

**Übungen zur Vorlesung**  
**Deklarative Programmierung: Sommersemester 2018**

Nr. 5, Abgabe bis 28.5.2018 11:59 Uhr

**Aufgabe 5.1:** Kontextfreie Grammatik

4 Punkte

Die Dyck-Sprache ist die Sprache aller wohlgeformten Klammerausdrücke (Bsp.: " $(()[]\{\})$ ").  $D_1$  bezeichnet dabei die Art der zu verwendenden Klammern ( ( und ) ),  $D_2$  erweitert  $D_1$  um [ und ] und  $D_3$  fügt { und } hinzu.

- a) Geben Sie die kontextfreie Grammatik der  $D_3$  Dyck-Sprache an.
- b) Geben Sie den Ableitungsbaum für das Wort  $(([])\{\})$  an.

Ein Palindrom ist ein Wort, das Vorwärts genauso wie Rückwärts gelesen werden kann (Bsp.: Lagerregal). Wir betrachten lediglich alle Palindrome, die sich mit den Symbolen 0 und 1 bilden lassen und sich mit dem Symbol C trennen lassen (Bsp.: 1010C0101).

- c) Geben Sie die kontextfreie Grammatik an, welche alle oben beschriebene Palindrome beschreibt.
- d) Geben Sie den Ableitungsbaum für das Wort 0101C1010 an.

**Aufgabe 5.2:** List und Tücke

4 Punkte

In der Vorlesung haben Sie bereits Listen kennengelernt und gesehen, wie Sie damit arbeiten. In dieser Aufgabe sollen Sie einige nützliche Listenfunktionen implementieren:

- a) `(list-remove e l)` entfernt alle Vorkommen von `e` aus der Liste `l` und liefert die resultierende Liste.
- b) `(list-reverse l)` dreht den Inhalt der Liste um.  
(Bsp.: `(list-reverse (list 1 2 3))`  $\rightarrow$  `(list 3 2 1)`)
- c) `(list-flat l)` klopft die Liste `l` flach.  
(Bsp.: `(list-flat '((1) 2 ((3 4) 5) ((( ))) ((6))) 7 8 ( )))`  $\rightarrow$  `'(1 2 3 4 5 6 7 8)`)

**Hinweis:** Sie können bei zur Bearbeitung der Teilaufgaben b und c auf die eingebaute `append`-Funktion zurückgreifen.

### Aufgabe 5.3: Listige Funktionen

4 Punkte
----------

Betrachten Sie folgende Funktion zum konkatenieren zweier Listen:

```
(define (append x y)
  (cond [(empty? x) y]
        [(cons? x) (cons (first x) (append (rest x) y))]))
```

Zeigen Sie unter Verwendung der Äquivalenzregeln (vgl. Skript 9.7), dass gilt:

- a)  $(\text{append } \text{empty } l) \equiv l$  für alle Listen  $l$ .
- b)  $(\text{append } l \text{ empty}) \equiv l$  für alle Listen  $l$ .
- c)  $(\text{append } (\text{cons } x \text{ } l1) \text{ } l2) \equiv (\text{cons } x \text{ } (\text{append } l1 \text{ } l2))$  für alle  $x$  und alle Listen  $l1, l2$ .
- d)  $(\text{append } l1 \text{ } (\text{append } l2 \text{ } l3)) \equiv (\text{append } (\text{append } l1 \text{ } l2) \text{ } l3)$  für alle Listen  $l1, l2, l3$ .

**Hinweis:** Führen Sie den Beweis für die Teilaufgaben b-c via Induktion über  $l$  bzw  $l1$ . Auch wenn Sie eine Teilaufgabe nicht lösen, können Sie bei nachfolgenden Teilen auf deren Aussagen zurückgreifen.

**Aufgabe 5.4: Rocket of the Storm 2.0**

4 Punkte
----------

Das Vermeiden von Kollisionen mit dem Rand der Szene ist auf Dauer etwas langweilig. In der nächsten Ausbaustufe soll der Spieler die Rakete dazu benutzen, Gegenstände einzusammeln. Und das in verschiedenen Levels. Die erreichten Punkte werden nun nicht mehr mit jedem Tick erhöht, sondern mit jedem eingesammelten Gegenstand.

- a) Die einzusammelnden Gegenstände sollen unterschiedliche Formen haben (mindestens Kreise und Quadrate). Mit jedem Gegenstand ist zudem eine Anzahl an Punkten verknüpft, die der Spieler erhält, wenn er ihn einsammelt. Modellieren Sie einen algebraischen Datentyp, der einen solchen Gegenstand abstrahiert.
- b) Ein Level besteht aus einem Namen und einer Anzahl von einzusammelnden Gegenständen. Modellieren Sie Levels durch eine entsprechende Struktur und passen Sie den World-State an, sodass er eine Liste von Levels führt, die der Spieler noch meistern muss.
- c) Bei jedem Tick muss nun überprüft werden, ob die Rakete einen Gegenstand aus dem aktuellen Level berührt. Ist dies der Fall, muss der Gegenstand aus dem Level entfernt werden. Implementieren Sie die nötigen Funktionen. Beachten Sie, dass hierbei Bilder (z.B. die Rakete) als Rechtecke behandeln werden.
- d) Passen Sie den Rest des Spiels an: Vor jedem Level, soll zunächst ein Text mit dem Level-Namen angezeigt werden. Nach einem Klick beginnt der Spieler in der Mitte der Szene mit dem aktuellen Level. Achten Sie darauf, die Geschwindigkeit der Rakete zurückzusetzen! Sind alle Gegenstände eingesammelt, rückt der Spieler ins nächste Level (mit vorherigem Screen). Sind alle Level geschafft, wird eine entsprechende Meldung ausgegeben. Das Spiel ist weiterhin verloren, sobald die Rakete den Rand der Szene berührt. Weiters soll auch die Geschwindigkeit – wie bisher – mit jedem Klick zunehmen.

**Hinweis:** Eine Musterlösung zur letzten Version des Spiels finden Sie im ILIAS.