Mathematisches Institut
der Universität München
H. Schwichtenberg
L. Crosilla, M. Seisenberger

Wintersemester 2002/2003
Blatt 6

Übungen zum Ferienkurs über Nichtnumerisches Programmieren

Aufgabe 22. Man schreibe eine einstellige Prozedur `flatten`, die einer Liste aus (geschachtelten) Listen diejenige Liste zuordnet, die aus Nicht-Listen-Elementen aller dieser Listen besteht.
Beispiel: `(flatten '((a b) ((c) d (e f g)))) => (a b c d e f g)`.

Aufgabe 23. Man schreibe eine Prozedur, die ein Objekt daraufhin überprüft, ob es ein korrekter SCHEME-Ausdruck ist.

Implementierung des Interpreters. Die aktuelle Umgebung $U = (U_v, U_p)$ wird durch zwei lange Vektoren (arrays) `var-env` und `pair-env` implementiert. Die baumartige Struktur der Variablenumgebung wird dadurch wiedergegeben, daß jeder Eintrag in `var-env` neben einem Rahmen noch den Index des darüberliegenden Rahmens enthält.

```
(define length-of-var-env 1000)
(define length-of-pair-env 1000)

(define var-env (make-vector length-of-var-env))
(define pair-env (make-vector length-of-pair-env))

(define largest-used-var-env-index -1)
(define largest-used-pair-env-index -1)
```

Die Einträge in `pair-env` sind Paare von Werten. Konstruktor- und Selektorfunktionen für die Paarumgebung:

```
(define (make-pair v1 v2)
  (set! largest-used-pair-env-index
        (+ largest-used-pair-env-index 1))
  (if (>= largest-used-pair-env-index length-of-pair-env)
      (error 'pair-env "Stack overflow")
      (begin
        (vector-set! pair-env
                     largest-used-pair-env-index
                     (list v1 v2))
        (list 'pair largest-used-pair-env-index))))

(define (pair-to-car v) (car (vector-ref pair-env (cadr v))))
(define (pair-to-cdr v) (cadr (vector-ref pair-env (cadr v))))
```

Aufgabe 24. Man schreibe eine Anzeigefunktion `(show-all-pairs)`.

Implementierung des Interpreters (Forts.). Ein typischer Eintrag im Vektor `var-env` hat die Form

$$(((x_1\ v_1)\ (x_2\ v_2)\ \ldots)\ k).$$

Hier ist k ein Zeiger auf den darüberliegenden Rahmen. Konstruktor- und Selektorfunktionen für Rahmen:

```
(define (make-frame vars vs k) (list (map list vars vs) k))
(define (frame-to-bindings frame) (car frame))
(define (frame-to-parent frame) (cadr frame))
```

Wir benötigen weiter folgende Hilfsfunktionen für Variablenumgebungen:

```
(define (insert-frame frame)
  (set! largest-used-var-env-index
        (+ largest-used-var-env-index 1))
  (if (>= largest-used-var-env-index length-of-var-env)
      (error 'var-env "Stack overflow")
      (begin
        (vector-set! var-env largest-used-var-env-index frame)
        largest-used-var-env-index)))

(define (index-to-frame k) (vector-ref var-env k))
```

Aufgabe 25. Man schreibe eine Prozedur `(var-env-lookup x k)`, die den Wert der Variablen `x` an der Stelle `k` der Variablenumgebung liefert.

Abgabetermin. Mittwoch, den 9. Oktober 2002, in der Übung oder (besser) per email an `seisenb@mathematik.uni-muenchen.de`