Indo-Iranian Journal of Scientific Research (IIJSR)
Peer-Reviewed Quarterly International Journal, Volume 2, Issue 2, Pages 100-104, April-June 2018

Simulation of Compiler Phases

P.Abinaya[1], G.Deepica[2] and Ms V.Jyothir Bindhu[3]

[1,2]UG Students, Department of Computer Science and Engineering, National Engineering College, Kovilpatti, India.
[3]Assistant Professor, Department of Computer Science and Engineering, National Engineering College, Kovilpatti, India.
Email: deepica2311@gmail.com[1], abiperumalskani@gmail.com[2]

Article Received: 30 December 2017 *Article Accepted: 29 March 2018* *Article Published: 31 May 2018*

ABSTRACT

This project describes simulators, which are programming tools that make available for constructing complier. This project consists of a set of educational software simulators built to improve teaching with quality and provide tools for the remote teaching project to assess the knowledge of the students through test and assignments, to develop a laboratory environment for the students. We have being introduced a simulator especially designed for compiler construction. Starting from token generation to intermediate code generation provides a user interface with the simulator. The objective of this research is to develop simulator that gives more flexibility for users by providing a friendly user interface, large set of operations, and knowledge base of these machines. This is the foundation for an integrated teaching environment on the Web. The motivation for this work was the lack of educational software for teaching theoretical computations, and also the importance of generating qualified human resources to work. This work is meant to help students, through simulated programs, to understand the computational formality studies in advanced simulators, which makes available formalisms such as token generation, syntax tree, code optimization, intermediate code generation. The objectives of these simulators are the development of a laboratory Environment for the students. Here students can develop programs in different machines, run programs step by step for learning and correction, solve exercises, and provide assistance for teachers in the working out and correction of exams. Due to the good quality of the works presented, it was decided to develop a project to make instructional packages available in a local environment. The final result of this project is to provide general knowledge about compiler design.

Keyword: Token generation, code optimization, intermediate code generation.

1. INTRODUCTION

The purpose of this document is to present a detailed description of the Simulation of Compiler Phases. It will explain the purpose and features of the tool, the interfaces of the software, what the simulator will do and the constraints under which it must operate. Typical Users, such as students, who want to use simulators for studying, learning and visualizing suppose. Trainers who ensure the exercise and assessment of the people they train. Programmers are interested in formulating machines such as compilers. The main scope of this stimulator is to create learning and practicing tool for computation, and hence design creative complier machines, to improve teaching with quality and provide tools for the remote teaching project, to assess the knowledge of the students through test and assignments, to develop a laboratory environment for the students that are Bachelors in Computer Science. Compilers and Operating System provides an interface between programmers and machine. Compiler is a program that converts high level to low level programming languages, but we are intended to do up to generation of intermediate code generation phase in the compiler.

2. RELATED WORKS

The early emphasis on correctness has consequences for your approach to the design of the implementation. Modularity and simplicity of the code are important for two reasons: first, your code is much more likely to be correct, and, second, you will be able to respond to changes in the source language specification. In a production compiler, efficiency of the generated code and also efficiency of the compiler itself are important considerations. In this course, we set very lax targets for both, emphasizing correctness instead. In one of the later labs in the course, you will have the opportunity to optimize the generated code.

Laboratory work may be organized either around an open or closed structure, but usually consists either in building one or more programs that implement the algorithms and techniques taught in the classroom to solve a set of selected problems or in writing applications according to some specified requirements. In addition, the teacher is available during some scheduled office hours to assist students with their personal work. Our early teaching practice was not very different from this cartoon and its results were quite similar: we decided to give pass degrees to students who had clearly failed in spite of having worked really hard. On the other side, we have really enjoyed teaching compiler design this way, and we find ourselves actively doing research in an area which is of high interest to us.

