Left-Recursion free grammer

program → declaration-list | comment| include_command

declaration-list → declaration declaration-list-cont

declaration-list-cont → declaration declaration-list-cont | **ε**

declaration → type-specifier **ID** declaration-cont

declaration-cont → **;** | **(** params **)** compound-stmt

type-specifier → **Iow | SIow | Chlo | Chain | Iowf | SIowf | Worthless**

params → params-list | **ε**

params-list → param params-list-cont

params-list-cont → **,** param param-list-cont | **ε**

param → type-specifier **IDENTIFIER**

compound-stmt → **{** compound-stmt-cont

compound-stmt-cont → comment local-declarations statement-list **} |** local-declarations statement-list **}**

local-declarations → declaration local-declarations-cont | **ε**

local-declarations-cont → declaration local-declarations-cont | **ε**

statement-list → statement statement-list-cont | **ε**

statement-list-cont → statement statement-list-cont | **ε**

statement → expression-stmt | compound-stmt | selection-stmt | iteration-stmt | jump-stmt

expression-stmt → expression **;** | **;**

selection-stmt → **if (** expression **) {** statement **}** selection-stmt-cont

selection-stmt-cont → **else {** statement **}** | **ε**

iteration-stmt → Loop-statement | Iterate-statement

Loop-statement → **Loopwhen (** expression **)** statement

Iterate -statement → **Iteratewhen (** expression **;** expression **;** expression **)** statement

jump-stmt → **Turnback** expression **; | Stop ;**

expression → **[** id-assign expression-cont | simple-expression

expression-cont → **=** expression **]** | **ε**

simple-expression → additive-expression simple-expression-cont

simple-expression-cont → relop additive-expression | **ε**

relop → **<= | < | > | >= | == | != | && | ||**

additive-expression → term additive-expression-cont

additive-expression-cont → addop term additive-expression-cont | **ε**

addop → **+** | **-**

term → factor term-cont

term-cont → mulop factor term-cont | **ε**

mulop → ***** | **/**

factor → **(** expression **)** | id-assign | num

id-assign → **IDENTIFIER** id-call

id-call → **(** args **) | ε**

args → arg-list | **ε**

arg-list → expression arg-list-cont

arg-list-cont → **,** expression arg-list-cont | **ε**

num → Signed-num |Unsigned-num

Unsigned-num → value

Signed-num → pos-num | neg-num

pos-num → **+** value

neg-num → **-** value

value → **CONSTANT**

comment → **/* STR */** | **/// STR**

include_command → **include (** F_name **.txt) ;**

F_name →**STR**

First function

First(program) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless** }

First(declaration-list) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless** }

First(declaration) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless** }

First(type-specifier) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless** }

First(comment) = { **/$**, **$$$** }

First(include-command) = { **Include** }

First(declaration-list-cont) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless**, **ε** }

First(declaration-cont) = { **;**, **(** }

First(params) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless**, **ε** }

First(params-list) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless** }

First(param) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless** }

First(params-list-cont) = { **,**, **ε** }

First(compound-stmt) = { **{** }

First(compound-stmt-cont) = { **/$**, **$$$**, **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless**, **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

First(local-declarations) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless**, **ε** }

First(statements-list) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}**, **ε** }

First(statement) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

First(expression-stmt) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;** }

First(expression) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[** }

First(simple-expression) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(** }

First(additive-expression) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(** }

First(term) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(** }

First(factor) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(** }

First(id-assign) = { **IDENTIFIER** }

First(num) = { **CONSTANT**, **+**, **-** }

First(signed-num) = { **+**, **-** }

First(pos-num) = { **+** }

First(neg-num) = { **-** }

First(unsigned-num) = { **CONSTANT**, **FLOAT_NUM** }

First(value) = { **CONSTANT**, **FLOAT_NUM** }

First(selection-stmt) = { **if** }

First(jump-stmt) = { **Turnback**, **Stop** }

First(iteration-stmt) = { **Loopwhen**, **Iteratewhen** }

First(loop-statement) = { **Loopwhen** }

First(iterate-statement) = { **Iteratewhen** }

First(local-declaration-cont) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless**, **ε** }

