Grammar

NOTE: Terminals will be bold before it

The deleted rules are 6 ,11,14,17,27,28,29.

Original Grammar

Symbol	Definition	Needs Modification
program	*type-specifier* **main** **(** *declaration-list* **)** {*declaration-list statement-list*}	
declaration-list	*declaration-list declaration* | *declaration*	YES (Left-Rec.)
declaration	*var-declaration*	
var-declaration	*type-specifier* **ID** **;** | *type-specifier* **ID** [**NUM**]	YES (factoring)
type-specifier	**int** | **float**	
DELETED 6	DELETED	
params	*param-list* | *void*	
param-list	*param-list* **,** *param* | *param*	YES (Left-Rec.)
param	*type-specifier* **ID** | *type-specifier* **ID** [ ]	YES (factoring)
compound-stmt	{*statement-list*}	
DELETED 11	DELETED	
statement-list	*statement-list statement* | **empty**	YES (factoring)
statement	*assignment-stmt* | *compound-stmt* | *selection-stmt* | *iteration-stmt*	
DELETED 14	DELETED	
selection-statement	**if** **(** *expression* **)** *statement* | **if** **(** *expression* **)** *statement* **else** *statement*	YES (Left-Rec.)

Symbol	Definition	Needs Modification
iteration-statement	**while** (*expression*) *statement*	
DELETED 17	DELETED	
assignment-stmt	*var* = *expression*	
var	**ID** | **ID** [*expression*]	YES (factoring)
expression	*expression relop additive-expression* | *additive-expression*	YES (Left-Rec.)
relop	<= | < | > | >= | == | **!=**	YES (factoring)
additive-expression	*additive-expression addop term* | *addop*	YES (Left-Rec.)
addop	+ | **-**	
term	*term mulop factor* | *factor*	YES (Left-Rec.)
mulop	***** | /	
factor	(*expression*) | *var* | **NUM**	
DELETED 27, 28, 29	DELETED	

Modified Grammar(LL1)

Symbol	Definition
program	*type-specifier* **main** (*declaration-list*) {*declaration-list statement-list*}
declaration-list	*declaration declaration-list-Tail*
declaration-list-Tail	*declaration declaration-list-Tail* | **empty**
declaration	*var-declaration*
var-declaration	*type-specifier* **ID** *var-declaration-Tail*
var-declaration-Tail	**;** | [**NUM**] **;**

Symbol	Definition
type-specifier	**int** | **float**
DELETED 6	DELETED
params	*param-list* | **void**
param-list	*param param-list-tail*
param-list-Tail	**,** *param param-list* | **empty**
param	*type-specifier* **ID** *param-Tail*
param-Tail	**empty** | [ ]
compound-stmt	{*statement-list*}
DELETED 11	DELETED
statement-list	*statement-list-Tail*
statement-list-Tail	*statement statement-list-Tail* | **empty**
statement	*assignment-stmt* | *compound-stmt* | *selection-stmt* | *iteration-stmt*
DELETED 14	DELETED
selection-statement	**if** (*expression*) *statement selection-statement-Tail*
selection-statement-Tail	**empty** | **else** *statement*
iteration-statement	**while** (*expression*) *statement*
DELETED 17	DELETED
assignment-stmt	*var* = *expression*;
var	**ID** *var-Tail*
var-Tail	**empty** | [*expression*]
expression	*additive-expression expression-Tail*
expression-Tail	*relop additive-expression expression-Tail* | **empty**
relop	< *relop-Tail* | > *relop-Tail* | == | **!=**
relop-Tail	= | **empty**
additive-expression	*term additive-expression-Tail*
additive-expression-Tail	*addop term additive-expression-Tail* | **empty**
addop	+ | **-**
term	*factor term-Tail*
term-Tail	*mulop factor term-Tail* | **empty**
mulop	***** | /
factor	(*expression*) | *var* | **NUM**
DELETED 27, 28, 29	DELETED

Implementation Issues and decisions:

1. *program* rule
    1. **problem**: The rule didn't work correctly because the definition of *declaration-list* couldn't be used to read the function parameters
    2. **decision**: replaced *declaration-list* rule with *params* rule.
2. *assignment-stmt*
    1. **problem**: assignment statements needed to be terminated with a semicolon.
    2. **decision**: added semicolon to assignment-stmt(rule 18).

Handling syntax error:

To handle syntax error we kept track of all the tokens read and their line number. On encountering an error, the program terminates after printing the token expected, the token found, and the line number at which the error happened.

Sample run

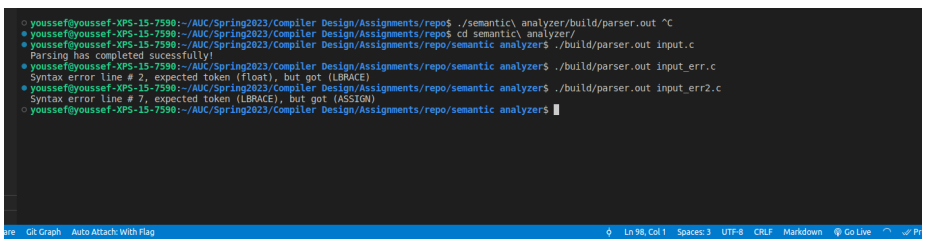

Figure 1: sample run