AI-KI-B: Einführung in die Künstliche Intelligenz
Übungsblatt 01
Abgabe-Deadline: 29.04.2020, 23:55

Racket Basics, Rekursion, Higher-Order Funktionen

Bitte geben Sie bei der Abgabe alle Gruppenmitglieder an. Bitte geben Sie eine .rkt-Datei ab. Bei genauerer Erläuterung, schreiben Sie entweder Kommentare im Code oder geben zusätzlich ein PDF ab. Bitte geben Sie kein Word o.ä. ab. Achten Sie bitte darauf, die geforderten Funktionen so wie angegeben zu benennen, da sonst automatisierte Tests fehlschlagen.

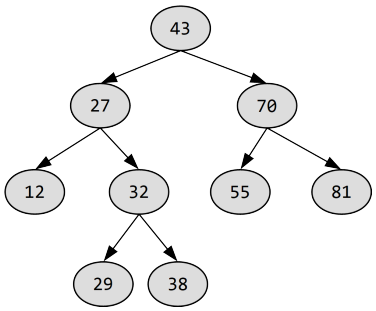
1. Definieren Sie eine rekursive Funktion `search`, welche einen Binärbaum nach einem gegebenen Element durchsucht. Wenn das Element im Baum enthalten ist, soll das Element zurückgegeben werden. Wenn nicht, so soll `#f` zurückgegeben werden. Als Argumente sollen ein beliebiger Binärbaum `tree` nach unten definiertem Schema und ein zu suchendes Element `elem` angegeben werden können.
   Bedingung: Benutzen Sie die `or` Funktion in Ihrer Funktion.

   Beispiele für den unten angegebenen Baum:

   - `(search tree 42) ---> #f`
   - `(search tree 29) ---> 29`

```
'(43
  (27
   12
   (32
    29
    38))
  (70
   55
   81))
```

≡



2. Definieren Sie eine rekursive Funktion `inorder`, die arithmetische Ausdrücke in 'Racket-Schreibweise' (s-Expressions) mit der inorder-Methode traversiert und das Ergebnis als Liste ausgibt. Die korrekte Klammerung soll bei der Ausgabe ignoriert werden.

   Beispiel:

   - `(inorder '(+ (* 5 (- 9 6)) (+ 3 4))) ---> '(5 * 9 - 6 + 3 + 4)`

Fortsetzung auf der nächsten Seite

3. Definieren Sie eine Funktion `zipper`, die zwei gleich lange Listen als Argumente erwartet. Das Ergebnis soll eine ebenso lange Liste sein, welche folgende Bedingungen für die Elemente erfüllen soll: Wenn die Zahl an einer Position in der ersten Liste kleiner ist als die Zahl an derselben Position in der zweiten Liste, soll an dieser Position in der Ergebnisliste die Summe der beiden Zahlen stehen. Ansonsten soll die Differenz der beiden Zahlen an dieser Position stehen.

   Beispiel:

   - `(zipper '(1 7 3 2) '(5 2 2 9)) ---> '(6 5 1 11)`