First(statement-list-cont) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}**, **ε** }

First(selection-stmt-cont) = { **else**, **ε** }

First(expression-cont) = { **=**, **ε** }

First(simple-expression-cont) = { **<=**, **<**, **>**, **>=**, **==**, **!=** , **&&**, **||**, **ε** }

First(relop) = { **<=**, **<**, **>**, **>=**, **==**, **!=** , **&&**, **||** }

First(additive-expression-cont) = { **+**, **-**, **ε** }

First(addop) = { **+**, **-** }

First(term-cont) = { *****, **/**, **ε** }

First(mulop) = { *****, **/** }

First(id-call) = { **(**, **ε** }

First(args) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **ε** }

First(arg-list) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[** }

First(arg-list-cont) = { **,**, **ε** }

First(f_name) = { **STR** }

Follow function

Follow(program) = { **$** }

Follow(declaration-list-cont) = { **$** }

Follow(declaration-list) = { **$** }

Follow(params) = { **)** }

Follow(params-list-cont) = { **)** }

Follow(params-list) = { **)** }

Follow(compound-stmt-cont) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless**, **$**, **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}**, **else** }

Follow(compound-stmt) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless**, **$**, **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}**, **else** }

Follow(declaration-cont) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless**, **$**, **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(declaration) = { **Iow**, **Siow**, **Chlo**, **Chain**, **Iowf**, **Siowf**, **worthless**, **$**, **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(local-declarations) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(local-declarations-cont) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(statement) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(statement-list) = { **}** }

Follow(statement-list-cont) = { **}** }

Follow(selection-stmt) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(selection-stmt-cont) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(loop-statement) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(iteration-stmt) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(iterate-statement) = { **IDENTIFIER**, **CONSTANT**, **+**, **-**, **(**, **[**, **;**, **if**, **Loopwhen**, **Iteratewhen**, **Turnback**, **Stop**, **{**, **}** }

Follow(expression-cont) = { **;**, **)**, **]**, **,**, }

Follow(expression) = { **;**, **)**, **]**, **,**, }

Follow(arg-list) = { **)** }

Follow(args) = { **)** }

Follow(arg-list-cont) = { **)** }

Follow(simple-expression-cont) = { **;**, **)**, **]**, **,**, }

Follow(simple-expression) = { **;**, **)**, **]**, **,**, }

Follow(additive-expression-cont) = { **<=**, **<**, **>**, **>=**, **==**, **!=** , **&&**, **||**, **;**, **)**, **]**, **,**, }

Follow(additive-expression) = { **<=**, **<**, **>**, **>=**, **==**, **!=** , **&&**, **||**, **;**, **)**, **]**, **,**, }

Follow(term-cont) = { **+**, **-**, **<=**, **<**, **>**, **>=**, **==**, **!=** , **&&**, **||**, **;**, **)**, **]**, **,**, }

Follow(term) = { **+**, **-**, **<=**, **<**, **>**, **>=**, **==**, **!=** , **&&**, **||**, **;**, **)**, **]**, **,**, }

Follow(id-call) = { **=**, **;**, **)**, **]**, **,**, ***, **/**, **+**, **-**, **<=**, **<**, **>**, **>=**, **==**, **!=** , **&&**, **||** }

Follow(id-assign) = { **=**, **;**, **)**, **]**, **,**, ***, **/**, **+**, **-**, **<=**, **<**, **>**, **>=**, **==**, **!=** , **&&**, **||** }

Follow(factor) = { ***, **/**, **+**, **-**, **<=**, **<**, **>**, **>=**, **==**, **!=** , **&&**, **||**, **;**, **)**, **]**, **,** }