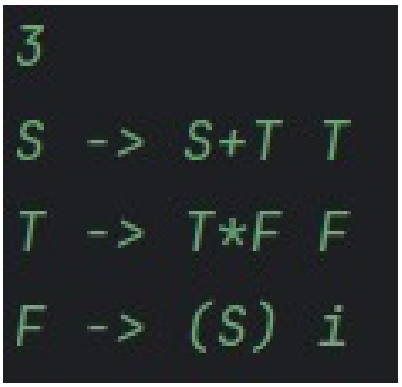
ORAL PRESENTATION COMPLEMENT

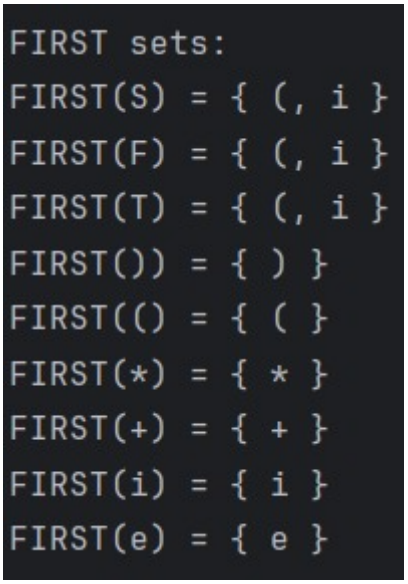
Samuel Calderon Duque & Matias Monsalve Ruiz

This document visually represents the process of the code.

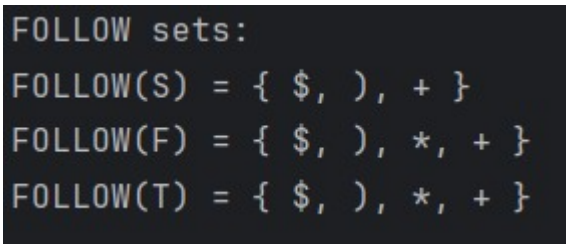
This is how you need to input a grammar.



It calculates the first sets of the grammar.



It calculates the follow sets of the grammar.



It creates the LL(1) table. In this case the grammar is not LL(1) because of the CONFLICTS.

```
LL(1) Table:
M[S, i] = [CONFLICT]
M[S, (] = [CONFLICT]
M[F, (] = (S)
M[F, i] = i
M[T, i] = [CONFLICT]
M[T, (] = [CONFLICT]
```

It creates the action table of the SLR(1).

```
SLR(1) ACTION Table:
ACTION[0, (] = s3
ACTION[0, i] = s5
ACTION[1, $] = rT->F
ACTION[1, )] = rT->F
ACTION[1, *] = rT->F
ACTION[1, +] = rT->F
ACTION[2, $] = rS->T
ACTION[2, )] = rS->T
ACTION[2, *] = s6
ACTION[2, +] = rS->T
ACTION[3, (] = s3
ACTION[3, i] = s5
ACTION[4, $] = acc
ACTION[4, +] = s8
ACTION[5, $] = rF->i
ACTION[5, )] = rF->i
ACTION[5, *] = rF->i
ACTION[5, +] = rF->i
ACTION[6, (] = s3
ACTION[6, i] = s5
```

```
ACTION[7, )] = s10
ACTION[7, +] = s8
ACTION[8, (] = s3
ACTION[8, i] = s5
ACTION[9, $] = rT->T*F
ACTION[9, )] = rT->T*F
ACTION[9, *] = rT->T*F
ACTION[9, +] = rT->T*F
ACTION[10, $] = rF->(S)
ACTION[10, )] = rF->(S)
ACTION[10, *] = rF->(S)
ACTION[10, +] = rF->(S)
ACTION[11, $] = rS->S+T
ACTION[11, )] = rS->S+T
ACTION[11, *] = s6
ACTION[11, +] = rS->S+T
```

It creates the GOTO table of the SLR(1).

```
SLR(1) GOTO Table:
GOTO[0, F] = 1
GOTO[0, S] = 4
GOTO[0, T] = 2
GOTO[3, F] = 1
GOTO[3, S] = 7
GOTO[3, T] = 2
GOTO[6, F] = 9
GOTO[8, F] = 1
GOTO[8, T] = 11
```

As there was no CONFLICT in the creation of the SLR(1) table (which is the ACTION and the GOTO tables), a message appears stating that the grammar is SLR(1).

```
Grammar is SLR(1).
```