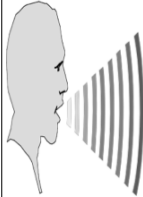

 EEL5840: Elements of Machine Intelligence

Rous

## Announcements




- Announcements:
  - > None
- Today's Handouts in WWW:
  - > LISP Notes 2 & 3
  - > Outline Class 11
  - > Tentative First Exam Date  
Thur. Oct. 8<sup>th</sup> in class
- Web Site
  - > [www.mil.ufl.edu/eel5840](http://www.mil.ufl.edu/eel5840)
  - > Software and Notes
  - > XLISP Documentation



University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

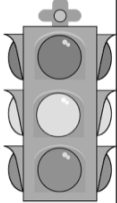
1

 EEL5840: Elements of Machine Intelligence

Rous


## Today's Menu

- LISP LAB 3
  - > Examples & Selected Homework 3 Solutions
  - > Optional Parameters in LISP
  - > Other Types of Parameters in LISP
  - > Functional Arguments



University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

2

 EEL5840: Elements of Machine Intelligence

Rous


## LISP Lab 2 - Examples

Are two SEXes equal?

```
(defun EQUAL (Sex1 Sex2) (cond
  ((ATOM Sex1) (EQ Sex1 Sex2))
  ((ATOM Sex2) nil)
  ((EQUAL (Car Sex1) (Car Sex2))
   (EQUAL (Cdr Sex1) (Cdr Sex2)))
  ))
```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

3

 EEL5840: Elements of Machine Intelligence

Rous

## LISP Lab 2 - Examples

What is the depth of a list?

```
(defun Depth (Lis) (Cond
  ((Null Lis) 0)
  ((Atom Lis) 0)
  (t (max (+ (Depth (Car Lis)) 1)
           (Depth (Cdr Lis)) ))
  ))
```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

4

EEL5840: Elements of Machine Intelligence

Rous

## LISP Lab 2 - Examples

VI. Rewrite union so that the objects remain in order. Call it union1

Ex. (union1 '(a b c) '(f g c d b)) ==> (a b c f g d)  
 (union1 '(it is a) '(big box)) ==> (it is a big box)

```
(defun union1 (s1 s2) (cond
  ((null s1) s2)
  ((member (car s1) s2) (union1 s1 (remove (car s1) s2)))
  ('else (cons (car s1) (union1 (cdr s1) s2))))))
```

UNION1

```
> (union1 '(a b) '(c d))
(A B C D)
> (union1 '(a b c) '(f g c d b))
(A B C F G D)
> (union1 '(it is a) '(big box))
(IT IS A BIG BOX)
```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Rous

## Hwk 3 - Selected Solutions

IV. Unnest

```
(defun unnest (lis) (cond
  ((null lis) lis)
  ((atom (car lis)) (cons (car lis) (unnest (cdr lis))))
  (t (append (unnest (car lis)) (unnest (cdr lis))))))
```

-OR-

```
(defun unnest (lis) (cond
  ((null lis) nil)
  ((atom lis) (list lis))
  (t (append (unnest (car lis)) (unnest (cdr lis))))))
```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Rous

## Hwk 3 - Selected Solutions

VII. Setdiff

```
(defun setdiff(s1 s2) (cond
  ((null s1) nil)
  ((member (car s1) s2) (setdiff (cdr s1) s2))
  ('else (cons (car s1) (setdiff (cdr s1) s2))))))
```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Rous

## Hwk 3 - Selected Solutions

VIII. EQSet

```
(defun eqset(s1 s2) (equal (setdiff s1 s2) (setdiff s2 s1)))
(defun eqset (s1 s2) (cond
  ((null s1) (null s2))
  ((null s2) nil)
  ((and (member (car s1) s2) (member (car s2) s1))
   (eqset (remove (car s2)(remove (car s1) s1))
           (remove (car s1) (remove(car s2) s2))))
  (t nil)))
```

IX. Makeset

```
(defun makeset(lis) (cond
  ((null lis) nil)
  ((member (car lis) (cdr lis)) (makeset (cdr lis)))
  (t (cons (car lis) (makeset (cdr lis))))))
```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Rous

### LISP Lab 3

- **Optional Parameters in LISP**  
We can elegantly define procedures that include optional parameters in LISP by using the &OPTIONAL keyword in the argument list of a function definition. By enclosing the parameter(s) that follow the &optional keyword in parentheses one can supply a default value (otherwise they are automatically defaulted to nil). More than one parameter can follow the &optional keyword.
- *template:*  
(defun fname (arg1 arg2 &optional arg3-argn) (<body>))
- **Example:** Write a function to count the top-level elements of a list  
(defun count-top (lis &optional (answer 0))  
 (if (endp lis) answer  
 (count-top (cdr lis) (+ 1 answer))))

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

9

EEL5840: Elements of Machine Intelligence

Rous

### LISP Lab 3

```
> (defun memset (sex lis &optional (f 'equal))(cond
  ((null lis) nil)
  ((funcall f sex (car lis)) t)
  ('else (memset sex (cdr lis) f))
  ))
MEMSET
> (memset 'a '(a b))      > (memset '(a) '((a b)))
T                          T
> (memset 'a '(a b) 'eq) > (memset '(a) '((a b) 'eq))
T                          NIL
> (memset 'a '(a b) 'equal) > (memset '(a) '((a b) 'equal))
T                          T
```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

