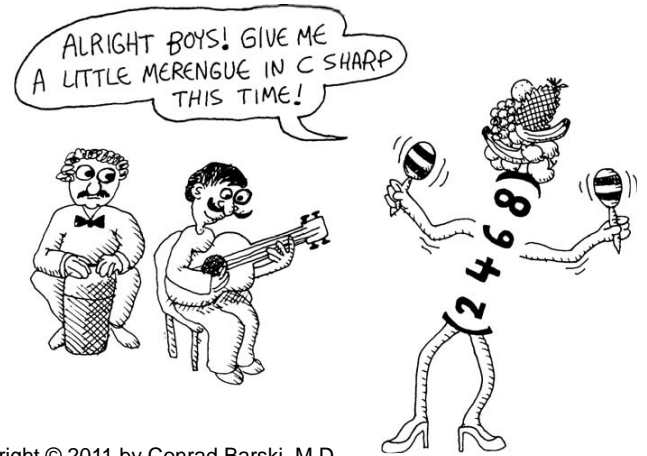


Ch. 7

## Going Beyond Basic Lists Ch. 7

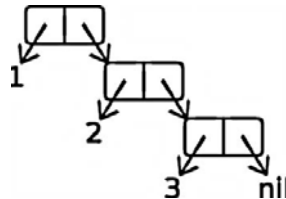


Copyright © 2011 by Conrad Barski, M.D.

### Plain Lists

- End with **nil**
- Example:  
> (cons 1 (cons 2 (list 3)))

(1 2 3)



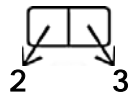
### Exotic Lists –

#### 1) Dotted Pairs and 2) Dotted Lists

- Pairs aka **dotted pairs**, when 1 cons cell has pointers to 2 values

> (cons 2 3)

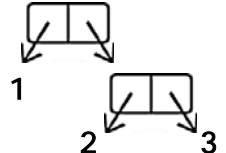
(2 . 3)



- Dotted lists** end with a dotted pair, i.e. they are not terminated w/nil

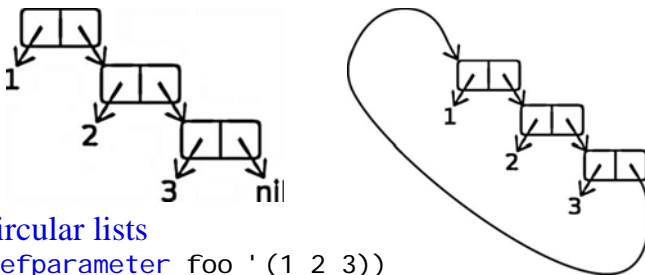
> (cons 1 (cons 2 3))

(1 2 . 3)



### Exotic Lists – 3) Circular Lists

- > (setf \*print-circle\* t) ; Lisp flag enables ; printing circular structures



#### 3. Circular lists

```
> (defparameter foo '(1 2 3))
FOO
```

```
> (setf (caddr foo) foo)
```

```
#1=(1 2 3 . #1#) ; #1= means that it points ; to an existing cell
```

### Exotic Lists – 4) Association Lists

- An **alist** consists of key/value pairs (dotted pairs) stored in a list

```
> (defparameter *drink-order* ; create an alist
  (quote ((bill . double-espresso)
          (lisa . small-drip-coffee)
          (john . medium-latte)) )
  "an alist of drink orders" ) ; doc string
```

```
> (assoc 'lisa *drink-order*) ; retrieve key/value pair
(LISA . SMALL-DRI P-COFFEE)
```

## Alists, cont.

7

```
> (push ; stack operation on list
    (quote (lisa . large-mocha-with-whipped-cream)
            *drink-order*))
((LISA . LARGE-MOCHA-WITH-WHIPPED-CREAM)
 (BILL . DOUBLE-ESPRESSO)
 (LISA . SMALL-DRIP-COFFEE)
 (JOHN . MEDIUM-LATTE))

> (assoc (quote lisa) *drink-order*)
(LISA . LARGE-MOCHA-WITH-WHIPPED-CREAM)

; what happened to the SMALL-DRIP-COFFEE???

; quiz: Lisp functions to extract the key only and then
the value only from a dotted pair, like the one above?
```

## More Complex Data – Structured Lists

8

```
(defparameter *house*
  (quote
    ((walls (mortar
              (cement)
              (water)
              (sand))
            (bricks))
     (windows (glass)
               (frame)
               (curtains))
     (roof (shingles)
            (chimney)) ) )
  )
```

## Context List Structure

9

- Each context-list is structured
  - (<context> <item>\*) ; BNF
  - Each <item> can be a symbol or a context-list
- Example:

```
(walls
  (mortar
    (cement)
    (water)
    (sand))
  (bricks))
```

## Summary

10

### Exotic Lists

- Dotted pairs
- Dotted lists
- Circular lists
- Association lists aka alists
- Other complex data structures – context lists