehr sollten Sie sich das Prinzip der strukturellen Induktion nochmals anschauen.

Präsenzaufgabe 1

Beweisen Sie, dass die Funktion `append`

```
(define (append xs ys)
  (if (empty? xs) ys
      (cons (first xs) (append (rest xs) ys))))
```

assoziativ ist. Zeigen Sie dazu per struktureller Induktion nach `xs`, dass für alle Listen `xs`, `ys` und `zs` gilt:

$$(\text{append } xs \text{ (append } ys \text{ } zs)) = (\text{append } (\text{append } xs \text{ } ys) \text{ } zs)$$

Präsenzaufgabe 2

Gegeben sei folgende Definition:

```
(define (reverse l)
  (reverse-iter l empty))

(define (reverse-iter l e)
  (if (empty? l) e (reverse-iter (rest l) (cons (first l) e))))
```

Beweisen Sie, dass die Funktion `reverse` korrekt ist. Zeigen Sie dazu, dass für alle Listen `l = (list x1 ... xn)` gilt:

$$(\text{reverse } l) \Rightarrow^* (\text{list } xn \text{ } \dots \text{ } x1)$$

Präsenzaufgabe 3

Verwenden Sie die Technik der Nachrichtenweitergabe um einen Zähler zu implementieren. Ein Zähler speichert einen aktuellen Zählerstand, der erhöht und erniedrigt werden kann. Der Zähler soll die folgenden Nachrichten verarbeiten können:

- (a) `'setzen` setzt den aktuellen Zählerstand auf den übergebenen Wert.
- (b) `'erhoehen` erhöht den Zählerstand um 1.
- (c) `'erniedrigen` erniedrigt den Zählerstand um 1.
- (d) `'ablesen` liest den aktuellen Zählerstand aus.