MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.001—Structure and Interpretation of Computer Programs
Spring 2006

Recitation 6 Solutions
Data Structures and Abstractions

Scheme

New procedures

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

Problems

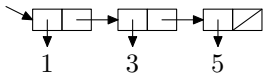
1. Draw box-and-pointer diagrams for the values of the following expressions. Also give the printed representation.

   (a) `(cons 1 2)`

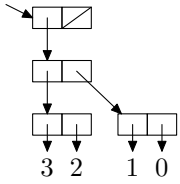
   `(1 . 2)`

   (b) `(cons 1 (cons 3 (cons 5 '())))`

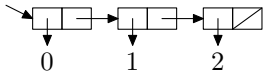
   `(1 3 5)`

   (c) `(cons (cons (cons 3 2) (cons 1 0)) '())`

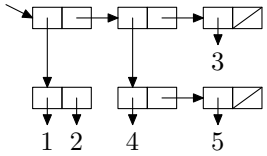
   `(((3 . 2) 1 . 0))`

(d) `(cons 0 (list 1 2))`



`(0 1 2)`

(e) `(list (cons 1 2) (list 4 5) 3)`



`((1 . 2) (4 5) 3)`

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 (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.