Assignment 2

Task 3

a)

Question: Formally describe the grammar of the lexems in Task 1.

Answer:

```
V = {c}
S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, −, ∗, /}
R = {(c, e), (c, 1c), (c, 2c), (c, 3c), (c, 4c), (c, 5c), (c, 6c), (c,
7c), (c, 8c), (c, 9c), (c, 0c), (c, +c), (c, −c), (c, ∗c), (c, /c)}
v_s = c
```

Where e = epsilon

b)

Question: Describe the grammar of the records return by the 'ExpressionTree' function in Task 2, using (E)BNF

Answer:

```
<c> ::== e
       | plus(<c>)
       | minus(<c>)
       | multiply(<c>)
       | divide(<c>)
       | 0<c>
       | 1<c>
       | 2<c>
       | 3<c>
       | 4<c>
       | 5<c>
       | 6<c>
       | 7<c>
       | 8<c>
       | 9<c>
```

Where e = epsilon

c)

Question: Which kind of grammar is the grammar you defined in stap a)? Is it regular, context-free, context-sensitive, or unconstrained? What about the one from step b)?

Answer:

Both of the grammars defined in a) and b) are regular, where all the rules follow one of these forms:

```
v ::== s w
v ::== s
v ::== e
```

Where v, w are any non-terminal and s is any symbol S and e = epsilon.