

Topic 9 Symbols and eq? vs equal?

Section 2.3.1

October 2008

Fall 2008

Programming Development
Techniques

1

Example of Symbol Manipulation

- One of the corner stones of Lisp/Scheme is its ability to easily manipulate symbols.
- Because of this it has been heavily used in Artificial Intelligence.
- As an example, we will write a program to manipulate English Sentences.

Fall 2008

Programming Development
Techniques

2

Symbolic data

```
(quote a) --> a
(quote this-is-a-symbol) -->
      this-is-a-symbol
'a --> a
'(a b d) --> (a b d)
'(1 2 (a (3 d) this is the life))-->
      (1 2 (a (3 d) this is the life))
```

Fall 2008

Programming Development
Techniques

3

Some cautions

```
'c --> (quote c)
'(1 'a 2) --> (1 (quote a) 2)
```

Fall 2008

Programming Development
Techniques

4

The eq? predicate

(eq? x y) returns #t iff x and y are both the same (small) integer, the same symbol or the same pointer to a data structure.

It always returns #f if x and y are real numbers.

Fall 2008

Programming Development
Techniques

5

Some examples

```
(eq? 'a 'a) --> #t
(eq? 'a 'b) --> #f
(eq? 1234 1234) --> #t
(eq? 1234567890 1234567890) --> #f
(define x (list 1 2))
(eq? x x) --> #t
(eq? (list 1 2) (list 1 2)) --> #f
```

Fall 2008

Programming Development
Techniques

6

An efficient membership test (see lisp file)

; takes a symbol named item and a list
; and checks for membership of item in lst

```
(define (memq item lst)
  (cond ((null? lst) #f)
        ((eq? item (car lst))
         lst)
        (else (memq item
                     (cdr lst)))))
```

```
(memq 'a '(b a d a b i n g)) -->
(a d a b i n g)
```

Fall 2008

Programming Development
Techniques

7

A way to count items in a list

```
; counts the number of times symbol item
; occurs in top level of lst
(define (item-count item lst)
  (define (item-count-iter sublist count)
    (let ((next-sublist (memq item sublist)))
      (if (not next-sublist)
          count
          (item-count-iter (cdr next-sublist)
                           (+ count 1)))))
  (item-count-iter lst 0))
```

```
(item-count 'a '(1 a (a a (a)) b 2 a)) --> 2
```

Fall 2008

Programming Development
Techniques

8

What about lists whose elements are not symbols?

; takes two expressions and is #t if they are the
; same symbols, or if they are lists with same elements
; and structure -- i.e., they look the same

```
(define (my-equal? s1 s2)
  (cond ((symbol? s1) (eq? s1 s2))
        ((symbol? s2) #f)
        ((and (null? s1) (null? s2)) #t)
        ((or (null? s1) (null? s2)) #f)
        (else (and (my-equal? (car s1) (car s2))
                    (my-equal? (cdr s1) (cdr s2))))))
```

Fall 2008

Programming Development
Techniques

9