Abstract

A compiler translates and/or compiles a program written in a suitable source language into an equivalent target language through a number of stages. Starting with recognition of token through target code generation provide a basis for communication interface between a user and a processor in significant amount of time. A new approach GLAP model for design and time complexity analysis of lexical analyzer is proposed in this paper. In the model different steps of tokenizer (generation of tokens) through lexemes, and better input system implementation have been introduced. Disk access and state machine driven Lex are also reflected in the model towards its complete utility. The model also introduces generation of parser. Implementation of symbol table and its interface using stack is another innovation of the model in acceptance with both theoretically and in implementation widely.
A New Approach of Compiler Design in Context of Lexical Analyzer and Parser Generation for NextGen Languages

Reference

- Lex table look up: www.gradsoft.kiev.ua
- Symbol table management: www.faculty.washington.edu

Index Terms

Computer Science Compilers

Key words

Tokens Lexeme

Lex

Lexeme

Tokenizer

PDA
Lookahead

Pushback