International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

Translation of English Algorithm in C Program using Syntax Directed Translation Schema

Hareesha Ganapati Gummani
Student, Dept. of Computer Science and Engineering, M. S. Ramaiah Institute of Technology, Bengaluru, India

***Abstract*: Natural language Processing (NLP) and machine learning is most trending area of research now days. Automatic Translation of any Application like that is written in English can be converted in to corresponding C programming language by applying some rule based approach. The technique is by using syntax directed translation schema. This technique is not using any intermediate representation. The input to the system is Algorithms that is written in natural languages like C and output is C program. The tools using in this paper is Flex or Scanner for token generation and rules defined for string. The other tool is Bison or parser is for Phrase structural grammar of NLP and related semantic action**

***Keywords*: Compiler optimization, machine learning, NLP, C programming.**

1. Introduction

Parser is basically used as a CFG(Context-Free-Grammar) to check and validate the input string and generate output for next steps of the compiler. Output could be either a parse tree or abstract syntax tree. Now to interleave semantic analysis with syntax analysis phase of the compiler, we use Syntax Directed Translation.

Definition: a CFG where each grammar production A → α is associated with a set of semantic rules of the form b = f (c1, c2, ..., ck);
where: b is a synthesized attribute of A or an inherited attribute of one of the grammar symbols in α c1, c2, ... are attributes of the symbols used in the production.

So in our paper we have propose the model to convert the English like language to the corresponding c programming language by sequence of steps by using the Syntax Directed Translation Shema (SDD).

2. Literature survey

This segment presents an idea of common language enhanced programming dialects. What's more, in this paper depicted framework related, little history of unadulterated naturalistic programming framework. A. Regular Language Supplemented Programming Languages what compiler heuristic or streamlining to apply so as to make the code better. Better regularly implies execute quicker, however can likewise mean littler code impression or decreased power. Machine learning can be utilized to fabricate a model utilized inside the compiler that settles on such choices for some random program.

A. KlarDeutsch

KlarDeutsch, signifying "clear German" is a more up to date, yet obscure and rather basic methodology of regular language enhanced programming in the specialized space. KlarDeutsch has been created by Andover Corporation in 1995.Its reason for existing is to control machines and hardware with essential characteristic language sentences, written in German. A KlarDeutsch program meant English could look as pursues

B. Pure naturalistic programming

Since the sixties there have been endeavours to create programming dialects which would permit composing programs in unadulterated normal language.

C. Apple script

Despite the fact that AppleScript utilizes regular language, as with any characteristic language enhanced programming language, AppleScript programs are somewhat a sort of etymological veil mirroring the structures of a customary programming language. For example, on the off chance that else articulations are not communicated by means of sentences but rather in the normal style of programming dialects. Notwithstanding that, there is no chance to get of including client characterized elements, all activities must be performed on existing articles like sound or application.

Then again, AppleScript is one of only a handful couple of programming dialects, which were multilingual, in any event for quite a while. Until form 8.5 of Mac OS, AppleScript projects could be written in a few common dialects, among them English, German, Spanish, French, Italian, just as Japanese and Chinese. Sadly, Apple surrendered the multilingual methodology, as a result of the convoluted costumer support.

3. Proposed methodology

A. Flex

FLEX is quick lexical analyzer generator device or PC that creates lexical analyzer. It is composed by Vern Paxson in C in 1987.It is utilized with Yacc parser or GNU Bison parser generator. Flex and Bison are produces quicker code and furthermore more adaptable than Lex and Yacc. When client input record Bison delivers the parser. The yylex() work is consequently produced naturally by the flex when it is given

with a .l record and this yylex() work is relied upon by to recover tokens created from current/this token stream.

Bison is a universally useful parser generator that changes over a punctuation depiction (Bison Grammar Files) for a LALR (1) setting free language into a C program to parse that sentence structure. The Bison parser is a base up parser. It attempts, by movements and decreases, to diminish the whole contribution down to a solitary gathering whose image is the language's begin image.

