Assignment #1 -- LISP
CSCI E-220 Artificial Intelligence
Due September 8, 2011

Begin by reading the first few sections of the assigned reading, especially on LISP.

The primary objective of every assignment in this course is to make sure you understand the material. If you are having trouble with any of the problems, ask your teaching assistant or instructor for help early on in the process of doing the assignment.

Assignments in computer science programming, and LISP in particular, have a built-in hint-giving device: namely, the computer itself. If you write a function and are not sure if it works, type it into the LISP interpreter and try it out. If it doesn't work, try individual function calls from within your program, again using the LISP interpreter, to see whether each piece of your function is doing what you think it should.

For all CommonLISP programming assignments, including this one, we will need both the program itself and a runtime session showing the program running on sample input, and we will need both paper and electronic submissions. The paper copy enables us to make corrections and comments and hand it back to you; the electronic copy lets us run and test your program.

There are several ways to make a runtime session script: from your Harvard (UNIX) account, if you type "script filename" at any Unix prompt, the rest of your session up to a ^D (Control-D) will be saved to the filename you specified. You can then run clisp, load and run your program, exit out of clisp, and exit out of script. Alternatively, if you are connected to Harvard from a workstation or home computer using a telnet program, your software probably allows you to save a log of your session to a local file, which can then be e-mailed as an attachment.

To submit an assignment electronically online, all the files associated with that assignment must be in a single subdirectory, and you must run the submit routine from the Unix prompt. As an example, if you have created a subdirectory "asst1" containing the files asst1.l (your LISP functions) and asst1.output (your output script), then you would submit your work by typing:

submit lib220 1 ~/asst1

where the "lib220" is the course name and must be typed exactly as shown, "1" is the assignment number and will change each week, and ~/asst1 is the full name of the directory you've created (the ~/ indicates that asst1 is a subdirectory inside your own home directory, and must be present).

If you submit an assignment and make changes to it later, you can submit it again and your new submission will overwrite your old one. This means, too, that if you need to resubmit your assignment you must submit all of it, not just the new or changed parts.

While you are working in clisp, if you call a function with input that it cannot handle, it will put you into a debugging mode. Type Control-D to return to the top level. If you type Control-D while at the top level, you will exit clisp entirely, so watch out!.

To run clisp you must log in to ice/nice.harvard.edu! You cannot run clisp from fas.harvard.edu.
Problem 1. Find the precise sequence of cars andcdrs that return the atom chair when applied to the following s-expressions. For example, to get chair from ((chair)) requires typing the expression (car (car '((chair)))) or the shortcut form, (caar ‘((chair)). Record your answers the same way:

(a) '((table lamp) (chair terminal))
(b) '(terminal (shelf) ((chair)) (((sofa))))
(c) '((((table) bookcase) chair) telephone)

Problem 2. What does each of the following expressions return as a value if it is typed into the clisp interpreter, and what side effects, if any, does each one have?

(a) (car (setq x '(a b c)))
(b) (car '(setq x (a b c)))

Problem 3. Evaluate each of the following s-expressions. If typing it in to the clisp interpreter would cause an error, tell what the error is. Otherwise, give the result of the evaluation. Predict your answers before trying them with the computer. Ask us if any of the results are unclear.

(a) (cons 'home '(sweet home))
(b) (cons 'home (sweet home))
(c) (list '3 '4 '5)
(d) (list '(3 4 5))
(e) (listp '(3 4 5))

Problem 4. Write a function CUBE that takes one numeric argument and returns the cube of that number. Test it with both positive and negative numbers. What happens with non-numeric input?

Problem 5. Write a predicate NEGATIVEP that returns T if its argument is less than zero, NIL otherwise. Don’t worry about non-numeric input, but notice what happens with it.

Problem 6. Define a predicate LONGER-LIST that takes two lists as arguments, and returns the longer list. If the two lists are equal in length, return T. If one of the arguments is not a list, print an appropriate error message (e.g. “First argument is not a list” or “Both arguments are not lists”) and return NIL.

Problem 7. Find one online news article or news clipping from this week or from the recent past (no earlier than January 2011) of direct relevance to artificial intelligence.

(a) Post a link to this article to the class bulletin board, along with a brief (one-paragraph) summary of the article and its relevance to Artificial Intelligence.
(b) Read postings from your classmates, and post a reply or comment to at least two.