Polynomial Syntax Language - PSL

CMPE 152

Team Mak & Cheese

Alexis Cheung Ho

Alex Norell

Timothy Wu

Daniel Soto

**Polynomial Syntax Language (PSL)**

Polynomial Syntax Language (PSL) is a simple polynomial-based language meant to perform a couple basic operations on polynomials.

PSL has three basic data types:

* Polynomials (made up of monomials)
* Constants
* Void

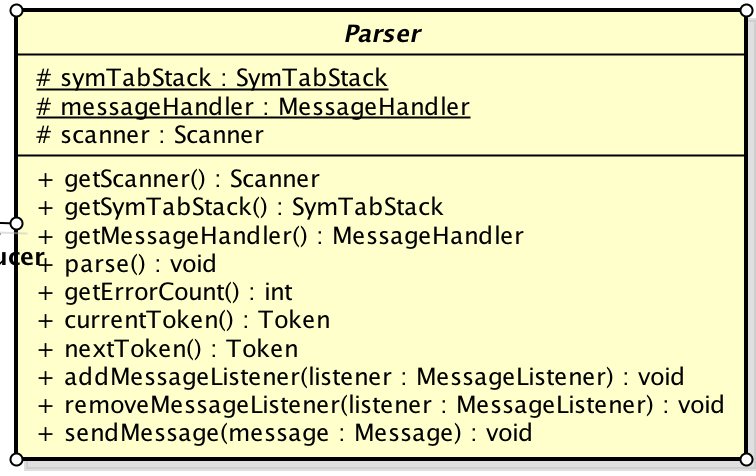
Polynomials are a recursive data type that are made up of monomials. Constants are also included. Currently the only operation supported is addition. Multiplication and division of polynomials are non-trivial operations which are not included in PSL. PSL also does not support negative exponents.

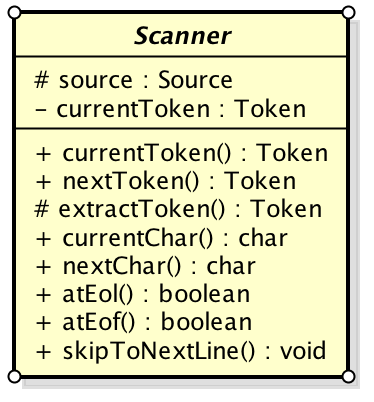
Arrays are the sole data structure used in PSL to store polynomial coefficients. For example, the polynomial below is stored as follows:

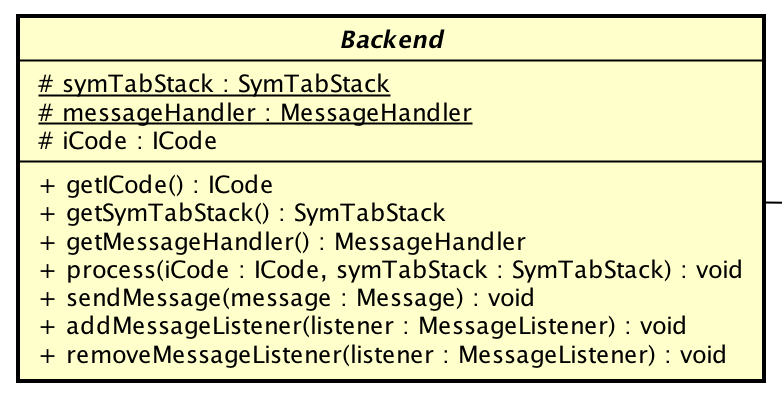
* 3 + 2x^1 + + 3x^3
* [3, 2, 0, 3]

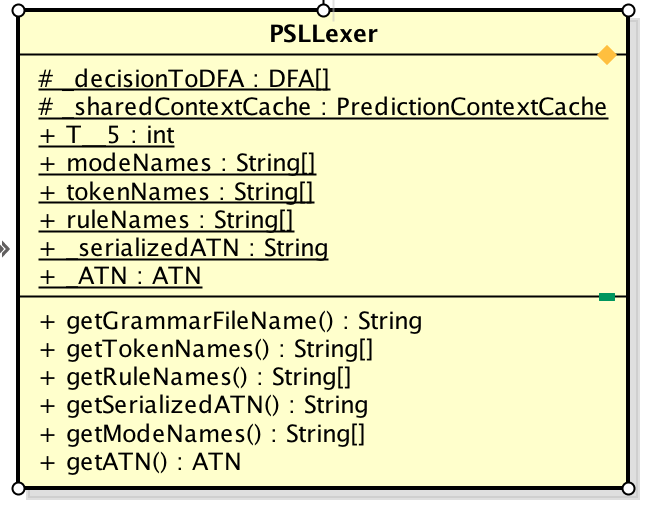
Currently, PSL supports assignment statements, declarations, print statements (PRINT), loop statements (REPEAT), conditional statements (ORDER), and expressions for “add”, “ () ”, and “ ;) ”.