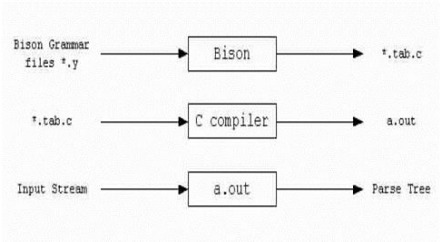

Fig. 1. Process

Write a lexical analyzer to process input and pass tokens to the parser (calc.lex).
Write the grammar specification for bison (calc.y), including grammar rules, yyparse() and yyerror().
Run Bison on the grammar to produce the parser. (Makefile)
Compile the code output by Bison, as well as any other source files.
Link the object files to produce the finished product.

Programmed Translation of any Application like that is written in English can be changed over in to relating C programming language by applying some standard based methodology. The strategy is by utilizing punctuation coordinated interpretation diagram. This procedure isn't utilizing any middle of the road portrayal. Translator of English calculation to C program has been created by following principles following framework modules

1. A scanner or Flex is do the coordinating information string and produce tokens for related string.
2. A parser generator or Bison for info string linguistic structure checking and producing related semantic activity.
3. At that point next module/stage a Gcc Compiler interface Flex and Bison created lex.yy.c and Y.tab.c to produce last framework executable document, by composing order on CMD like Gcc lex.yy.c Y.tab.c –o TEC.exe.
4. After that as forward module/last advance executable record create yield .c document with given information content document, by composing direction on CMD like TEC.exe Algo.txt Output.c

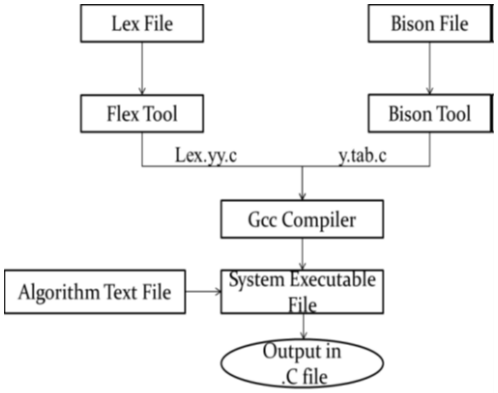

Fig. 2. Flow chart

Tatiana Vert, Tatiana Krikun and Mikhail Glukhikh in their paper "Detection of Incorrect Pointer Dereferences for C/C++ Programs utilizing Static Code Analysis and Logical Inference" have done static code analysis accuracy utilizing classic code algorithm with conditions. The key characteristics of mistake discovery strategies are based on soundness, exactness, and performance. To achieve all the characteristics is contradictory as one of them is to be undermined to increase the proficiency of the other two. This is understood by a logical interface apparatus by building a source code model for the program. The most helpful model which can be utilized for code analysis is a control Flow graph (CFG).

4. Conclusion

The framework was tried with fundamental C idea like scanf(), printf(), if(), if() else(), for(), automatic variable, cluster and pointer statement. The framework returned right important interpretations in the majority of the cases of code.

References

[1] Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, “Compilers Principles, Techniques & Tools,” Pearson Publication.
[2] Nisha N. Shirvi, Mahesh H. Panchal, “Translation of English Algorithm in C Program using Syntax Directed Translation Schema.”
[3] https://www.geeksforgeeks.org/flex-lexical-analyzer-generator/
[4] http://alumni.cs..edu/teaching/bison.html
[5] https://www.geeksforgeeks.org/parsing-set-3-slr-and-lalr-parsers/
[6] Huang, Liang, Kevin Knight, and Aravind Joshi. "A syntax-directed translator with extended domain of locality." in Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing, pp. 1-8. Association for Computational Linguistics, 2006.
[7] Amengual, Juan Carlos, Asunción Castaño, Antonio Castellanos, Victor M. Jiménez, David Llorens, Andrés Marzal, Federico Prat et al., "The eutrans spoken language translation system." Machine Translation 15, no. 1-2 (2000): 75-103.