

Author: Nghia Duc Hong

Student Number: 292119

Email: [duc.hong@tuni.fi](mailto:duc.hong@tuni.fi)

### 1. Syntactic analysis:

After all the tokens have been constructed, the syntactic analyzer seeks to construct a derivation tree. The syntactic tree would define the grammar for the language we are building on. In the tree, each leaf corresponding to a token from the list. Moreover, the leaves must form a meaningful phrase in the language. The node other than the leaves can be understood with grammar defined by the subnode of this node. Based on this hierarchical tree, the syntax of the input string can be defined.

### 2. How is the syntactic structure of the language expressed in the PLY tool.

In the PLY tool, grammar rules are defined by a Python function where docstring to that function contains the appropriate context-free grammar specification. The detected grammar can be recognized by a sequence containing the values of each grammar symbol in the corresponding rule.

### 3. Brief explanation for rules:

- Sheet variable definition:

A sheet definition is start with a keyword sheet, then the identifier of the sheet, followed by the optional for sheet initialization ( sheet\_init). The sheet initialization is started with an equal symbol, then either a sheet\_init\_list or an int, a symbol star \*, and an int. The sheet\_init\_list is a list which is bounded by curly brackets, containing sheet\_row(s).

- Function call:

A function call would require a function identifier, followed by list of arguments bounded by square brackets. The arguments are the arg(ument)\_expression, separated by commas between them.

- Sheet variable definition with initialization list:

A sheet definition with initialization list would contain keyword and sheet identifier, initialized by a equal symbol and sheet\_row(s) bounded by curly brackets. Every sheet row is consequence simple expression separated by commas.

### 4. Answer the questions :

- It is not possible to define a new function inside another function. The reason is that the syntax of the definition of the function only accepts only variable definitions and statements, therefore, the function definition inside another function would not be recognized.
- It is not possible to recognize to perform arithmetic with integers as there is no rule defining the integers with plus/minus/multiply/divide arithmetically.
- $2.0 + 3.0$  should be recognized as scalar\_expr or simple\_expr. And the range\_expr grammar doesnot include scalar\_expr or simple\_expr so it is not possible to initialize a range with range initialization =  $2.0 + 3.0$

- --xx is not recognized as xx is an identifier, xx is an atom and -xx would be a factor/term. Then there is no rule defining the MINUS factor or MINUS term. But xx--yy is still be recognized as mentioned above, -yy is a factor, then would be a factor and a term; xx is an atom then a factor then a term. There exist rule defining simple\_expr to be term MINUS term then xx—yy belong to a simple\_expr as it follows the syntactic rules.
- It is not possible to initialize the list like that because the sheet\_init\_list part does not match the rules, the rule says it should be sheet\_row(s) inside curly brackets, not a scalar\_expr.
- The rules defining term, factor, and simple\_expr are defined to make sure the precedence that the multiplication and division are recognized first, as the terms always capture the multiplication and division first, then the simple\_expr then recognizes the addition and subtraction with those terms.
- In sheetScript there is no semicolons or linebreaks, but by defining the rule, the statements and rules often begin with keywords, even in variable definition, so these would be also the pattern for recognizing a command begin.

## 5. Describing the implementation:

The syntax work well and give result exactly as those tests. The first version provides some extra line as the assignments and statement are the same. After eliminating one of this two, I have the results which are identical with the outputs.

No extra also ( I don't know what should be done as extra in this phase)

6. The assignment is not difficult in my opinion. The syntax rules are not so bulky to implement. The most annoying part of the phase is that we were coding and did not know if we are on the right track or not. The testing also difficult as you don't know what is wrong with the tree. So it did bring some anxiety to coder to test and to fix the errors there. I struggled thinking why this line was printed out but this line was not. In this phase, the tree might be correct but the way we print the results might be different then we don't know if it valid or not. Finally, I still managed to fix bugs and error myself.