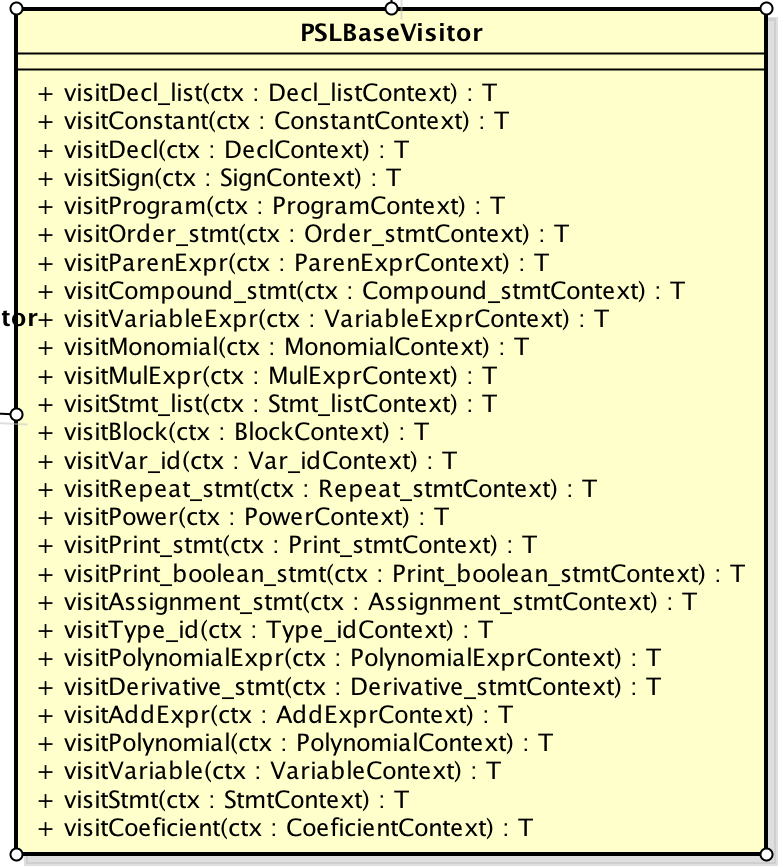
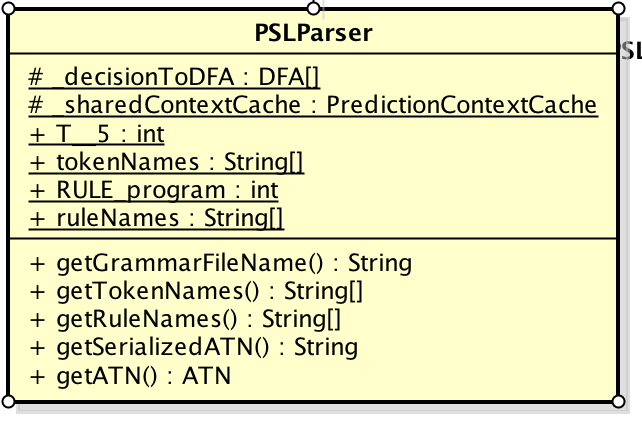
