Lütfullah Türker
141044050

CSE – 342 Programming Languages

HW2

Q1.)

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

A1.)

There are 2 main cause for this problem.

First, changes that occur depending on the state of the machine are too small and too much.

Second, the memory of computers is so large and complicated.

Q2.)

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 cannot be mixed in expressions, but assignment statements need not have the same types on both sides of the assignment operator.

A2.)

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) Syntax rule: <var> → A | B | C

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

Lütfullah Türker
141044050

Q3.)

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

A3.)

Because BNF provides a clear and concise syntax description. And Parsers which based on BNF are easy to maintain. And BNF provides generalized and useful solution.

Q4.)

Describe briefly the three approaches to building a lexical analyzer.

A4.)

1. Writing a description of the token patterns of the language with using a descriptive language related to regular expressions.

2. Designing state transition diagram that describes the token patterns of the language. Writing a program that implements that.

3. Hand-construct a table-driven implementation of the state transition diagram.