

Info 1 diverse Klausuraufgaben

Aufgabe 1 [2010], [Ergänzung v. Prozeduren, Verwendung der Konstruktionsanleitung]

Betrachten Sie die folgende Prozedur:

```
(define ???  
  (lambda (p lis)  
    (cond  
      [(empty? lis) empty]  
      [(pair? lis) (append (p (first lis)) (??? p (rest lis)))]  
    )  
  )  
)
```

1. Schreiben Sie eine möglichst allgemeine aber aussagekräftige Signatur für die Prozedur.
2. Schreiben Sie eine aussagekräftige Kurzbeschreibung für die Prozedur.
3. Schreiben Sie einen aussagekräftigen Testfall für die Prozedur.
4. Formulieren Sie mit Hilfe von check-property und for-all zwei Eigenschaften für die Prozedur.

Aufgabe 2 [2010], [Rekursion, Endrekursion]

Geben Sie eine endrekursive Version der folgenden Prozedur an:

```
; Potenz einer Zahl berechnen  
(: power (natural natural -> natural))  
(define power  
  (lambda (base exponent)  
    (if (= exponent 0)  
        1  
        (* base (power base (- exponent 1)))))  
  )  
)
```

Aufgabe 3 [2008], [H.O.F.]

Schreiben Sie eine Prozedur fold-cond, welche ein Prädikat pred?: $N \rightarrow \text{boolean}$, ein neutrales Element unit, eine Prozedur f: $N \ N \rightarrow N$ und eine Liste mit natural als Parameter akzeptiert.

Die Prozedur soll alle Zahlen der Liste,

auf die das Prädikat pred? zutrifft, mit f in eine Zahl kombinieren und diese zurückgeben. Die Reihenfolge, in der die Zahlen miteinander kombiniert werden, spielt keine Rolle. Verzichten Sie auf die Verwendung von list-fold und filter.

Kurzbeschreibung und Testfälle werden nicht erwartet!

Aufgabe 4 [2008], [Prozeduren für Listen]

Schreiben Sie eine Prozedur `cycle`, die eine natürliche Zahl `n` und eine Liste mit beliebigen Elementen als Parameter konsumiert. Die Prozedur soll `n`-Mal, jeweils das letzte Element der Liste an den Anfang verschieben und die resultierende Liste zurückgeben.

Beispiel:

```
(cycle 3 (list 1 2 3 4 5 6 7)) -> (list 5 6 7 1 2 3 4)
```

Gehen Sie wie folgt vor:

1. Schreiben Sie eine Prozedur `last`, die eine nicht-leere Liste konsumiert und das letzte Element der Liste zurückgibt.
2. Schreiben Sie eine Prozedur `without-last`, die eine nicht-leere Liste konsumiert und die Liste ohne das letzte Element zurückgibt.
3. Schreiben Sie die Prozedur `cycle` unter Verwendung der Prozeduren `last` und `without-last`.

Aufgabe 3 [<http://ps.informatik.uni-tuebingen.de/teaching/ws15/info1/exercises/02-lab.pdf>], [Erkennen von Fehlern]

(Inkl. eigenen Ergänzungen)

Beim Programmieren und während der Ausführung von Programmen können verschiedene Fehler auftreten.

Sind die folgenden Ausdrücke Syntaxkorrekt? Haben Sie eine Bedeutung? Was ist die Ausgabe, wenn Sie die Ausdrücke von DrRacket auswerten lassen?

1. `(+ 1 0)`

2. `(/ 1 0)`

3.

```
(lambda (a b c)
  (/ (+ (- 0 b) (sqrt (- (* b b) (* 4 a c)))) (* 2 a))
)
```

4. `(define zero (0))`

5.

```
((lambda (n)
  (letrec [(aux
            (lambda (n acc)
              (cond [(= n 0) acc]
                    [else (aux (- n 1) (* acc n))]))])
   (aux (n 1)))
 ) 5)
```