CSE341 - Programming Languages HW2

Chapter 3: Review Questions 14:

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

Answer:

We can not use machine language to define statements in operational semantic because of two major reasons.

The first reason is, when the execution of machine languages, the individual steps and the changes which is depending on the current state of machine, are too small and too numerous.

The second reason is storing in a computer is too large and it makes too complex to storage in a computer.

Chapter 3: Problem set 19:

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

Answer:

```
I replace the second semantic rule with the following rules :
<var>[2].env <- <expr>.env
<var>[3].env <- <expr>.env
<expr>.actual_type <- <var>[2].actual_type
predicate  :  <var>[2].actual_type == <var>[3].actual_type
```

Chapter 4: Review Question 1:

What are the reasons why using BNF is advantageous over using an informal syntax description?

Answer:

Using BNF has many advantages instead of using some informal syntax description. Although it has many advantages there are three important advantages.

Firstly, people can easly understand the BNF description the syntax of program. In other Word the BNF description the syntax of program is more clear and more understandable.

Secondly, people can use directly BNF description for basis syntax analyzer.

The last one is, the implementations, which are based on BNF, can easly uptaded because of thir modularity.

Chapter 4: Review question 5:

Describe briefly the three approaches to building a lexical analyzer.

Answer:

1. First of all you should write a formal description. This formal description has to express patterns of the language with regular expressions. This formal description are used as input a lexical analyzer

2. You should design a state transition diagram. This diagram has to describe the token patterns of the language. And also you should write a program which is implementing the diagram.

3. Lastly you should design a state transition diagram. This diagram has to describe the token patterns of the language. And this diagram has to describe hand-construct a table-driven implementation of the state transition diagram.

ALİ HAYDAR KURBAN 151044058