CSE 341 – PROGRAMMING LANGUAGES – HOMEWORK 2

CHAPTER 3

Review Qestion 14 : Why can machine language not be used to define statements in operational semantics ?

Answer:

Because using machine language to define statements in operational semantics cause some problems.
1 – The step in the execution of machine languge and resulting changes to the state of the machine are too small and too numerous.
2 – The storage of computer is too large and comlex.

Problem Set 19 : Write an attribute grammar whose BNF basis is that of Example 3.6 in Section 3.4.5 but whose languages rules are as follows: Data types cannot be mixed in expressions, but assigment statements need not have the same types on both sides of the assignment operator.

Answer:

1. Syntax rule : <assign> → <var> = <expr>

2. Syntax rule : <expr> → <var>[2] + <var>[3]

Predicate : <var>[2].actual_type == <var>[3].actual_type

3. Syntax rule : <expr> → <var>

4. Sytax rule : <var> → A | B | C

Semanti rule : <var>.actual_type ← lookup(<var>.string)

CHAPTER 4

Review Question 1 : What are the reasons why using BNF is adventageus over using an informal syntax description ?

Answer:

Using BNF has at least three adventages.
1 – BNF descriotions of the syntax of the program are clear and concise, both humans and for software system that use them.
2 – BNF description can be used as the direct basis for the sytax analyzer.
3 – İmplementations based on BNF are relatively easy to maintain because of their modularity.

Review Question 5 : Describe birefly the three approaches to building a lexical analyzer?

Answer :

1 . Write formal description of the token patterns of the languages using a descriptive language related regular expressions. These descriptions are used as input to a software tool that automatically generates a lexical analyzer There are many such tools available for this this. The oldest of these, named lex, is commonly used as part of UNIX systems.

2. Design a state transition diagram that describes the token patterns of the language and wirte a program that implements the diagram.

3. Design a state transition diagram that describes the token patterns of the language and hand-construct a table-driven implementation of the state state diagram.

Efkan DURAKLI
161044086