

---

**Informatik I**, Blatt Nr. 5, Abgabe: 27.11.2007 um 11 Uhr  
<http://proglang.informatik.uni-freiburg.de/teaching/info1/2008/>

---

## Hinweise zur Abgabe

Bitte reichen Sie Ihre Abgaben bis zum 27.11.2007 um 11 Uhr ein. Abgaben in elektronischer Form schicken Sie **per Email** an **Ihren** Tutor. Abgaben in Papierform werfen Sie bitte in den **Briefkasten** Ihrer Übungsgruppe im Geb. 051 im Erdgeschoss. Bei jeder Aufgabe ist angegeben, ob Sie elektronisch oder auf Papier abgegeben werden muss.

Bei allen Aufgaben, die Sie per Mail abgeben, müssen Sie sich an die Namenskonventionen der Aufgaben halten. Dies gilt sowohl für die Dateinamen der Abgabe, als auch für Namen von Funktionen. Bitte geben Sie bei der elektronischen Abgabe nur eine Zip-Datei ab. Diese muss alle in den Aufgaben angegebenen `.scm` Dateien (DrScheme) enthalten. Alle Dateien müssen sich in der Zip-Datei in einem Ordner befinden. Der Name dieses Ordners muss Ihrem Loginnamen für den Rechnerpool des Instituts für Informatik entsprechen. Geben Sie unter keinen Umständen Worddokumente usw. ab!

Achten Sie bei der Papierabgabe darauf, dass jedes Blatt Papier Ihrer Abgabe Ihren Namen, Ihre Übungsgruppe, die Blattnummer und den Namen Ihres Tutors trägt. Falls Ihre Papierabgabe aus mehreren Seiten besteht, tackern Sie die Blätter.

Sie können DrScheme im Pool verwenden (starten mit `drscheme`). Achten Sie darauf, dass Sie jeweils das richtige Sprachlevel ausgewählt haben!

## Punktevergabe

Um für die Programmieraufgaben Punkte zu erhalten, folgen Sie den Konstruktionsanleitungen der Vorlesung: Schreiben Sie eine Kurzbeschreibung, führen Sie eine Datenanalyse durch und schreiben Sie einen Vertrag. Erstellen Sie dann die Testfälle und den Rumpf mit der passenden Schablone. Vervollständigen Sie dann den Rumpf der Prozedur und vergewissern Sie sich, dass die Tests erfolgreich laufen. Schreiben Sie Hilfsprozeduren an den Stellen, an denen Sie Teilprobleme lösen!

## Hinweis Endrekursion

Achtung: Wie in der Vorlesung gezeigt, verhindern Verträge die "korrekte" Ausführung endrekursiver Funktionen. Bitte kommentieren Sie in Aufgaben, in denen Sie eine endrekursive Funktion schreiben, den Vertrag aus.

## 1 Aufgabe

*[Sprache: Die Macht der Abstraktion, 6 Punkte]*

Schreiben Sie eine Scheme Prozedur `zip` die zwei Listen nimmt und diese wie ein Reißverschluss zusammenfügt, Beispiel:

`(zip (list 1 2 3) (list 4 5 6 7))`  $\longrightarrow$  `(list 1 4 2 5 3 6 7)`

*Abgabe:* Elektronisch in einer Datei `zip.scm`

## 2 Aufgabe

[Sprache: Die Macht der Abstraktion, 12 Punkte]

Sei die folgende Scheme Prozedur gegeben:

```
; "cat l1 l2" verkettet die Listen l1 und l2.
(: cat ((list %a) (list %a) -> (list %a)))
(define cat
  (lambda (l1 l2)
    (cond
      ((empty? l1) l2)
      ((pair? l1) (make-pair (first l1) (cat (rest l1) l2))))))
```

- (a) Beweisen Sie die Behauptung  $(\text{cat } xs (\text{cat } ys zs)) = (\text{cat } (\text{cat } xs ys) zs)$  durch Listeninduktion. Sie können dabei die in der Vorlesung bewiesene Behauptung

$$(\text{cat empty } ys) = ys \quad (1)$$

verwenden.

- (b) Ist `cat` endrekursiv? Wenn ja: begründen Sie. Wenn nein: Schreiben Sie eine endrekursive Variante `cat-fast`.

*Abgabe:*

- (a) In einer Datei `induct.txt` bzw. `induct.pdf` oder auf Papier  
(b) Elektronisch in einer Datei `endrekursiv.scm`

## 3 Aufgabe

[2 Punkte]

Beschreiben Sie den Unterschied zwischen den beiden Konstrukten `let` und `let*` in eigenen Worten. Geben Sie Beispiele an, die die Unterschiede der beiden Konstrukte verdeutlicht.

*Abgabe:* Elektronisch in einer Datei `let.txt` bzw. `let.pdf` oder auf Papier.