

CS-257L

Nonimperative Programming: Scheme!

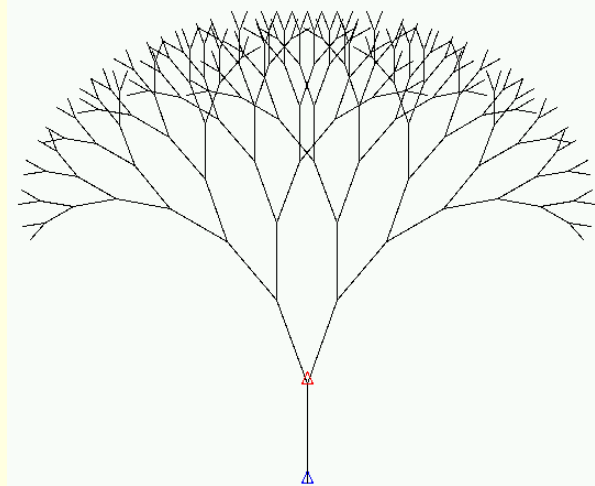
Instructor:

Joel Castellanos

e-mail: joel@unm.edu

Web: <http://cs.unm.edu/~joel/>

Office: Farris Engineering
Center (FEC) room 321



Kenneth Burke - #1

- What is *cdr* (*cdr* (*s*))
where *s* is (a (d e f))?

```
(define s '(a (d e f)))  
(cdr (cdr (s)))
```

- No answer. *Cdr* is a function that expects a single argument. The argument cannot be in parentheses.

Kenneth Burke - #2

- What is *cdr* (*cdr* *s*)

where *s* is ((a (d e f)))?

Note: from the last example, the parentheses were removed from around the *s* and added to the definition of *s*.

```
(define s '((a (d e f))))
```

```
(cdr (cdr s))
```

- No answer, because (*cdr* *s*) is the empty set, ().

Kenneth Burke - #3

- What is *cdr* (*cdr* *s*)
where *s* is (a (d e f))?

```
(define s '(a (d e f)))  
(cdr (cdr s))
```

- () because *cdr* *s* is ((d e f)).

Andrei Buium - #1

- Is this true?

`(atom? cdr)`

- Yes,

because function names [even primitive function names] are also atoms.

atom?

In order to use `atom?` in DrScheme, we first need to define it from the black box given on page 10.

```
(define atom?  
  (lambda (x)  
    (and (not (pair? x)) (not (null? x))  
         )  
  )  
)
```

Andrei Buium - #2

- if p is the list (a b c d e f) ,
how can the functions we know so far be used to
create from p , the list (a b) ?

```
(define p '(a b c d e f))  
(define empty ())  
(cons (car p)  
      (cons (car (cdr p)) empty))  
)
```

Homework

- Create a DrScheme definition file (.scm) that when run will create a list consisting of the first three S-expressions from a given list.
- For this homework, the given list should be hardcoded:

```
(define p '(a b c d e f))
```

- Even though the input is hard-coded, check for bad input.
- Turn-in via WebCT.
- Due Tuesday, 1/29/2008 at midnight.

Andrei Buium - #3

- Why does this return true?

```
(define five 5)
```

```
(define alsoFive 5)
```

```
(eq? five alsoFive)
```

- It shouldn't!
- The atoms are numeric.
- The book lies!
- The Law of Eq?

"The primitive *eq?* takes two arguments. Each must be a non-numeric atom.

eq?

- What is?

(eq? 3 3)

- #t,

because implementations of Scheme are not well standardized.

- The Standard (R5RS) implementation in DrScheme lets the arguments of **eq?** be numeric as well as non-numeric atoms.

Prefix

1. (eq? 3 3)

2. (+ 3 3)

3. (+ 3 (+ 5 2))

(+ 3 7)

10

Convert to Prefix

$$3 \left(\frac{1 + 7}{4} \right)$$

(* 3 (/ (+ 1 7) 4))

(* 3 (/ 8 4))

(* 3 2)

6

Evaluate

(/ 121 / 121 11 = 11
(+ 7 + 7 4 = 11
(/ 12 / 12 3 = 4
(+ + 2 1 = 3
(/ 10 5 / 10 5 = 2
) 1
)
)
)
)
)