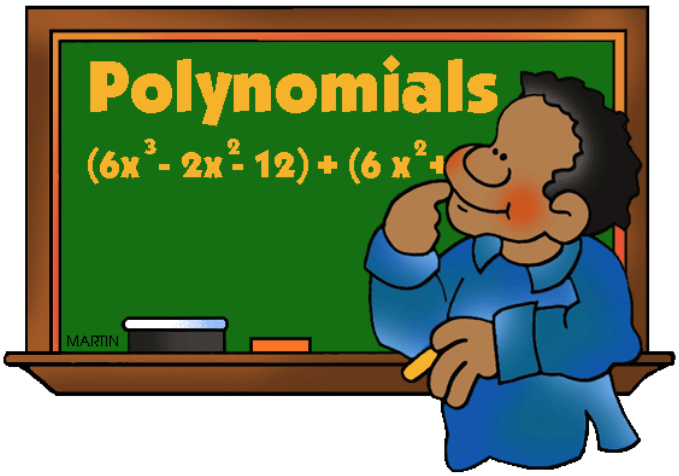
UNIVERSITY OF PUERTO RICO
MAYAGUEZ CAMPUS
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING



Polynomial Solver Language
Final Report

By:
Eduardo Vázquez
Rosedanny Ortiz
Daniel Navarro
Carmelo González

Dr. Wilson Rivera
Department of Electrical & Computer Engineering
University of Puerto at Mayaguez
ICOM4036 – Programing Languages
December 17, 2015

Introduction

Polynomial Solver Language (PSL) is a brand new programming language design to solve the most common mathematical operations on all types of polynomials. The main motivation is based under the impression that writing polynomials in any current language is a little hard, because the user needs to implements data structure and it will not be able to use any type of variable. Therefore using PSL the user will be able to express any polynomial expression with any variable he desires and perform any of the mathematical functions, with the main goal of focusing on the math part and keep practicing of solving polynomials. It would be like solving exercises on paper but using a programming language.

Language Reference Manual:

- String Format:
  - “Your String”
- Number Format:
  - Your number
  - n1 + n2
  - n1 * n2
  - n1 / n3
- Polynomial Format: (n being your last input)
  - [(c1,e1)(c2,e2)(c3,e3)…(cn,en)]
  - [(c1,e1)(c2,e2)(c3,e3)…(cn,en)] + [(c1,e1)(c2,e2)(c3,e3)…(cn,en)]
  - [(c1,e1)(c2,e2)(c3,e3)…(cn,en)] - [(c1,e1)(c2,e2)(c3,e3)…(cn,en)]
  - [(c1,e1)(c2,e2)(c3,e3)…(cn,en)] * [(c1,e1)(c2,e2)(c3,e3)…(cn,en)]
  - [(c1,e1)(c2,e2)(c3,e3)…(cn,en)] / [(c1,e1)(c2,e2)(c3,e3)…(cn,en)]
  - [(c1,e1)(c2,e2)(c3,e3)…(cn,en)] #
  - [(c1,e1)(c2,e2)(c3,e3)…(cn,en)] @(integer value)

Language Development:

For the development of PSL python was used. We wanted to experiment with one of the newest language out there, and used python for the translator of PSL. The first step was to design the lexer with the specifications of our language manual. The main purpose was to give the user an easy language to work with polynomials, the syntax used is very simple and the user can concentrate on just the result.

After the lexer and parser were concluded we implemented a Polynomial class also using python that does all the operations behind the scenes. The user writes the operation he wants for the polynomial, after this is done the class is called and performs the operation desired. A toString method was implemented with the purpose of making the user life easier when reading the output of the desired operation.

For testing purposes we wrote all the possible operations available for this betta version of PSL. The code for testing is available at our GitHub link along the working code and the polynomial class. For more detail info the code will be available for download.

Conclusion:

We found this project interesting, because we didn't know what was behind the scenes of every programming language. Now we know how hard is to implement a language, and appreciate more the famous languages like C and Java that are so complex. It took us time to do a simple language as it is PSL we can now imagine the complexity behind all those famous languages. Right now PSL is only on the betta version, it can get much better with time and dedication. For the future some of the features that can be added are different polynomials variables (for this betta version all polynomials use x as variable). The operation of multi-variable derivate can be implemented. This could be huge for the new generation to study using PSL on tablets or laptops.