CS 161: Fundamentals of Artificial Intelligence

Prof. Guy Van den Broeck

Fall 2021 – Assignment 1 – Due 11:55pm, Friday, October 8, 2020

Questions

1. This problem concerns a sequence. The first few elements of the sequence are 1 1 1 3 5 9 17 31 57

$$SEQ(n) = SEQ(n-1) + SEQ(n-2) + SEQ(n-3)$$

with

$$SEQ(0) = SEQ(1) = SEQ(2) = 1.$$

Write a single LISP function, called `SEQ`, that takes a single integer argument `N` (`N` >= 0), and returns the Nth Padovan number. For example `(SEQ 0)` returns 1, `(SEQ 3)` returns 3, and `(SEQ 4)` returns 5, `(SEQ 5)` returns 9.

Test your program on at least the first 10 numbers in the sequence. Also test your program for larger values of N. What happens? Explain why in your **hw1.txt** file.

2. Write a single LISP function, called `SUMS`, that takes a single numeric argument `N`, and returns the number of additions required by your `SEQ` function to compute the Nth Padovan number. `SUMS` should not call `SEQ`, but rather you should design the recursion for `SUMS` by examining your `SEQ` code.

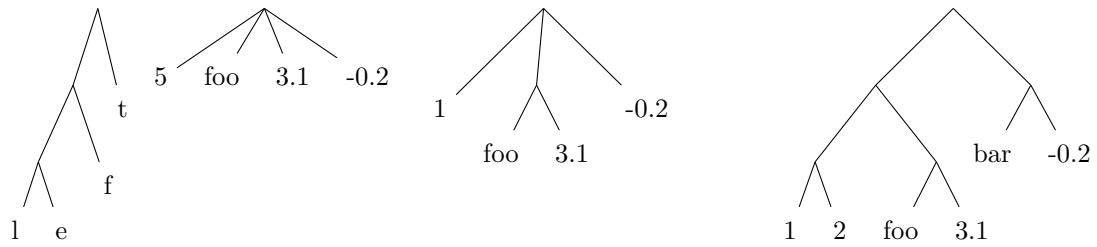
Test your program on at least the first 10 values.

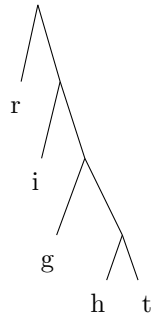
3. A tree can be represented in LISP as follows:

(a) if the tree contains a single leaf node L, it can be represented by atom `L`

(b) if the tree has more than one node and is rooted at N, then it can be represented by a list `(S1 S2 ... Sk)` where `Si` represents the ith subtree of N.

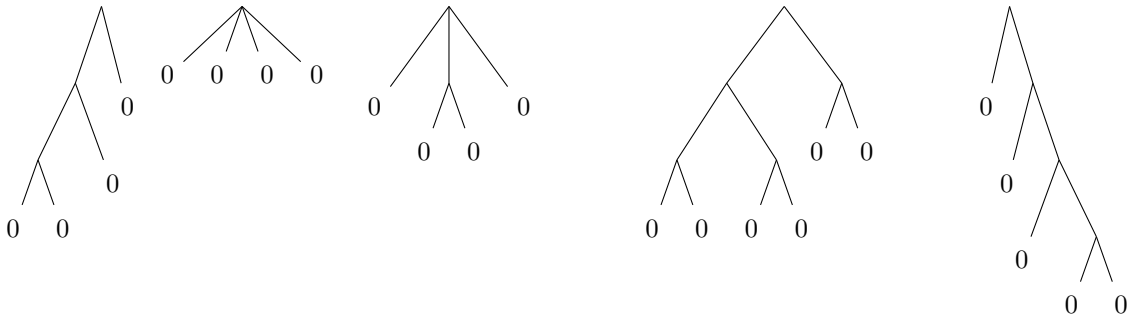
Consider for example the following five trees.





