MAT 305: Review #6

April 17, 2015

Directions: The usual counsels apply.

Part I
Modifying an already-written function

1. In class, we wrote a function, `method_of_bisection`, that approximates the root of a function to 2 decimal places. Rewrite the function so that it approximates the root of a function to d decimal places, where d is an argument specified by the client. (A "client" can be either a user, or a program that calls the function.)

Part II
Functions with definite loops

On a previous assignment, you had to decide whether a given set (the *quaternions*) was a group. This required a lot of calculations that you had to request by hand. This time, you'll use functions and loops, so that you can check this for arbitrary sets and operations.

2. Write pseudocode for four algorithms, which will determine whether a finite set S and an operation $\otimes$ satisfy:

   - *closure,* that is, $x \otimes y \in S$ for all $x, y \in S$;
   - *associative,* that is, $x \otimes (y \otimes z) = (x \otimes y) \otimes z$ for all $x, y, z \in S$;
   - *identity,* that is, we can find $\iota \in S$ such that $x\iota = x$ and $\iota x = x$ for all $x \in S$;
   - *inverse,* that is, for any $x \in S$, we can find $y \in S$ such that $xy = yx = \iota$.

   Be sure to specify the inputs needed, and remember that pseudocode should look like english with mathematics, not like Python. *You will lose points* for using Python-isms in pseudocode. **I will do an example of one property in class, which should make the others easier.**

3. Implement three of your algorithms as a Sage function. The operation should be used as a function with two inputs x and y; you can use it by invoking `set_operation(x,y)`. **I will do an example of one property in class, which should make the others easier.**

For the next problem, you need to know what it means to perform an operation "modulo" another number. For full details, take Number Theory or Modern Algebra (or both!) but in essence you can think of "modulo" as the the remainder after division. Sage has a convenient command for this: the `.mod()` command. For instance, `(7+20).mod(12)` gives you 3, because the remainder after dividing 27 by 12 is 3.

4. Test your answers on each of the following possibilities:

   (a) S is the quaternions, and $\otimes$ is matrix multiplication.

   (b) S is the set of integers from `0` to 11, and $\otimes$ is addition, *modulo 12*.

   (c) S is the set of integers from `0` to 11, and $\otimes$ is multiplication, *modulo 12*.