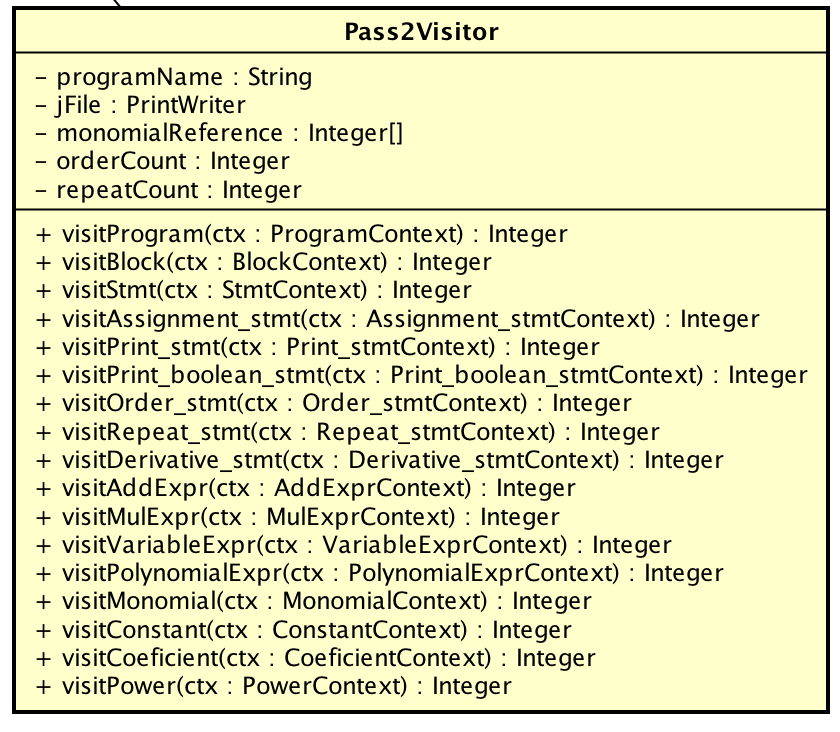
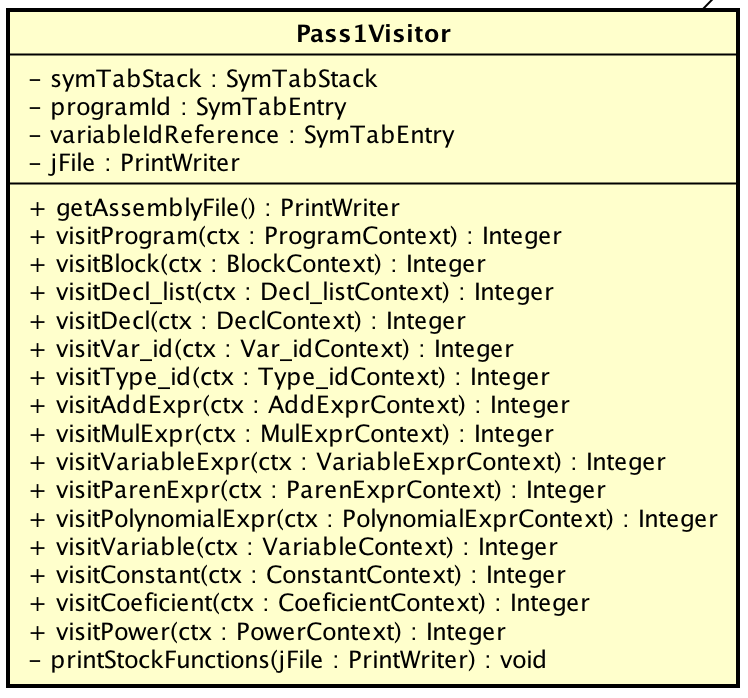
**UML Diagrams**



