LAYMAN INTERPRETED PROGRAMMING LANGUAGE LIPROL

HUSSAIN AHMAD AND MUHAMMAD ZUBAIR ASGHAR
Institute of Computing and Information Technology, Gomal University Dera Ismail Khan (NWFP) Pakistan

ABSTRACT

This work deals with the development of language interpreter. Developing a complete language compiler is somewhat tedious, but creating a language Interpreter is quite simple and easy. The language interpreter that has been developed here can be used very easily even by those, who have no previous programming background. The major phases of LIPROL (layman interpreted programming language) are: Lexical analysis, Syntax analysis, Symbol table management, and Semantic analysis. Methods of creating language interpreters are rarely taught in computer science courses. This paper unleashes the mechanics of token extraction, expression composition and expression parsing. LIPROL provides the students/users an interactive environment, easier to use than Basic language interpreter. It also provides excellent interactive debugging facilities.

INTRODUCTION:

In case of interpretive process, no object program is generated. In microcomputer environment, interpreters are becoming more popular, because overhead of interpretation is less as compared to compilation process (Trembly, 1987). Basic, Java, Allis, and small talk-80 are interpreted Languages (Herbert,1987;Trembly,1985). LIPROL stands for layman interpreted programming language. It is BASIC like interpreter with small instruction set. Commands used in LIPROL are very easy to understand. Even a layman having no previous programming experience can use this interpreter very easily. LIPROL recognizes the following constructs;
Show, Get, If, Then, Next, To, Go to, Call sub, Return, End.

METHODOLOGY

Internal working of LIPROL interpreter is given as under.

Token Handling : First phase in translation process, whether it is compilation or interpretation is token generation. It is also called lexical analysis or scanning process. In this phase input string is decomposed into small pieces. Each piece is called lexeme or token. For example, the expression X + Y - (Z +3) has following tokens X, +, Y, -, Z, +, 3. The function that decomposes input expression into tokens has following tasks.

a. Omit tabs and spaces
b. Extraction of each token
c. Each class of token is given a unique internal representation no.
d. Determine the type of token.

There are two formats of token representation: External format, internal format
External Format, also called string representation of token. e.g. INPUT is the external format of INPUT command in BASIC language. Rules of Interpreter design say that external format for token representation should be avoided, because it is inefficient approach. In Internal Format, unique integer is assigned to each command e.g. INPUT command is given 1, FOR is represented by 2 etc. In internal representation, integers are used instead of strings. Using .this approach, quite faster and efficient routines can be written.

Analysis:
In LIPROL, each instruction of program is represented and stored as a string. A function named next_ token ( ) is used that