***Synthesized* attributes**

As we know, an *attribute grammar* is a context-free grammar to which *attribute* and *semantic functions* have been added:

Consider a production *p* in the set of production *P* :

p : X0  X1 X2 X3. . . Xn n > = 1

For any Xi in the production *p*, there may be finite disjoint sets I(Xi) and S(Xi.) The are the *inherited* attributes and *synthesized* attributes, respectively.

In general, the values of inherited attributes are passed *down* the parse tree. The value of synthesized attributes are passed *up* the parse tree.

The set of all attributes for X i is denoted A(Xi):

A(Xi) = I(Xi) U S(Xi)

An attribute *a* of Xi is denoted Xi.*a* whether it is a reference to attribute *a* in the parse tree or the grammar.

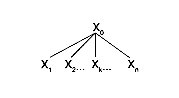


fig 1: attribute *a* in the parse tree

Predefined attribute values are called *intrinsic*. Synthesized, intrinsic attributes of terminals are computed by the lexical analysis phase. Inherited intrinsic attributes of the start symbol are passed as parameters before evaluation begins from the parser.

A production *p* : X0  X1 X2 X3. . . Xn possesses an *attribute occurrence, (a, k),* if Xk is at node #*m* ( in some tree-numbering scheme), then we say Xk possesses an *attribute realization, (a, m),* if Xk has an attribute *a*. *A semantic function*, sometimes called an attribute rule, gives a value to an attribute occurence*(a, k),* in production *p*. Such a function is denoted fp(*a*,k)

The set of values upon which fp(*a*,k)depends is called the *dependency set* for *(a, k)* and is denoted Dp(*a*,k). Thus,

Dp(*a*,k) = { (b, j) | fp(*a*,k) = g (. . .Xj.*b*, . . . ) }

This is read "the value of attribute occurrence *(a, k),* or Xk. *a* depends on the set of attributes { (*b*, j) } or { Xj.*b* } where each Xj.*b* is used in the calculation of Xk. *a*".

Example 1 illustrates these definitions. This example is an adaptation from Knuth (1968) where he first defined attribute grammars.

Thus, *Scale* is an inherited attribute, and *Value* and *Neg* are synthesized attributes. The intrinsic attributes are Sign.*Neg* in productions one and two, BinaryDigit.*Value* in production five, and List.*Scale* in production zero

**Inherited Attributes**

Inherited Attributes: These attributes pass information from root to the leaves of a parse tree, or sideways among siblings.

Attributes are variables to which values are assigned. Each attribute variable is associated with one or more nonterminals or terminals of the grammar.

In the grammar of Section 6.2.2, the attribute *Value* is associated with the nonterminals E, T, and F as well as with the terminal Constant. For nonterminal E, this is written:

* E.*Value*

This notation indicates that nonterminal "E" has an attribute called *Value* attribute names will be italicized.

Although it is possible to evaluate some attributes at parse time, we will asume that all attributes are evaluated after the program has been parsed. Values are assigned to local attributes by equations called semantic functions. Local attributes are those which fall within the scope of a production as it appears in the parse tree. For example:

**Syntax Semantics**

E0  E1 + T E0.*Value* := E1.*Value* + T.*Value*

Here, the production is EE+T. The subscripts are used only to distinguish the two E's. The attribute *Value* is associated with both E's and with T. If this production were in a parse tree, it would be denoted:

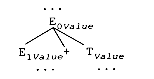


Fig 2 : Attribute *Value* is associated with both E's and with T

The value of the attribute *Value* is passed *up* the parse tree because E1.*Value* and T.*Value* are used to compute E0.*Value*. Such attributes are termed *synthesized attributes*. Attributes whose values are apssed down the tree are called *inherited attributes.*

Terminals may have only synthesized attributes, and their values are assigned by the lexical analyzer. Inherited values of the Start symbol. If any, are given values by way of parameters when attribute evaluation begins.

Ref 1: <http://homepage.cs.uiowa.edu/~hzhang/c123/Lecture3.pdf>

Ref 2 : <http://cecs.wright.edu/~tkprasad/papers/Attribute-Grammars.pdf>

Ref 3: <https://en.wikipedia.org/wiki/Attribute_grammar>