3. SYSTEM MODEL

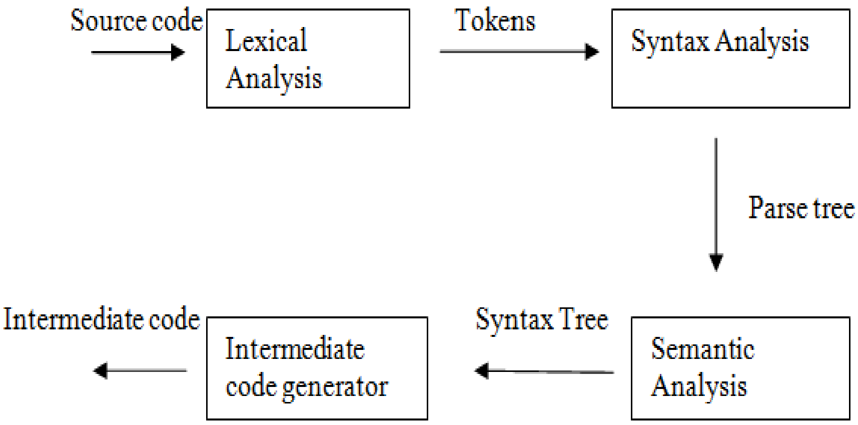

Fig: System flow

4. PROPOSED SYSTEM

4.1. Study of the Theory of computation

Simulation of compiler phases is developed in C#, Microsoft Visual Studio. There are several APIs available to make plug-in development easy. This screen will be the opening page of the tool that allows studying the introduction, basis, terminologies, definitions of every phases of a compiler and examples of every transactions. This page is very mandatory since many tools lag in the basics teaching specially with the non-user. This also includes the procedure for working in the tool. Since it is a simulator user needs to refer tutorials from other websites and manual. But with this feature added user can refer the tool itself for working in it. Most of the tools available in the market do not provide manual or any instructions for the tool. In this modern world people like to work with less effort rather than tedious browsing and surfing the internet. This tool will break that scenario and be a complete tool for helping students in working with the simulator tool.

4.2. Lexical Analysis

In programming language words are objects like variable names, numbers and keywords. A lexical analysis takes input from the source language consider string as individual letters and divides strings into tokens. Additionally, it will filter out whatever comments separates the tokens (the so-called white-space), i.e., lay-out characters (spaces, newlines etc.). The main purpose of lexical analysis is to make life easier for the subsequent syntax analysis phase. The set of all integer constants or the set of all variable names are sets of strings, where the individual letters are taken from a particular alphabet. Such a set of strings is called a language.

4.3. Syntax Analysis

Lexical analysis splits the input into tokens, the purpose of syntax analysis is to recombine these tokens not back into a list of characters but into something that reflects structure that is., in the form of tree. Hence, what is important in the syntax tree is how these leaves are combined to form the structure of the tree and how the interior nodes of the tree are labeled. In addition to finding the structure of the input text, the syntax analysis must also reject invalid texts by reporting syntax errors. The root of the tree is the start symbol of the grammar, and whenever we rewrite a no terminal we add as its children the symbols on the right-hand side of the production that was used.

