**LL(1) Parse Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Given LL(1) grammar =  program ::= expr '\n'  expr ::= term expr\_tail  expr\_tail::= '+' term expr\_tail  | '-' term expr\_tail  | ε  term ::= factor\_p term\_tail  term\_tail::= '\*' factor\_p term\_tail  | '/' factor\_p term\_tail  | ε  factor ::= NUMBER  | '(' expr ')'  factor\_p ::= '-' factor  | factor | Construct a table M[A,a] for A in N, a in T     |  |  |  | | --- | --- | --- | | **A -> α** | **FIRST(α)** | **FOLLOW(A)** | | program -> expr '\n' | ( - NUMBER | \n $ | | expr -> term expr\_tail | ( - NUMBER | \n ) | | expr\_tail -> + term expr\_tail | + | \n ) | | expr\_tail -> - term expr\_tail | - | \n ) | | expr\_tail -> ε | ε | \n ) | | term -> factor\_p term\_tail | NUMBER – ( | + - ) \n | | term\_tail -> \* factor\_p term\_tail | \* | + - ) \n | | term\_tail -> / factor\_p term\_tail | / | + - ) \n | | term\_tail -> ε | ε | + - ) \n | | factor -> NUMBER | NUMBER | \* / + - ) \n | | factor -> ( expr ) | ( | \* / + - ) \n | | factor\_p -> - factor | - | \* / + - ) \n | | factor\_p -> factor | NUMBER ( | \* / + - ) \n | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | NUMBER | ( | ) | + | - | / | \* | \n |
| program | program -> expr '\n' | program -> expr '\n' |  |  | program -> expr '\n' |  |  |  |
| expr | expr -> term expr\_tail | expr -> term expr\_tail |  |  | expr -> term expr\_tail |  |  |  |
| expr\_tail |  |  | expr\_tail -> ε | expr\_tail -> + term expr\_tail | expr\_tail -> - term expr\_tail |  |  | expr\_tail -> ε |
| term | term -> factor\_p term\_tail | term -> factor\_p term\_tail |  |  | term -> factor\_p term\_tail |  |  |  |
| term\_tail |  |  | term\_tail -> ε | term\_tail -> ε | term\_tail -> ε | term\_tail -> / factor\_p term\_tail | term\_tail -> \* factor\_p term\_tail | term\_tail -> ε |
| factor | factor -> NUMBER | factor -> ( expr ) |  |  |  |  |  |  |
| factor\_p | factor\_p -> factor | factor\_p -> factor |  |  | factor\_p -> - factor |  |  |  |

Parse Steps for (1+2)\*3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Token | Production Applied | Parse Stack | Action |
| 0 |  |  | [program] |  |
| 1 | ( | program -> expr '\n' | [\n expr] |  |
| 2 | ( | expr -> term expr\_tail | [\n expr\_tail term] |  |
| 3 | ( | term -> factor\_p term\_tail | [\n expr\_tail term\_tail factor\_p] |  |
| 4 | ( | factor\_p -> factor | [\n expr\_tail term\_tail factor] |  |
| 5 | ( | factor -> ( expr ) | [\n expr\_tail term\_tail ) expr (] | POP ( |
| 6 | NUMBER,1 | expr -> term expr\_tail | [\n expr\_tail term\_tail ) expr\_tail term] |  |
| 7 | NUMBER,1 | term -> factor\_p term\_tail | [\n expr\_tail term\_tail ) expr\_tail term\_tail factor\_p] |  |
| 8 | NUMBER,1 | factor\_p -> factor | [\n expr\_tail term\_tail ) expr\_tail term\_tail factor] |  |
| 9 | NUMBER,1 | factor -> NUMBER | [\n expr\_tail term\_tail ) expr\_tail term\_tail NUMBER] | POP NUMBER |
| 10 | + | term\_tail -> ε | [\n expr\_tail term\_tail ) expr\_tail] |  |
| 11 | + | expr\_tail -> + term expr\_tail | [\n expr\_tail term\_tail ) expr\_tail term +] | POP + |
| 12 | NUMBER,2 | term -> factor\_p term\_tail | [\n expr\_tail term\_tail ) expr\_tail term\_tail factor\_p] |  |
| 13 | NUMBER,2 | factor\_p -> factor | [\n expr\_tail term\_tail ) expr\_tail term\_tail factor] |  |
| 14 | NUMBER,2 | factor -> NUMBER | [\n expr\_tail term\_tail ) expr\_tail term\_tail NUMBER] | POP NUMBER |
| 15 | ) | term\_tail -> ε | [\n expr\_tail term\_tail ) expr\_tail] |  |
| 16 | ) | term\_tail -> ε | [\n expr\_tail term\_tail )] | POP ) |
| 17 | \* | term\_tail -> \* factor\_p term\_tail | [\n expr\_tail term\_tail factor\_p \*] | POP \* |
| 18 | NUMBER,3 | factor\_p -> factor | [\n expr\_tail term\_tail factor] |  |
| 19 | NUMBER,3 | factor -> NUMBER | [\n expr\_tail term\_tail NUMBER] | POP NUMBER |
| 20 | \n | term\_tail -> ε | [\n expr\_tail] |  |
| 21 | \n | expr\_tail -> ε | [\n] | POP \n; ACCEPT |