Project work – Phase 2

Principles of programming languages

Pinja Mikkonen 99219

1. *What is syntax analysis and how is it related to other parts in compilation?*

   Syntax analysis is a part of code compilation process where the syntax of a programs gets analyzed. Syntax analyzer checks that a program follows the syntax rules of a language and gives an error message if the syntax is incorrect. Syntax analyzer works as an important step between lexical analysis and the later steps of compilation process.

2. *How is the syntactic structure of the language expressed in the PLY tool? I.e., what parts are needed in the code and how are they related to syntactic rules of the language?*

   Syntactic structure of a language is expressed in PLY tool using a list of reserved words and tokens gained from the lexer tool. A single rule is, in essence, a group of tokens in a certain order and the variants of that order. PLY tool matches lines of code to syntax rules it's given and issues an error message if a group of tokens match no rule.

3. *Explain in English what the syntax of the following elements mean (i.e. how would you describe the syntax in textual form):*

   1. *Sheet variable definition*

      Sheet variable definition is the definition of a two-dimensional sheet of values. The definition of sheet is comprised of the keyword sheet, the sheet's name and then either a list of values or the dimensions of the sheet.

   2. *Function call*

      A function call is the process of calling a previously defined function. It's comprised of a function identifier and a list of arguments the function is given.

   3. *Sheet variable definition with initialization list ("{…}").*

      Sheet variable definition with initialization list initializes a two-dimensional sheet. A row is comprised of several decimal values separated by commas. There can be multiple rows.

4. *Answer the following based on the syntax definition:*

   1. *Is it possible to define a "nested" function, i.e. to define a new function inside another function? Why?*

      No. This is because a function definition can include variable definitions and a statement list, and a function definition is defined as neither of these.

   2. *Is it syntactically possible to perform arithmetic with integers (1+2)? Why?*

No. This is because all arithmetic is performed with atoms, which can be either identifiers or decimal numbers. Integers are utilized in other fashions.

3. *Is it syntactically possible to initialize a range variable with a decimal value (range _rng = 2.0+3.0)? Why?*

   No. Range variables are initialized with either cell references or integer ranges, neither of which can be a decimal value.

4. *Are the following allowed by the syntax: xx--yy and --xx? Why?*

   The first example is allowed, second is not. This is because arithmetic is performed with factors, which can be comprised of a positive or a negative atom. So a `MINUS atom` can be a factor, followed by `factor MINUS factor` -statement. On the other hand there is nothing that allows for a factor to have a minus sign in front of it.

5. *Can comparisons appear in a sheet variable's initialization list (sheet SS = { 1.0 < 2.0 })? Why?*

   No. This is because comparisons are scalar expressions, whereas sheet rows can include only simple expressions. Simple expressions are scalar expressions, but not the other way around.

6. *How is it ensured that addition/subtraction are done after multiplication/division?*

   Multiplication and division are defined as terms, whereas addition and subtraction are simple expressions. Terms can be simple expressions, but not the other way around. This ensures that multiplication and division are handled first.

7. *In SheetScript, statements and definitions are not separated by semicolons (like in Java/C++) or line breaks (like in Python). How does the syntax known when one thing ends and another begins?*

   The syntax knows where one thing ends and another begins because of syntax rules. Sheetscript has strict syntax rules that determine the symbols or keywords that start and end certain parts of code. Syntax analysis uses these to determine where things end and start.

5. *Please mention in the document if you didn't implement functions (i.e. you are ok with passing with the minimum grade).*

   Functions have been implemented and I aim to complete all four phases. :)

6. *What did you think of this assignment? What was difficult? What was easy? Did you learn anything useful?*

   This assignment wasn't too difficult. Syntax checker was mostly easy to complete, most problems came from finding errors and determining their cause correctly. It was a satisfying exercise that demonstrated the process of syntax checking very well.

Lexer was also modified for phase 2 due to the feedback given on phase 1. The following modifications have been made:

1. Integers and decimal numbers now return the value.
2. The definition of `range_ident` tokens have been modified.
3. Some reserved word checks have been removed.
4. A `def_remove_comment()` -function for removing comments from inside info strings has been implemented.