Compiler Design Predictive Translation

- The following algorithm generalizes the construction of predictive parsers to implement a translation scheme based on a grammar suitable for top-down parsing.

Algorithm:

- Construction of a predictive syntax-directed translator.

Input:

- A syntax-directed translation scheme with an underlying grammar suitable for predictive parsing.

Output:

- Code for a syntax-directed translator.

Method:

The technique is a modification of the **predictive-parser** construction.

- For each nonterminal A, construct a function that has a formal parameter for each inherited attribute of A and that returns the values of the synthesized attributes of A
- The code for nonterminal A decides what production to use based on the current input symbol.

- The code associated with each production does the following. We consider the tokens, nonterminals, and actions on the right side of the production from left to right.
    - For token **X** with synthesized attribute x, save the value of x in the variable declared for X.x. Then generate a call to match token X and advance the input.
    - For nonterminal B, generate an assignment **c := B (b1, b2, ..., bk)** with a function call on the right side, where b1, b2, ..., bk are the variables for the inherited attributes of B and c is the variable for the synthesized attribute of B.
    - For an action, copy the code into the parser, replacing each reference to an attribute by the variable for that attribute.

For More Details Click Here:

https://www.wikitechy.com/tutorials/compiler-design/predictive-parsing-algorithm

