HW2 – CS 341 Programming Languages

151044012 – Uğurkan Ateş

Chapter 3 – Review Question 14

14. Why can machine languages not be used to define statements in operational semantics?

Answer :

There are three reasons for this situtation. First is that for Operational semantics we need appropriate language to do it. Machine language is not easily understandable and readable.

Second is that if ignored first problem and tried anyway in machine language , steps while execution of Machine Language and resulting changes to state of machine would be too numerous.(Hard to check in control when they are too small and too numerous).

Third is that because of second point there would be too many complex memory instructions , thus making hard to process (reading and writing from computer memory). Which would result slowness in a network.

Chapter 3 – Problem Set 19

syntax r : <assigment> -> <variable> = <expression>

<expression> -> <variable>[2] + <variable>[3]

<variable>[2].actual_type = <variable>[3].actual_type

<expression> -> <variable>

<variable> -> A | B | C

semantic r: <variable>.actual_type <- lookup(<variable>.string)

Chapter 4 - Review Question 1

Answer :

Reasons to use Backus Naur Form (BNF) can be described as ;

- It gives us a clear and brief information about syntax description
- You can build parser to be directly on BNF. Which makes easy to understand how parser works if you are familiar with BNF. This helps both for parser developer and programming language implementors.
- If you defined your parsed based on BNF they are easy to main cause of less complex approaches can be used for while design it. When you want to change or fix some part its easily can be done due nature of BNF.

Chapter 4- Review Question 5

Describe briefly the three approaches to building a lexical analyzer.

Answer :

- Write a formal description of token patterns of language using a descriptive language related to REGEX(regular expressions).
- Create a state – transition diagram which describes token patterns of language and implement a program based on that diagram.
- Create a state – transition diagram which describes token patterns of language and hand made table driven implementation of that state diagram.