

Informatik I

PD Dr. J.-G. Smaus
G. Röger, R. Mattmüller
Wintersemester 2010/2011

Universität Freiburg
Institut für Informatik

Übungsblatt 2

Abgabe: 18. November 2010

Aufgabe 2.1 (Die Macht der Abstraktion – Anfänger, 1+1+2 Punkte)

Nach Klicken auf **Syntaxprüfung** zeigt DrRacket, wenn man mit der Maus über einen Variablennamen fährt, die Verwendung bzw. die Definition an. Verwenden Sie dieses Feature, wenn Sie bei der lexikalischen Bindung Hilfe benötigen.

(a) Betrachten Sie das folgende Scheme-Programm.

```
(define x 2)      ; --> zwei
(define y -1)    ; --> minuseins
(define z -4)    ; --> minusvier

(define f
  (lambda (x z)
    (+ (* x x) z y)))

(f 4 -2)
```

Benennen Sie die Variablen `x`, `y` und `z`, die in den ersten drei Zeilen des Programms definiert werden, im kompletten Programm um. Der neue Name der Variable steht als Kommentar im Programm hinter dem Pfeil (`-->`). Achten Sie bei der Umbenennung auf die lexikalische Bindung. Benennen Sie keine Parameter der Funktion `f` um.

Berechnen Sie, nachdem Sie die Umbenennung durchgeführt haben, von Hand `(f 4 -2)`.

(b) Betrachten Sie das folgende Scheme-Programm.

```
(define x 2)      ; --> zwei
(define y 5)      ; --> fuenf

(define z          ; --> f
  (lambda (x y z)
    (+ x (z y))))

(z y x (lambda (z) (+ x z)))
```

Benennen Sie die Variablen x , y und z , die in den ersten drei (ohne die Leerzeile) Zeilen des Programms definiert werden, im kompletten Programm um. Der neue Name der Variable steht als Kommentar im Programm hinter dem Pfeil ($-->$). Achten Sie bei der Umbenennung auf die lexikalische Bindung. Benennen Sie keine Parameter der Funktion z um.

Berechnen Sie, nachdem Sie die Umbenennung durchgeführt haben, von Hand $(z\ y\ x\ (\text{lambda}\ (z)\ (+\ x\ z)))$.

(c) Betrachten Sie das folgende Scheme-Programm.

```
(define x 1)
(define y 3)
(define z 5)

(define f
  (lambda (x)
    ((lambda (y)
      ((lambda (z)
        (+ z (* x y)))
        (+ x z)))
      (+ x y))))

(f y)
```

Benennen Sie die lokalen Variablen, die innerhalb der Funktion f definiert werden, um. Verändern Sie nicht den Namen der Variablen x , y und z aus den ersten drei Zeilen des Programms.

Welches Ergebnis liefert der Ausdruck $(f\ y)$?

Abgabe: Papier

Hinweis: In diesen Aufgaben finden Sie keine Kommentare und Signaturen zu den Funktionen. Hier können Sie an einem Beispiel sehen, dass es wichtig ist, diese Informationen anderen Programmierern immer zur Verfügung zu stellen. Denn es kann auch bei kleinen Programmen schwer sein, die Funktionsweise der einzelnen Funktionen ohne Kommentare zu verstehen.

Aufgabe 2.2 (Die Macht der Abstraktion – Anfänger, 1+1+2+2 Punkte)

In dieser Aufgabe geht es um Züge. Ein *Waggon* ist eines der folgenden: ein „Erstklassewagen“, ein „Zweitklassewagen“ oder ein „Speisewagen“. Des weiteren soll gelten:

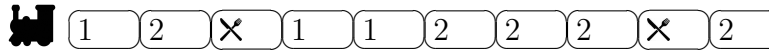
- Eine Lok ist ein Zug;
- wenn z ein Zug ist und *letzterwaggon* ein Waggon, so wird mittels

$(\text{make-kupplung}\ z\ \text{letzterwaggon})$

ein neuer Zug konstruiert.

Beachten Sie, dass ein Zug genau eine Lok hat!¹

- (a) Definieren Sie Sorten für Waggons, für eine Lok, für „nichtleere“ Züge (mit Kupplung), und schließlich für Züge.
- (b) Konstruieren Sie den Zug



und binden Sie ihn an den Namen `l12s11222s2`. (Tipp: um den Überblick zu behalten, definieren Sie Namen `l`, `1`, `11`, `112`, ... für alle Zwischenschritte.)

- (c) Definieren Sie eine Prozedur `waggonanzahl`, die die Waggons in einem Zug zählt. Benutzen Sie `l12s11222s2` und zwei weitere Züge als Testfälle.
- (d) Definieren Sie eine rekursive Prozedur `rationalisiere`, die einen Zug als Eingabe nimmt und den Zug ausgibt, der sich durch Entfernen aller Speisewagen ergibt. Benutzen Sie `l12s11222s2` und zwei weitere Züge als Testfälle.

Abgabe: Papier oder elektronisch als `zug.rkt`.

Um für die Programmieraufgaben Punkte zu erhalten, folgen Sie den Konstruktionsanleitungen der Vorlesung, d. h.:

1. Geben Sie die Signatur an.
Falls die Signatur fehlt, gibt die Aufgabe 0 Punkte.
2. Wählen Sie abhängig von der Signatur das richtige Funktionsgerüst aus.
3. Geben Sie Testfälle an.
4. Schreiben Sie den Funktionsrumpf. Dieser Schritt gliedert sich in weitere Unterschritte, die Sie entsprechend auswählen müssen.
Sie erhalten für diesen Teil der Aufgabe nur Punkte, wenn Sie Schritte 1 bis 3 befolgt haben.

Die Übungsblätter müssen individuell bearbeitet werden. Gruppenabgaben sind nicht zulässig.

¹Dies ist keine Stellungnahme zum Sinn oder Unsinn von Kopfbahnhöfen.