Context-free Grammar Expression

BNF and BNF w/o Left Recursion for LL(1) Parser

Compiler Theory - Programming Project 3 : Recursive Descent Parser

21400646 Lim Chae Eon

# BNF Expression

*Italic word : Non-Terminal*, **blue colored word : Terminal,** Yellow box : left recursion occured or left factored

|  |  |
| --- | --- |
| **1** | **# Script Definition part** |
| **2** | *Program* → **<script\_start>** *stmt-sequence* **<script\_end>** |
| **3** | *stmt-sequence* → *statement* **;** *stmt-sequence* | comment **nextLine** | |
| **4** | *statement* → *if-stmt | declaration-stmt* | *loop-stmt* | *assign-stmt*  | *function-stmt* | *switch-stmt* | *increment-stmt*  *|* **break** |
| **5** | *simple-stmt* → *stmt-sequence* | **{** *stmt-sequence* **}** |
| **6** |  |
| **7** | **# If-Conditional Statement Part** |
| **8** | *if-stmt*  → **if** *function-parameter* *simple-stmt*  | **if** *function-parameter* *simple-stmt* **else** *simple-stmt* |
| **9** |  |
| **10** | **# Variable Declaration Statement Part** |
| **11** | *declaration-stmt* *→* **var** *id* |
| **12** | *id* → *assign-stmt* | *id* **,** *id* | **IDENTIFIER** |
| **13** |  |
| **14** | **# Assignment Definition Statement Part** |
| **15** | *assign-stmt* → **IDENTIFIER***assign-op exp* |
| **16** | *assign-op* → **=** | |**+=** |
| **17** |  |
| **18** | **# Operation Expression Part** |
| **19** | *exp* → *simple-exp logic-op simple-exp | simple-exp* |
| **20** | *logic-op*→ **<** |**>** |**<=** | **>=** | **==** | **!=** |
| **21** | *simple-exp* → *simple-exp add-op term* | *term* |
| **22** | *add-op* → **+** | |
| **23** | *term*  → *term mul-op factor* | *factor* |
| **24** | *mul-op* →**\***| **/** |
| **25** | *factor* → **(** *exp* **)** | **NUMBER** | **IDENTIFIER** |
| **26** |  |
| **27** | **# Loop Statement Part** |
| **28** | *loop-stmt* → **for** *function-parameter* *simple-stmt*   | **while** *function-parameter* *simple-stmt* |
| **29** |  |
| **30** | **# Function Statement Part** |
| **31** | *function-stmt* → *function-keyword* *function-parameter* |
| **32** | *function-keyword* → **window.prompt** | **window** | **parseFloat**   | **document.writeIn** | **document.write** | **document** |
| **33** |  |
| **34** | **# Increment/Decrement Operation Part** |
| **35** | *increment-stmt* → *increment-op* **IDENTIFIER** | **IDENTIFIER** *increment-op* |
| **36** | *increment-op* → **++** | |
| **37** |  |
| **38** | **# Switch-Conditional Statement Part** |
| **39** | *switch-stmt* → **switch** *function-parameter* **{** *case-part**default-block* **}** |
| **40** | *case-part* → *case-block* *case-part* | |
| **41** | *case-block* → *case-condition* **:** *stmt-sequence* |
| **42** | *case-condition* *→*  **case** *case-parameter* |
| **43** | *case-parameter* *→*  **NUMBER** | **LITERAL** |
| **44** | *default-block* → **default :** *stmt-sequence* | |
| **45** |  |
| **46** | **# Function Parameter Handling Part** |
| **47** | *function-parameter* → **(** *exp* **)** | **(** *exp* **;** *exp* **;** *exp* **)** | **(** **LITERAL****)** |
| **48** |  |
| **49** | **# Comment Part** |
| **50** | *comment* → **// anything EOL\***  \*EOL : End of Line |

# BNF Expression without Left-Recursion

*Italic word : Non-Terminal*, **blue colored word : Terminal,** Green box : Resolved left recursion or factored

|  |  |
| --- | --- |
| **1** | **# Script Definition part** |
| **2** | *Program* → **<script\_start>** *stmt-sequence* **<script\_end>** |
| **3** | *stmt-sequence* → *statement* **;** *stmt-sequence* | *comment* **nextLine**| |
| **4** | *statement* → *if-stmt | declaration-stmt* | *loop-stmt* | *assign-stmt*  | *function-stmt* | *switch-stmt* | *increment-stmt*  *|* **break** |
| **5** | *simple-stmt* → *stmt-sequence* | **{** *stmt-sequence* **}** |
| **6** |  |
| **7** | **# If-Conditional Statement Part** |
| **8** | *if-stmt*  → **if** *function-parameter* *simple-stmt else-part* |
| **9** | *else-part*  → **else** *simple-stmt |* |
| **10** |  |
| **11** | **# Variable Declaration Statement Part** |
| **12** | *declaration-stmt* *→* **var** *id* |
| **13** | *id* → *assign-stmt* id2 | **IDENTIFIER** id2 |
| **14** | *id2* → | **,** idid2 |
| **15** |  |
| **16** | **# Assignment Definition Statement Part** |
| **17** | *assign-stmt* → **IDENTIFIER***assign-op exp* |
| **18** | *assign-op* → **=** | |**+=** |
| **19** |  |
| **20** | **# Operation Expression Part** |
| **21** | *exp* → *simple-exp logic-exp* |
| **22** | *logic-exp* → *logic-op simple-exp* | |
| **23** | *logic-op*→ **<** |**>** |**<=** | **>=** | **==** | **!=** |
| **24** | *simple-exp* → term simple-exp2 |
| **25** | *simple-exp2*  *→* add-opterm simple-exp2| |
| **26** | *add-op* → **+** | |
| **27** | *term*  → *factor term2* |
| **28** | *term2*  → *mul-op factor term2* | |
| **29** | *mul-op* →**\***| **/** |
| **30** | *factor* → **(** *exp* **)** | **NUMBER** | **IDENTIFIER** |
| **31** |  |
| **32** | **# Loop Statement Part** |
| **33** | *loop-stmt* → **for** *function-parameter* *simple-stmt*   | **while** *function-parameter* *simple-stmt* |
| **34** |  |
| **35** | **# Function Statement Part** |
| **36** | *function-stmt* → *function-keyword* *function-parameter* |
| **37** | *function-keyword* → **window.prompt** | **window** | **parseFloat**   | **document.writeIn** | **document.write** | **document** |
| **38** |  |
| **39** | **# Increment/Decrement Operation Part** |
| **40** | *increment-stmt* → *increment-op* **IDENTIFIER** | **IDENTIFIER** *increment-op* |
| **41** | *increment-op* → **++** | |
| **42** |  |
| **43** | **# Switch-Conditional Statement Part** |
| **44** | *switch-stmt* → **switch** *function-parameter* **{** *case-part**default-block* **}** |
| **45** | *case-part* → *case-block* *case-part* | case-block |
| **46** | *case-block* → *case-condition* **:** *stmt-sequence |* |
| **47** | *case-condition* *→*  **case** *case-parameter* |
| **48** | *case-parameter* *→*  **NUMBER** | **LITERAL** |
| **49** | *default-block* → **default :** *stmt-sequence* | |
| **50** |  |
| **51** | **# Function Parameter Handling Part** |
| **52** | *function-parameter* → **(** *parameter-part* **)** |
| **53** | *paramter-part* → *exp* condition-part | **LITERAL** |
| **54** | *condition-part* → **;** *exp condition-part* | |
| **55** |  |
| **56** | **# Comment Part** |
| **57** | *comment* → **// anything EOL\***  \*EOL : End of Line |