Parser

Student: Alexandrescu Andrei-Robert, 931/1

Current lab: [Repository Link]

Implementation details

Class Parser is responsible for implementing the LL(1) parser. It contains the first and follow methods that are used to obtain the *first* and *follow* tables. These two methods are implemented in the same way they were specified in the lectures.

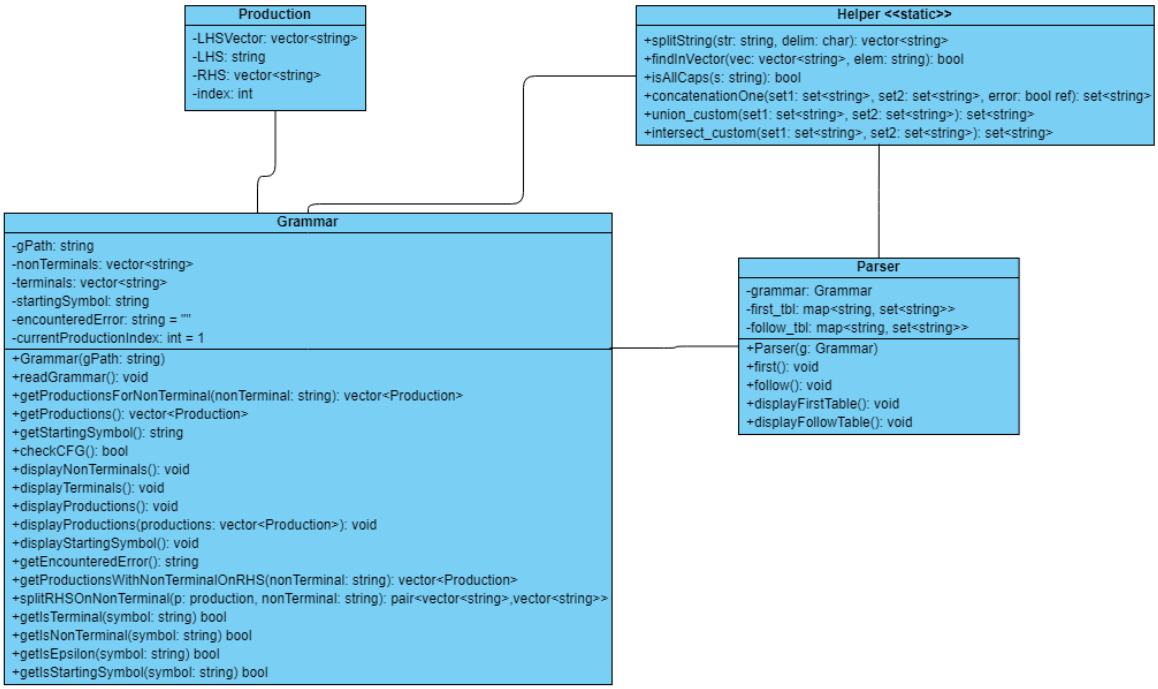
The class Parser keeps only the final version of the *first* and *follow* tables. That means, the steps that lead to the final version of *first* and *follow* are not stored because they are not relevant for the problem.

In order to write a cleaner implementation for the *first* and *follow* methods, some methods were added to the Grammar class. These methods are:

- *getIsTerminal*, *getIsNonTerminal*, *getIsEpsilon*, *getIsStartingSymbol*: these are used for checking whether a given symbol belongs to one of the above-mentioned classes
- *getProductionsWithNonTerminalOnRHS*: this method retrieves a vector of productions containing a given symbol in the right-hand side. This method is used in the *follow* algorithm.
- *splitRHSOnNonTerminal*: this method takes a production and a non-terminal (which must belong to the right-handside of the production). It splits the production into two parts, denoted by alpha and gamma. Alpha is the sequence of symbols before the non-terminal and gamma is the sequence after.

The set operations (union, concatenation of length one) were implented in the Helper class.

The class diagram becomes:



Experiments:

Example 1 (g1.in)

S A B C D
a b c
5
S -> a A b | B A
A -> a A | c A | c
B -> D C
D -> Epsilon | b
C -> c
S

```
[First table ...]
A: a c
B: b c
C: c
D: Epsilon b
S: a b c
a: a
b: b
c: c
```

```
[Follow table ...]
A: b Epsilon
B: a c
C: a c
D: c
S: Epsilon
```

First table:

	F_0	F_1	F_2	$F_3=F_2$
S	a	a	a, b, c	
A	a, c	a, c	a, c	
B	∅	b, c	b, c	
C	c	c	c	
D	ε, b	ε, b	ε, b	

Follow table:

	L_0	L_1	L_2	$L_3=L_2$
S	ε	ε	ε	
A	∅	b, ε	b, ε	
B	∅	a, c	a, c	
C	∅	∅	a, c	
D	∅	c	c	

Example 2 (g2.in)

S A
a b c
2
S -> A a | a
A -> b A | c
S

```
[First table ...]
A: b c
S: a b c
a: a
b: b
c: c
```

```
[Follow table ...]
A: a
S: Epsilon
```

First table:

	F_0	F_1	$F_2=F_1$
S	a	a, b, c	
A	b, c	b, c	

Follow table:

	L_0	L_1	$L_2=L_1$
S	ε	ε	
A	∅	a	

Example 3 (g3.in)

S A B C
a b c d
4
S -> A a | B B
A -> b A | c
B -> C a
C -> d
S

```
[First table ...]
A: b c
B: d
C: d
S: b c d
a: a
b: b
c: c
d: d
```

```
[Follow table ...]
A: a
B: d
C: a
S: Epsilon
```

First table:

	F_0	F_1	F_2	$F_3=F_2$
S	∅	b, c	b, c, d	
A	b, c	b, c	b, c	
B	∅	d	d	
C	d	d	d	

Follow table:

	L_0	L_1	$L_2=L_1$
S	ε	ε	
A	∅	a	
B	∅	d	
C	∅	a	

Example 4 (g4.in)

S B C
a b c d
3
S -> B b | C d
B -> a B | Epsilon
C -> c C | Epsilon
S

```
[First table ...]
B: a Epsilon
C: c Epsilon
S: a b c d
a: a
b: b
c: c
d: d
```

```
[Follow table ...]
B: b
C: d
S: Epsilon
```

First table:

	F_0	F_1	$F_2=F_1$
S	∅	a, b, c, d	
B	a, ε	a, ε	
C	c, ε	c, ε	

Follow table:

	L_0	L_1	$L_2=L_1$
S	ε	ε	
B	∅	b	
C	∅	d	