

Evaluating Expressions - Report

Adrian Hjert

31/Jan/2023

Introduction

The aim of this report is to briefly describe how environments can be used in relation to evaluating expressions. The environment should be able to hold variables with different values, basically a key/value database which will be done using the built in **map()** module in Elixir. The report will show code examples of how expressions are handled and what the output of the program evaluating an expression is. The expressions should be reduced all the way and expressed as rational numbers that are simplified as far as they can be rather than floats, for example $\frac{15}{2}$ rather than 7.5.

Methods

The methods for this task were rather straight forward. It was simply about how we were to handle different expressions and present them as fractions if needed, otherwise as integers. We did however have to account for some cases. These cases can be seen below.

```
#---evaluatiton cases-----
def eval({:num, num}, _) do num end
def eval({:var, var}, map) do Map.get(map, var) end
def eval({:add, e1, e2}, map) do
  add(eval(e1, map), eval(e2, map)) end
def eval({:sub, e1, e2}, map) do
  sub(eval(e1, map), eval(e2, map)) end
def eval({:mul, e1, e2}, map) do
  mul(eval(e1, map), eval(e2, map)) end
def eval({:div, e1, e2}, map) do
  div(eval(e1, map), eval(e2, map)) end
def eval({:q, e1, e2}, _) do rdc(e1, e2) end
```

It was then time to implement the addition, subtraction, multiplication and division functions which is essentially the basic rules for each of these

operations. Division however had some differences as we want to express divisions as simplified fractions if the result is not an integer. Since it is not enough to simply use *rem()* to find the remainder then divide the numbers, we need to use a function called *trunc()* which returns the integer part of a number so we can have answers such as $\frac{15}{2}$ rather than $\frac{15.0}{2.0}$. The *trunc()* function also allows us to present integer solutions as 5 rather than 5.0, for example. Some functions responsible for dividing expressions and expressing them in the way we want them to be expressed (*{:q, n, m}*) can be seen below.

```
def dv({:num, a}, {:num, b}) do
  if rem(a, b) == 0 do
    {:num, trunc(a/b)}
  else
    {:q, a, b}
  end
end

def gcd(n, 0) do n end
def gcd(a, b) do gcd(b, rem(a, b)) end
```

In order to fully reduce some numbers, the following *mul()* case was implemented.

```
def mul({:q, a, b}, c) do
  r = rem(a,b)
  case r do
    0 -> if a < 0 || b < 0 do -trunc(a/b)*c
          else
            trunc(a/b)*c end
    _ -> {:q, a*c, b}
  end
end
```

The map was used based on the following example.

Creating an environment and populating it with key/value pairs:

```
map = %{:x => 1, :y => 2, :z => 3, :w => 4}
```

Retrieving a value associated with a key:

```
Map.get(map, var)
```

Results and Conclusions

Some examples will be presented in this section which show an input expression and the output from the terminal.

```

def test0 do
  map = %{:x => 1, :y => 2, :z => 3, :w => 4}
  expr = {:mul, {:add, {:mul, {:num, 2}, {:var, :w}},
    {:num, 3}}, {:q, 9, 7}}

  eval(expr, map)
end

```

Gives the output $\{q, 99, 7\}$ which would be $\frac{99}{7}$.

```

def test1 do
  map = %{:x => 7, :y => 4, :z => 1, :w => 6}
  expr = {:sub, {:add, {:add, {:num, 2}, {:var, :y}},
    {:num, 3}}, {:q, 5, 9}}

  eval(expr, map)
end

```

Gives the output $\{q, 76, 9\}$ which is $\frac{76}{9}$.

```

def test2 do
  map = %{:x => 7, :y => 2, :z => 8, :w => 5}
  expr = {:mul, {:add, {:mul, {:num, 2},
    {:var, :z}}, {:q, 6, 5}}, {:q, 9, 7}}

  eval(expr, map)
end

```

Gives the output $\{q, 774, 35\}$ which is $\frac{774}{35}$ and finally

```

def test3 do
  map = %{:x => 9, :y => 6, :z => 18, :w => -1}
  expr = {:mul, {:sub, {:mul, {:add, {:mul, {:num, 2},
    {:var, :x}}, {:num, 3}}, {:q, 9, 7}}, {:num, 7}}, {:var, :w}}

  eval(expr, map)
end

```

Gives the output -20 which is a fully reduced version of $\{q, -140, 7\}$.

All of these outputs are reduced as far as possible. This was checked using a calculator.