CO225 Lab 1

Ziyan Maraikar

June 29, 2014

Types

1. The Ocaml type system is much *stronger* than C. To explicitly convert between types the Ocaml library provides a collection of functions named x`_of_`y, where x and y are types. For example the function `float_of_int` converts an integer to a floating point, while `bool_of_string` converts string to a boolean.

   Convert the following expressions to valid Ocaml, by using the proper operators. Do not change the operands but use type conversion functions instead. Note down the resultant type of each expression.

   1. `1.1 + 2 * 3`
   2. `true && 0.0`
   3. `"Hello I am" ^ 80 ^ "years old"`
   4. `"true" || "false"`

Functions

Be sure to write each function's type explicitly. Use a local **`let`** to define variables where necessary.

2. Write a function called `capitalise c` that capitalises a single character using the formula `C = c - 'a' + 'A'`, where `C` stands for the character `c` capitalised. (Define constants for `'A'` and `'a'`.)
3. Write functions to calculate nth term of a geometric sequence ar^n and the sum of the first n terms $\frac{a(1-r^m)}{1-r}$.

   Note that the Ocaml exponentiation operator `**` expects a float for the exponent.
4. Define the function `rotate x y theta` that rotates a cartesian point (x, y) about the origin by angle θ.

   To construct a pair of values as the result enclose them in parentheses, e.g. `(x,y)`.