CS 161: Discussion 1

Friday, September 30, 2011

TA: Seph Barker
Email: jbarker@cs.ucla.edu (include CS 161 in the subject line)
Office hours (in BH 2432): Monday 1 - 3 PM (*or by appointment*)

LISP, which stands for LISt Processing, is the most popular AI programming language. It's normally interpreted, not compiled, and note that case does not matter.

Good LISP environments to use are Allegro CL (for windows) or clisp (for OS X or Linux). It's worth learning how to use a good text editor like emacs or vim for editing, as they will take care of a lot of troublesome details like keeping parentheses matched (among other things).

1. **S-expressions**

   - Lisp program = Sequence of S-expressions
   - S-expression = List or Atom
   - List = List of elements (which are lists or atoms)
   - Atom = Building block of lists

2. **Comments**

```
;; This is a Lisp comment
#| This is a block comment in Lisp #|
```

3. **Basic Lisp syntax**

   Every LISP expression is of the form *(¡operator¿ ¡arg1¿ ¡arg2¿ ... ¡argn¿)*. Order of evaluation:

   (a) Evaluate the operator
   (b) Evaluate the arguments
   (c) Apply the value of the operator to the value of the arguments and return the result

4. **Arithmetic Operators**

Operator	Example
$+$	$(+\ 2\ 4\ 10\ 20)$ equal to $2+4+10+20=36$
$-$	$(-\ 2\ 4\ 10\ 20)$ equal to $2-4-10-20=-32$
$*$	$(*\ 2\ 4\ 10\ 20)$ equal to $2*4*10*20=1600$
/	(/ 2 4 10 20) equal to $((2/4)/10)/20=1/400$

5. **Logical Operators**

Operator	Example
not	(not (= a 1))
or	(or (= a b) (= c d))
and	(and (= a b) (= c d))

6. **Conditionals**

The cond operator checks each condition in order and stops as soon as a true condition is found; e.g. the following statement returns 9:

```
(cond
  ((= 3 4) "robot")
  ((= 4 4) 9)          ; <- stops when a true condition is met
  (T "else clause"))
```

7. **List Operations**

Function	Return value
nth	returns specified element of the list (element index taken as parameter)
cons	(cons a b) creates list whose car is a and cdr is b
append	(append a b) appends elements of list b to list a
member	(member x L) searches for element x in list L; returns non-nil list beginning at x if x is found
length	(length L) returns the number of elements in a list
remove	(remove x L) removes atom x from list L; returns T if x was in L; otherwise returns original list L
car	first element of a list
cdr	list with first element removed
caar	car of car of a list
cddr	cdr of cdr of a list
cadr	car of cdr of a list

8. **Useful functions**

```
(load "hw1.lsp")              ; load a file
(load "/ewah/cs161/hw1.lsp")

(trace myfunction)            ; examines recursive invocation of function
(untrace myfunction)

(print 3)                     ; prints argument and then returns it
```

Source: *Lisp cheat sheet*