Syntax-Directed Definitions and Attribute Evaluation

66.648 Compiler Design Lecture (02/19/97)

Computer Science

Rensselaer Polytechnic

Lecture Outline

- Syntax directed definitions
- Attribute Evaluations
- Administration

Phases of a compiler

We are in chapter 5 of the text book. Please read

sections 5.1-5.9

So far, we have covered lexical analyzer and parser. The output of the parser is a rightmost derivation (parse tree)

Our next step is to study "Semantic Analyzer" which creates an abstract syntax tree with attributes.

Syntax-Directed Definition

Syntax-directed definitions extend context-free grammars by including attributes for each grammar symbol and semantic rules that specify semantic actions to be performed during parsing. In fact one can carry out this within our parser tool.

Attributes

Consider a context-free grammar production of the form

Y--> X_1 X_2…. X_n. Attributes are associated

Attributes - Contd

With each instance of a grammar symbol I.e., each node of the parse tree.

Example attributes could be value, type, font-size, base line etc.

For each separate instance of the grammar symbol, there will be separate instance of the attribute.

Attributes can either be inherited or synthesized.

Attributes - Contd

An attribute of a grammar symbol is synthesized if it is computed from the attributes of its children in the parse tree corresponding to the rightmost derivation.

Example: E --> E + num | num

E2.val = E1.val + num.val

Other examples could be base-line in latex.

An attribute of a grammar symbol is inherited, if it is computed from the attributes of its parents and siblings.

Attributes - Contd

Example:

Declarations--> Type varlist

varlist --> varlist, var | var

int x, y;

var1. Type = int;

var2.. Type = var1. Type = int

The start symbol has no inherited attributes.

Attributes -Contd

The synthesized attribute values for tokens/terminals are provided by the lexical analyzer. Tokens are assumed to have no inherited attributes.

Statement: assignmentstmt

assignmentstmt: var '=' expr ;

The type of expr is inherited from the type of var.

eg. In java the following is illegal:

Iamaninteger = 2.3 + 3.4;

Semantic Rules

A grammar production Y --> X_1 … X_n may have zero or more associated semantic rules. Each semantic rules has the form, b=f(c_1,…,c_k) are attributes of the grammar symbol in the production such that:

1. b is a synthesized attribute and c_1,…,c_k are the attributes of the grammar symbols on the rhs or
2. b is an inherited attribute of one of the RHS grammar symbols and c_1,…,c_k are any other attributes in the production.

Example

A syntax-directed definition in which semantic rules have no side effects is an attribute grammar.

Example of Syntax-Directed Definition

Production	Semantic Rules
S --> E	print(E.val)
E_1 --> E_2+T	E_1.val = E_2.val+T.val
E--> T	E.val =T.val
T_1--> T_2* num	T_1.val=T_2.val*num.val
T --> num	T.val=num.val

Example-Contd

Comment: A semantic rule with a side effect such as print(E.val) can be thought as computing a synthesized attribute for S as S.val = print(E.val);

Evaluation Order of attributes:

Semantic Rules define dependencies among attribute instances, which in turn define legal evaluation order (orders in which an attribute value is computed before it is used).

Evaluation of Attributes

If the parsing order is consistent with some legal evaluation order, then attribute evaluation can be done on-the-fly while parsing without explicitly building a parse tree.

S-attributed Definition

An S-attributed definition is a syntax directed definition that contains only synthesized attributes.

The synthesized attributes can be stored in extra symbols for each symbol on the parse stack.

YACC Examples

Comments and Feedback

Project 2 is out.

We have finished chapter 5. We will do some more examples in the next class. Please keep reading chapter 4 and understand the material. Work out as many exercises as you can.

We move onto chapter 6 next.