Scheme

New procedures

1. \( \text{(cons a b)} \) - Makes a cons-cell (pair) from a and b
2. \( \text{(car c)} \) - extracts the value of the first part of the pair
3. \( \text{(cdr c)} \) - extracts the value of the second part of the pair
4. \( \text{(caddr c)} \) - shortcuts. \( \text{(caddr x)} \) is the same as \( \text{(car (cdr x))} \)
5. \( \text{(list a b c ...)} \) - builds a list of the arguments to the procedure
6. \( \text{(adjoin a lst)} \)? - doesn't exist (use \( \text{(cons a lst)} \))
7. \( \text{(list-ref lst n)} \) - returns the \( n \)th element of \( \text{lst} \)
8. \( \text{(append l1 l2)} \) - makes a new list containing the elements of both lists

Problems

1. Draw box-and-pointer diagrams for the values of the following expressions. Also give the printed representation.

(a) \( \text{(cons 1 2)} \)

(b) \( \text{(cons 1 (cons 3 (cons 5 '())))} \)

(c) \( \text{(cons (cons (cons 3 2) (cons 1 0)) '())} \)

\( (((3 . 2) 1 . 0)) \)
2. Write expressions whose values will print out like the following.

(a) (1 2 3)
   (list 1 2 3) or (cons 1 (cons 2 (cons 3 '())))
(b) (1 2 . 3)
   (cons 1 (cons 2 3))
(c) ((1 2) (3 4) (5 6))
   (list (list 2 3) (list 3 4) (list 5 6))

3. Write expressions using car and cdr that will return 4 when the lst is bound to the following values:

(a) (7 6 5 4 3 2 1)
   (car (cdr (cdr (cdr lst))))
(b) ((7) (6 5 4) (3 2) 1)
   (car (cdr (car (cdr lst))))
(c) (7 (6 (5 (4 (3 (2 (1)))))))
   (car (car (cdr (car (cdr (car (cdr lst)))))))
(d) (7 ((6 5 ((4)) 3) 2) 1)
   (car (car (car (cdr (cdr (car (car (cdr lst))))))))

4. Define a predicate list? that takes any value and returns #t if it is a list and #f otherwise.

(define (list? l)
   (cond ((null? l) #t)
         ((pair? l) (list? (cdr l)))
         (else #f)))
Abstraction

Suppose you’re working for the registrar, and she asks you to develop a scheme system to keep track of each student’s registration...

Structures?

Procedures?

See recitation 8 solutions.