

# Evaluating an Expression

Aditya Gupta

January 2023

In this assignment, we write a program that evaluates a mathematical function for different environments i.e. different mappings of variable names to values. Hence, to evaluate an expression, we want to substitute the variable mappings we obtain from the environment argument we give to the eval() function. So for example for an expression like this :

```
expression = {:add, {:add, {:mul, {:num, 2}, {:var, :x}}, {:num, 3}}, {:q, 1,2}}
```

and for an environment mapping of  $x = 10$ , we get a value of  $\{q, 47, 2\}$  or the rational number  $47/2$  after evaluating the "expression".

## The eval() function

The following code was used to define the evaluation procedure for this assignment.

```
def eval({:num, n}, environment) do {:num, n} end
def eval({:var, v}, environment) do {:num, environment[v]} end
def eval({:add, a1, a2}, environment) do
  add(a1, a2, environment)
end
```

If we have been given a number to evaluate then we return it directly as it's already in its most reduced form. If a variable is passed as an argument then we substitute its value from the environment mapping. If a rational number then we try and reduce it to its most reduced form by dividing both the numerator and the denominator by their greatest common factors and then returning their ratio. In the case that we want to evaluate a bigger expression that contains :add, :sub, :mul or :div then we call the add/3, sub/3, mul/3 or div/3 function based on which operation is part of the expression.

```
def div({:num, n}, {:num, 1}, _) do {:num, n} end
def div({:num, n}, {:num, m}, _) do {:q, n, m} end
def div({:q, n, m}, {:num, a}, _) do {:q, n, a*m} end
def div({:num, a}, {:q, n, m}, _) do {:q, a*m, n} end
def div(a, b, env) do {:div, eval(a, env), eval(b, env)} end
```

This is how I defined my div/3 function where the first 2 arguments are for the expression and the 3rd argument is for the environment.