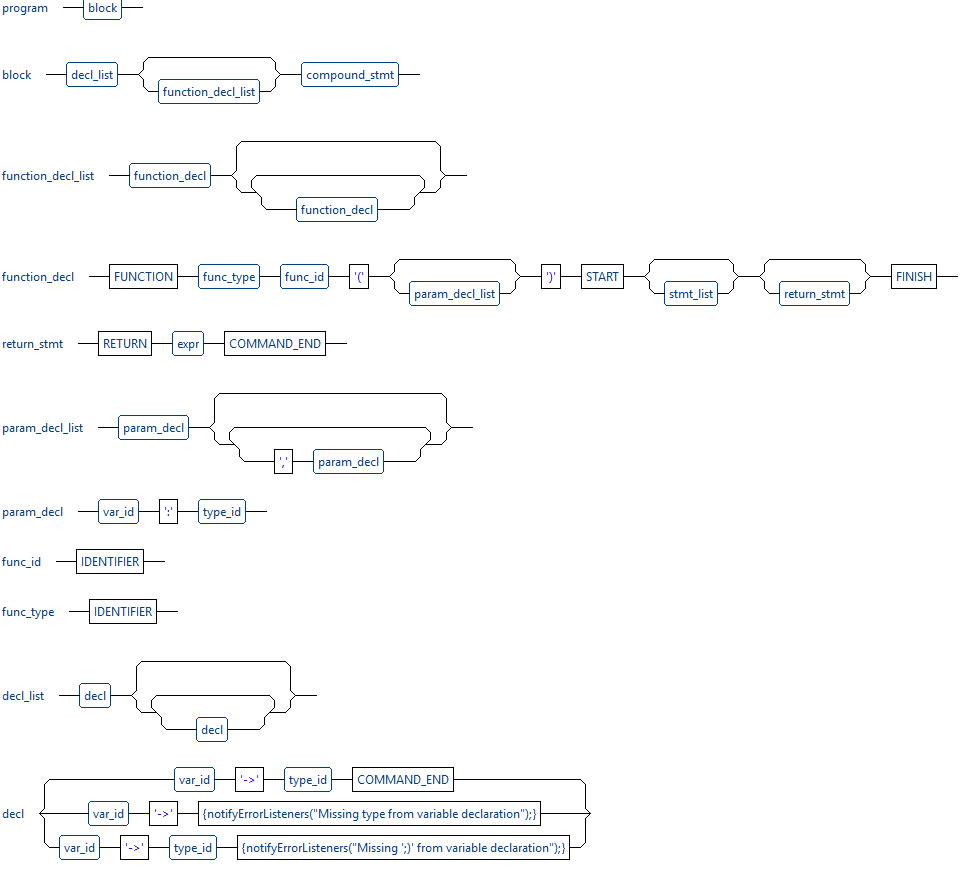


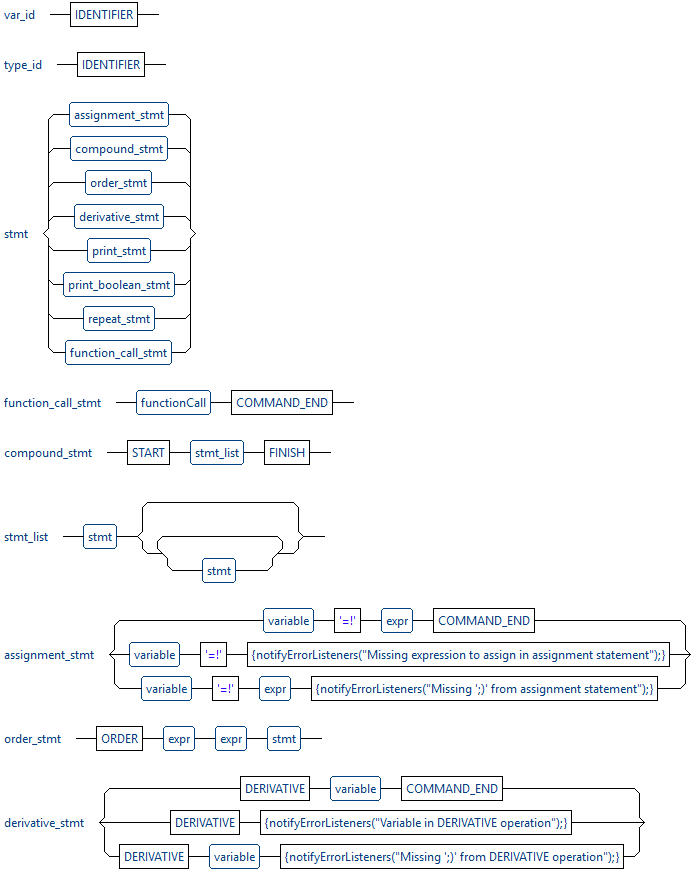


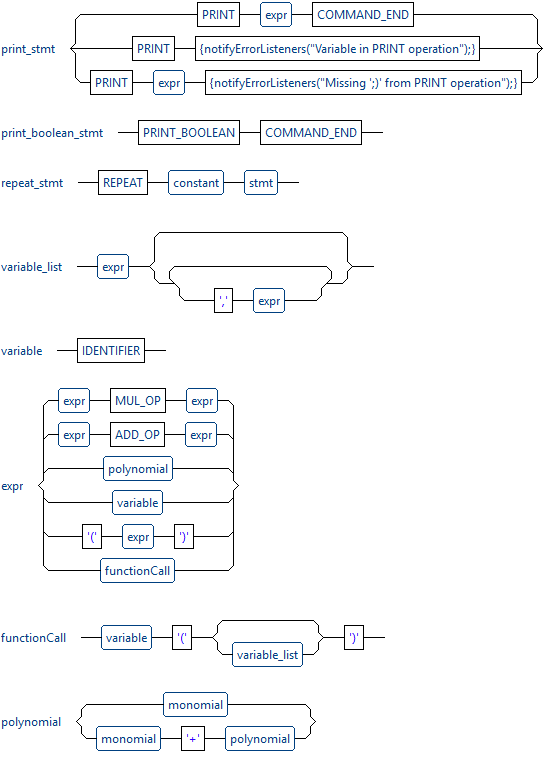


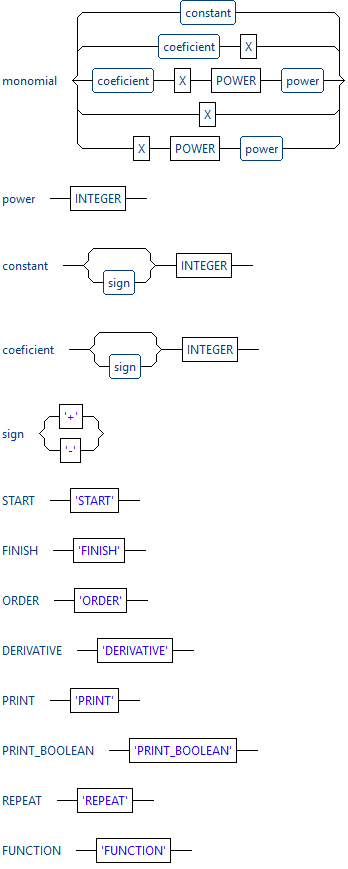
****

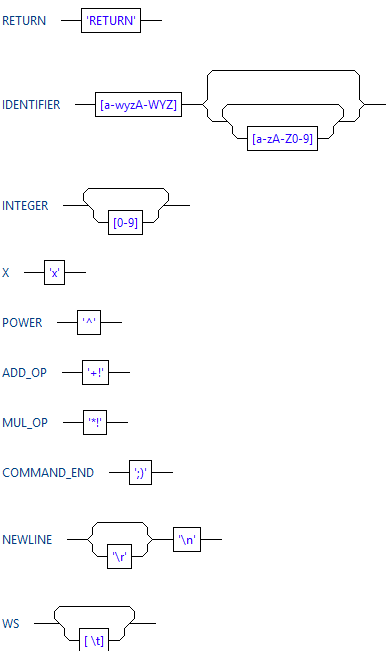
**Grammar File as Syntax Diagrams**











**Generated Jasmin Code Templates**

**ORDER 3 p2 START**

**DERIVATIVE p2 ;)**

**FINISH**

; ORDER3p2STARTDERIVATIVEp2;)FINISH

bipush 10

newarray int

putstatic PSL/temp1 [I

getstatic PSL/temp1 [I

bipush 0

bipush 3

iastore

getstatic PSL/temp1 [I

getstatic PSL/p2 [I

invokestatic PSL/order([I[I)I

ifeq LabelExitOrder1

; DERIVATIVEp2;)

getstatic PSL/p2 [I

invokestatic PSL/derive([I)[I

putstatic PSL/p2 [I

LabelExitOrder1:

**REPEAT 2 START**

**DERIVATIVE p1 ;)**

**PRINT p1 ;)**

**FINISH**

; REPEAT2STARTDERIVATIVEp1;)PRINTp1;)FINISH

bipush 2

LabelBeginRepeat1:

dup

ifeq LabelExitRepeat1

; DERIVATIVEp1;)

getstatic PSL/p1 [I

invokestatic PSL/derive([I)[I

putstatic PSL/p1 [I

; PRINTp1;)

getstatic PSL/p1 [I

invokestatic PSL/print\_array([I)V

bipush 1

isub

goto LabelBeginRepeat1

LabelExitRepeat1:

### 

### **How to Build / Run**

### **Windows Only**

Use in main directory /

.\run\_jfile.bat <filename.j>

Example

.\run\_jfile.bat loop\_control

### **Sample programs & outputs can be found in /src**

Output file format: programname.output

#### **arithmetic\_example.psl**

Assigns polynomials to p1, p2, p3

Adds p1 and p2 and stores it in p3

#### **function\_example.psl**

Void type functions

Polynomial type functions

Return value in function createPolynomial(polynomial)

Pass by value in function foo

#### **loop\_control.psl**

Loop statement (REPEAT >> repeats n times)

Control statement (ORDER >> True if highest exponent is larger than constant)