Foundations of Computer Science – Supervision 1

Heidi Howard

October 13, 2014

This supervision will introduce you to programming in ML. We will cover writing functions and variables, the if-then-else construct, the difference between recursion and iteration and the bool, int and float types. We will cover the syntax for lists and introduce some common list operations. You will write polymorphic functions and use pattern matching to implement recursion.

The most challenge part for most of you will be recurrence relations, covered in exercise 4. If you are struggling: try solving the easier relations from slide 211 of the lecture note or contact me for a hint.

This supervision will cover the material from lectures 1 to 3.

Exercise 1:

One naive solution to the year 2000 bug involves continuing to storing years as two digits, but interpreting them such that 50 means 1950 and 49 means 2049. Code ML functions to:

(a) convert from this new representation to the classical 4 digit representation and back.

(b) test if one year is greater than another years.

(c) add/subtract some given number of years from another year.

[adapted from exercise 1.1 and 1.2 from the course notes]

Exercise 2:

Write a function for finding the median of three integers.

Exercise 3:

Write a function to compute the factorial of a number. Provide both recursive and iterative versions of your function and comment on the improvement in efficiency.

Exercise 4:

An algorithm requires $T(n)$ units of space given an input of size n, where the function T satisfies the recurrence.

$$T(1) = 1$$

and when $(n > 1)$

$$T(n) = T(n/2) + n$$

Prove that the algorithms space requirement in big O notations is linear.

[adapted for the last part of tripos question 1999 P1 Q5]

Exercise 5:

Write a function to compute the sum of a list's elements. Provide both recursive and iterative versions of your function and comment on the improvement in efficiency. Is this function polymorphic? If so, then demonstrate it and if not, explain why.

[adapted from exercise 3.1 from the course notes]

Exercise 6:

Write a function which takes a list and returns a new list containing only elements at even indexes, e.g. given [a, b, c, d] it should return [b, d].

[adapted from exercise 3.3 from the course notes]

Exercise 7:

Write a function called nth, which takes a list and an integer n and returns the nth element of the list. Assume lists are indexed from 0. Is this function polymorphic? If so, then demonstrate it and if not, explain why.