CS4XX - INTRODUTION TO COMPILER THEORY

CS4XX - WEEK 5

Reading:

Chapter 5 from Principles of Compiler Design, Alfred V. Aho & Jeffrey D Ullman

Objectives:

1. To understand in detail about syntax directed definitions to construct syntax trees .

2 .To learn the methods for constructing syntax trees.

Concepts:

1. Syntax-directed definitions--- 1/2 hour
2. Construction of syntax trees --- -1/2 hour
3. Bottom-up evaluation of S-attributed definitions ------------------------------1/2 hour
4. L-attributed definitions ---1/2 hour
5. Top down translation--1/2 hour
6. Bottom-up evaluation of inherited attributes-----------------------------------1/2 hour

Outlines :

1. Syntax-Directed Translation

a. Syntax Directed Definitions

Generalization of a context free grammar in which each grammar symbol has an associated set of attributes.

- Form of a Syntax Directed Definition – Explains the grammar production A->α associated with a set of semantic rules.
- Synthesized attributes – Defining S attributed synthesized attributes.

- Inherited attributes – Discussing the importance of inherited attributes in a parse tree.
- Dependency graphs – Defining the dependency graph which shows the interdependency among inherited and synthesized attributes in a parse tree.
- Evaluation Order – Explains how the directed graph is evaluated.

b. Construction of Syntax trees

Explains how syntax-directed definitions can be used to specify the construction of syntax trees and other graphical representations of language constraints.

- Syntax Tree – Defining a syntax tree.
- Constructing syntax tree for expressions – Discussing the construction mechanism of a syntax tree by translating an expression into postfix form.
- A syntax directed definition for constructing syntax trees – Gives the S-attribute definition for constructing a syntax tree.
- Directed a-cyclic graphs for expressions – Directed acyclic graph for an expression identifies the common sub-expressions in the expression.

c. Bottom-up evaluation of S-attributed definitions

Implementation of translators using bottom-up evaluation methods for syntax-directed definitons

- Synthesized attributes on the parser stack – Implementing translator for S-attributed definition using an LR-parser generator.

d. L-Attributed definitions

Explaining the class of syntax-directed definitons, called L-Attributed definitions.

- L-Attributed definitions – Syntax directed definition in terms of L-attributes.
- Translation schemes – Explains how context free grammars are associated with the grammar symbols and semantic actions enclosed between braces.

e. Top down translation

Explains implementation of L-attributed definitions during predictive parsing.

- Eliminating Left Recursion from a translation scheme – Discussing the use of left recursive grammars for expressions.
- Design of a predictive translator – Generalizing the construction of predictive parsers to implement a translation scheme based on a grammar using top-down parsing.

f. Bottom-up evaluation of inherited attributes

Discuss the method to implement L-attributed definitions in the framework of bottom-up parsing.

- Removing Embedding actions from translation schemes – Explains the techniques for removing embedding actions from translation schemes.
- Inheriting attributes on the parser stack – Determines how the attributes are inherited on the parser stack.
- Simulating the evaluation of inherited attributes – Explains the simulation methodologies used for evaluating the inherited attributes.
- Replacing inherited by synthesized attributes – Explains the solution for problem of translation using parsing by restructuring the grammar.
- A difficult syntax directed definition – Discusses the algorithm for implementing the inherited attributes during bottom-up parsing.