

Übungen zu Softwareentwicklung III, Funktionale Programmierung

Blatt 4, Woche 5

Leonie Dreschler-Fischer

WS 2012/2013

Ausgabe: Freitag, 16.11.2012,

Abgabe der Lösungen: bis Montag, 26.11.2012, 12:00 Uhr per email bei den Übungsgruppenleitern.

Ziel: Listen, Symbole und Reduktionsstrategien: Die Aufgaben auf diesem Zettel dienen dazu, sich mit der funktionalen Abstraktion vertraut zu machen. Außerdem beschäftigen Sie sich mit dem Unterschied zwischen vorgezogener und verzögerter Auswertung.

Bearbeitungsdauer: Die Bearbeitung sollte insgesamt nicht länger als 4 Stunden dauern.

Homepage:

http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Uebungen_Se_III/Uebungen_Se_III.html

Bitte denken Sie daran, auf den von Ihnen eingereichten Lösungsvorschlägen *Ihren Namen und die Matrikelnummer, den Namen der Übungsgruppenleiterin / des Übungsgruppenleiters und Wochentag und Uhrzeit der Übungsgruppe* anzugeben, damit wir ihre Ausarbeitungen eindeutig zuordnen können.

1 Auswertung von Ausdrücken

Zu welchen Werten evaluieren die folgenden Racket-Ausdrücke? 6 Pnkt.

1. $(\max (\min 2 (- 2 3)) 0)$
2. $'(+ (- 2 4) 2)$

3. (cadr '(Von drauss vom Walde))
4. (cddr '(komm (ich her)))
5. (cons 'Listen '(sind einfach))
6. (cons 'Paare 'auch)
7. (equal? (list 'Racket 'Prolog 'Java) '(Racket Prolog Java))
8. (eq? (list 'Racket 'Prolog 'Java) (cons 'Racket '(Prolog Java))))

2 Textgenerierung

Bearbeitungszeit 2 1/2 Std.

Funksprüche auf See müssen gewissen Formvorschriften genügen: Eine Notmeldung beispielsweise hat die Form:

1. Überschrift:
 - Notzeichen MAYDAY (3×)
 - HIER IST (oder DE, buchstabiert als DELTA ECHO)
 - Schiffsname (3×),
 - Rufzeichen (eine Kennung aus 4 Buchstaben), buchstabiert
 - Notzeichen MAYDAY, Schiffsname, Schiffsname buchstabiert, Rufzeichen buchstabiert
2. Standortangabe,
3. Angabe über die Art des Notfalls,
4. weitere Angaben zur erforderlichen Hilfeleistung,
5. Zwei Peilzeichen (Sendetaste für 10-15 Sekunden drücken)
6. Unterschrift (Schiffsname, Rufzeichen buchstabiert)
7. OVER

Ein Beispiel:

Ein Beispiel: Die Segelyacht *Babette* mit dem Rufzeichen DEJY, die am Sinken ist, könnte so um Hilfe bitten:

MAYDAY MAYDAY MAYDAY
 HIER IST
 BABETTE BABETTE BABETTE DELTA ECHO JULIETT YANKEE
 MAYDAY BABETTE ICH BUCHSTABIERE BRAVO ALFA BRAVO ...
 RUFZEICHEN DELTA ECHO JULIETT YANKEE
 NOTFALLPOSITION UNGEFÄHR 10 SM NORDÖSTLICH LEUCHTTURM KIEL
 NOTFALLZEIT 1000 UTC
 SCHWERER WASSEREINBRUCH WIR SINKEN
 KEINE VERLETZTEN
 VIER MANN GEHEN IN DIE RETTUNGSINSEL
 SCHNELLE HILFE ERFORDERLICH
 ICH SENDE DEN TRÄGER --
 BABETTE DELTA ECHO JULIETT YANKEE
 OVER

1. **Die Grammatik:** Geben Sie eine Grammatik in Backus-Naur-Form 4 Pnkt.
für Notmeldungen an.

2. **Der Generator:** Definieren Sie eine Funktion, die entsprechend Ihrer 7 Pnkt.
Grammatik Notmeldungen generiert.

Die Funktion soll die folgende Argumente haben (alles Strings):

- Der Schiffsnamen,
- das Rufzeichen,
- die Position
- Angaben über die Art des Notfalls und weitere Angaben zur Hilfeleistung.

3. **Der Test:** Erzeugen Sie automatisch den obigen Notruf der *Babette*, 3 Pnkt.
sowie einen Notruf für die Yacht *Bye Bye Love*, Rufzeichen DESY,
die auf der Position 055°45'N, 006°31'E nach Kenterung in schwerer
See sinkt: 8 Mann an Bord, das Schiff ist 12m lang, blauer Rumpf.
Notfallzeit 1020 UTC.

Noch ein Hinweis: Strings können mit **string-append** zu längeren Strings verbunden werden, ähnlich dem **append** für Listen. Orientieren Sie sich bei der Lösung am Beispiel zur Textgenerierung für Listen mit Symbolen, das wir vorletzten Freitag in der Vorlesung besprochen haben.

3 Funktionen vs. Spezialformen:

Bearbeitungszeit 1 Std. 10 Pnkt.

- Erklären Sie den Unterschied zwischen innerer Reduktion und äußerer Reduktion und zeigen Sie beispielhaft die Auswertung des Ausdrucks

```
(sqr (* 3 (+ 1 (sqr 2))))
```

mit

```
(define (sqr x) (* x x))
```

mit beiden Reduktionsstrategien.

- Welche Reduktionsstrategie wird in Racket für Funktionen angewendet und welche für Spezialformen (special form expressions)?
- Alyssa P. Hacker will nicht so ganz einsehen, warum **if** unbedingt eine Spezialform sein muss. Sie fragt sich: „Warum kann ich **if** nicht einfach als ganz normale Funktion definieren und dabei **cond** benutzen?“ Alyssas Freundin Eva Lu Ator meint, dass man dies tatsächlich tun könnte und definiert ihr die folgende neue Version von **if** als Funktion:

```
(define (new-if condition? then-clause else-clause)
  (cond (condition? then-clause)
        (else else-clause)))
```

Hoherfreut benutzt Alyssa das new-if sogleich, um sich ein Programm zur Berechnung der Fakultät zu schreiben:

```
(define (fact-iter product counter max-count)
  (new-if (> counter max-count)
    product
    (fact-iter (* counter product)
                (+ counter 1)
                max-count)))
```

Was passiert, wenn Alyssa versucht, dieses Programm zu benutzen? Erklären Sie anhand dieses Beispiels, warum für manche Kontrollstrukturen Spezialformen unerlässlich sind.

Erreichbare Punkte: 30

Erreichbare Zusatzunkte: 0