10

EEL5840: Elements of Machine Intelligence

Rous

### LISP Lab 3

```
> (defun union2 (S1 S2 &optional (Ans S2)) (cond
  ((null S1) Ans)
  ((member (car s1) S2) (union2 (cdr S1) S2 Ans))
  ('else (union2 (cdr s1) S2 (cons (car S1) Ans))))
  ))
UNION2
> (union2 '(a b) '(c d))    > (union2 '(a b) nil)
(B A C D)                  (B A)
> (union2 '(a b c) '(c d)) > (union2 '(a b) '(a b))
(B A C D)                  (A B)
> (union2 nil '(a b))      > (union2 '(a b) '(b a c))
(A B)                      (B A C)
```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

11

EEL5840: Elements of Machine Intelligence

Rous

### LISP Lab 3

```
/* Function union2 using append instead of cons */
> (defun union2 (S1 S2 &optional (Ans)) (cond
  ((null S1) (append Ans S2))
  ((member (car s1) S2) (union2 (cdr S1) S2 Ans))
  ('else (union2 (cdr s1) S2 (append Ans (list (car S1)))))
  ))
UNION2
> (union2 '(a b) '(c d))
(A B C D)
> (union2 '(a) '(a b c d))
(A B C D)
> (union2 '(a b) '(b c a))
(B C A)
```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

12

EEL5840: Elements of Machine Intelligence

Rous

### LISP Lab 3

- Other Types of Parameters in LISP**  
The &REST parameter in LISP is bound to a list of all otherwise accounted for argument values. Using this we could define a multiple list append as follows:  
> (defun manyappend (&rest lists) (app-h lists))  
MANYAPPEND  
> (defun app-h (lists)  
 (if (endp lists) nil (append (car lists)  
 (app-h (cdr lists))))))

APP-H

```

> (manyappend nil)
NIL
> (manyappend nil '(a b))
(A B)
> (manyappend '(a b) '(c d) '(d e))
(A B C D D E)
> (manyappend '(a b) nil '(c d))
(A B C D)

```

Why do we need a helper fcn?  
The arguments to manyappend are bound in a list on each call.  
\*It adds an extra level of parens\*  
> (manyappend '(a b) '(c d) '(d e))  
creates lists = ((a b) (c d) (d e))  
Recursing on app-h does not do this!  
But recursing on manyappend adds another level of parentheses per call

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Rous

### LISP Lab 3

- Keyword Parameters** are designated by &KEY to allow for the easy identification of many parameters and to bypass positional binding. Using this we can write two versions of member, one using EQ the other using EQUAL.  
(defun MYMEMBER (SEX LIS &KEY TESTFCN) (COND ((NULL LIS) NIL) ((IF (NULL TESTFCN) (EQ SEX (CAR LIS)) (TESTFCN SEX (CAR LIS))) LIS) (T (MYMEMBER SEX (CDR LIS) :TESTFCN TESTFCN))))

```

> (mymember 'a '(a b))
(A B)
> (mymember '(a) '((a) (b)) :testfcn 'equal)
Error: The function TESTFCN is not defined.
Happened in: #-FSubr-COND: #735a14>

```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Rous

### LISP Lab 3

- Functional Arguments**  
Unfortunately this version looks good but it will not run! The LISP reader interprets TESTFCN as the actual name of a function (which it does not have) rather than a variable that represents the actual name of the function. Functional arguments (or parameters) are called *funargs* and demand special processing. The system supplies two functions to deal with funargs: FUNCALL and APPLY.  
template: (funcall #<fname> <arg1> <arg2> ... <argN>)  
(car '(a b c)) == (funcall #'car '(a b c))  
(append '(a b) '(c d)) == (funcall #'append '(a b) '(c d))

Apply works similarly, by using its first argument which must be a *funarg* on the elements of its second argument value, a list.  
template: (apply #<fname> <list of arguments>)  
(car '(a b c)) == (apply #'car '((a b c)))  
(append '(a b) '(c d)) == (apply #'append '((a b) (c d)))

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

EEL5840: Elements of Machine Intelligence

Rous

### LISP Lab 3


- Using funcall we can fix MYMEMBER as follows:  
(defun mymember (sex lis &key testfcn) (cond ((null lis) nil) ((if (null testfcn) (eq sex (car lis)) (funcall testfcn sex (car lis))) lis) (t (mymember sex (cdr lis) :testfcn testfcn))))

```

> (mymember '(a) '((a) (b)))
NIL
> (mymember '(a) '((a) (b)) :testfcn 'equal)
((A) (B))
> (mymember '(a) '((a) (b)) :testfcn 'eq)
NIL

```

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan



EEL5840: Elements of Machine Intelligence

Rous

*Continue with  
LISP Notes 2 & 3  
The End!*

University of Florida  
EEL 5840 - Class #11 - Fall 2009  
© Dr. A. Aravamudan

17