Calculating *Follow* for the grammar:

```
Expr -> Term Expr'
Expr' -> + Term Expr' | - Term Expr' | ε
Term -> Factor Term'
Term' -> * Factor Term' | / Factor Term' | ε
Factor -> (Expr) | num | id
```

First set:

| | Expr | Expr' | Term | Term' | Factor |
|---|---|---|---|---|---|
| First | (, id, num | +, -, ε | (, id, num | *, /, ε | (, id, num |

Algorithm for constructing Follow:

1. Put **eof** in Follow(S), where S is the start symbol
2. If there is a production A → αBβ, then everything in First(β) except for ε is placed in Follow(B)
3. If there is a production A → αB, then everything in Follow(A) is in Follow(B)
4. If there is a production A → αBβ where First(β) contains ε, i.e. β =>* ε, then everything in Follow(A) is in Follow(B)

Follow set:

| Rule | Expr | Expr' | Term | Term' | Factor |
|---|---|---|---|---|---|
| 1 | eof | | | | |
| 2 | ) | | +,- | | *,/ |
| 3 | | eof,) | | +,- | |
| 4 | | | eof,) | | eof,),+,- |
| 3 | | | | eof,) | |