Their LISP representations are respectively `(((l e ) f ) t )`, `(5 foo 3.1 -0.2)`, `(1 (foo 3.1) -0.2)`, `( ((1 2) (foo 3.1)) (bar -0.2))`, and `(r ( i ( g ( h t))))`.

Write a single LISP function, called `ANON`. It takes a single argument `TREE` that represents a tree, and returns an anonymized tree with the same structure, but where all symbols and numbers in the tree are replaced by 0. The anonymized versions of the trees above are as follows.



Test your program on at least these inputs:

```
> (ANON '42)
0
> (ANON 'FOO)
0
> (ANON '(((L E) F) T))
(((0 0) 0) 0)
> (ANON '(5 FOO 3.1 -0.2))
(0 0 0 0)
> (ANON '(1 (FOO 3.1) -0.2))
(0 (0 0) 0)
> (ANON '(((1 2) (FOO 3.1)) (BAR -0.2)))
(((0 0) (0 0)) (0 0))
> (ANON '(R (I (G (H T)))))
(0 (0 (0 (0 0))))
```

Submission

- Submit all solution files **on CCLE**.

- Submit your commented LISP program in a file **named hw1.lsp**.
- In a separate file **named hw1.txt** submit a sample execution showing the values your program returns for the test cases given in the questions. **Do NOT zip your files.**
- Your programs should be written in good style. In LISP, a comment is any characters following a semicolon (;) on a line. Provide an overall comment explaining your solutions. Furthermore, every function should have a header comment explaining precisely what its arguments are, and what value it returns in terms of its arguments. In addition, you should use meaningful variable names.
- The physical layout of the code on the page is very important for making LISP programs readable. Make sure that you use blank lines between functions and indent properly. Programming style will be a consideration in grading the assignment.
- You are restricted to using the following functions, predicates, and operators: quote ('), car, cdr (cadadr, etc.), first, second (third, etc.), rest, cons, list, append, length, numberp, listp, atom, symbolp, oddp, evenp, null, not, and, or, cond, equal, defun, let, let*, =, +, -, *, /. Note: you are **not permitted** to use mutable state (setq, setf, etc.) or last.
- You may assume that all input to your functions are legal; i.e. you do not need to validate inputs.
- Do not write any additional helper functions for your code unless this is explicitly allowed. Test functions are OK.
- Your function declarations should be **exactly as specified** in this assignment. Make sure the functions are spelled correctly, take the correct number of arguments, and those arguments are in the correct order.
- Even if you are not able to implement working versions of these functions, please **include a correct skeleton** of each. Some of these assignments are auto graded and having missing functions is problematic.
- By submitting this homework, you agree to the following honor code.

  You are encouraged to work on your own in this class. If you get stuck, you may discuss the problem with up to two other students, **PROVIDED THAT YOU SUBMIT THEIR NAMES ALONG WITH YOUR ASSIGNMENT. ALL SOLUTIONS MUST BE WRITTEN UP INDEPENDENTLY, HOWEVER**. This means that you should never see another student's solution before submitting your own. You may always discuss any problem with me or the TAs. **YOU MAY NOT USE OLD SOLUTION SETS UNDER ANY CIRCUMSTANCES**. Making your solutions available to other students, **EVEN INADVERTENTLY** (e.g., by keeping backups on github), is aiding academic fraud, and will be treated as a violation of this honor code.

  You are expected to subscribe to the highest standards of academic honesty. This means that every idea that is not your own must be explicitly credited to its author. Failure to do this constitutes plagiarism. Plagiarism includes using ideas, code, data, text, or analyses from any other students or individuals, or any sources other than the course notes, without crediting these sources by name. Any verbatim text that comes from another source must appear in quotes with the reference or citation immediately following. Academic dishonesty will not be tolerated in this class. Any student suspected of academic dishonesty will be reported to the Dean of Students. A typical penalty for a first plagiarism offense is suspension for one quarter. A second offense usually results in dismissal from the University of California.