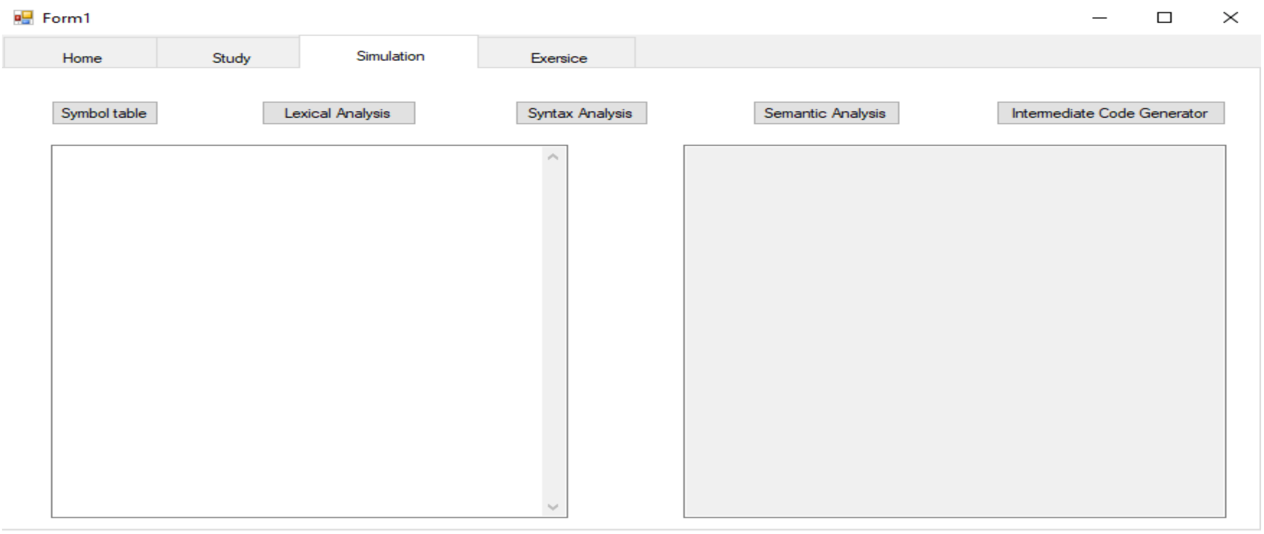

Figure 2: GUI

5. CONCLUSION

Simulation of compiler phases provides the users with both simple and advanced features. Due to its well designed and easy to use interface it can be used by both experts and typical users. However, users must already have a basic knowledge of computation before using it. The objective of this project is to develop software that gives more flexibility for its users in editing and creating abstract machines, by providing a friendly user interface, large set of operations, and knowledge base of these machines. On more aspects of compilation than a student is likely to see in a compiler course for more tradition all Languages. With these simulators it is possible to teach Compiler Design in a new way, where students may visualize numerical computed functions and check phases transactions including, the chapters of symbol table. This will eventually lead to better understanding of compiler and machines leading to creation of better compliers and algorithm for enhancing the technology for the evolution of human race.

REFERENCES

[1] Niklaus WirthThis is a slightly revised version of the book published by Addison-Wesley in 1996ISBN 0-201-40353-6Zürich, November 2005.

[2] Muthukumaran. N and Ravi. R, 'Hardware Implementation of Architecture Techniques for Fast Efficient loss less Image Compression System', Wireless Personal Communications, Volume. 90, No. 3, pp. 1291-1315, October 2016, SPRINGER.

[3] Aho, Alfred V., Hop croft, J. E., and Ullman, Jeffrey D. [1974]. The Design and Analysis of Computer Algorithms. Addision Wesley, Reading, MA.

[4] Muthukumaran. N and Ravi. R, 'The Performance Analysis of Fast Efficient Lossless Satellite Image Compression and Decompression for Wavelet Based Algorithm', Wireless Personal Communications, Volume. 81, No. 2, pp. 839-859, March 2015, SPRINGER.

[5] Sale, Arthur H. J. [1977]. Comments on 'report on the programming language Euclid'.ACM SIGPLAN Notices, 12(4):10.

[6] Muthukumaran. N and Ravi. R, 'VLSI Implementations of Compressive Image Acquisition using Block Based Compression Algorithm', The International Arab Journal of Information Technology, vol. 12, no. 4, pp. 333-339, July 2015.

[7] Schulz, Waldean A. [1976]. Semantic Analysis and Target Language Synthesis in a Translator.Ph.D.thesis, University of Colorado, Boulder, CO.

[8] Muthukumaran. N and Ravi. R, 'Simulation Based VLSI Implementation of Fast Efficient Lossless Image Compression System using Simplified Adjusted Binary Code & Golumb Rice Code', World Academy of Science, Engineering and Technology, Volume. 8, No. 9, pp.1603-1606, 2014.

[9] Ruban Kingston. M,Muthukumaran. and N, Ravi. R, 'A Novel Scheme of CMOS VCO Design with reduce number of Transistors using 180nm CAD Tool', International Journal of Applied Engineering Research, Volume. 10, No. 14, pp. 11934-11938, 2015.

[10] Research paper on Compiler Design, Jatin Chhabra, Hiteshi Chopra, Abhimanyu Vats, Gurgaon, India.

[11] Muthukumaran. N and Ravi. R, 'Design and analysis of VLSI based FELICS Algorithm for lossless Image Compression', International Journal of Advanced Research in Technology, Vol. 2, No. 3, pp. 115-119, March 2012.

[12] Manoj Kumar. B and Muthukumaran. N, 'Design of Low power high Speed CASCADED Double Tail Comparator', International Journal of Advanced Research in Biology Engineering Science and Technology, Vol. 2, No. 4, pp.18-22, June 2016.

[13] N. Muthukumaran, 'Analyzing Throughput of MANET with Reduced Packet Loss', Wireless Personal Communications, Vol. 97, No. 1, pp. 565-578, November 2017, SPRINGER.

[14] P.Venkateswari, E.Jebitha Steffy, Dr. N. Muthukumaran, 'License Plate cognizance by Ocular Character Perception', International Research Journal of Engineering and Technology, Vol. 5, No. 2, pp. 536-542, February 2018.

[15] N. Muthukumaran, Mrs R.Sonya, Dr.Rajashekhara and Chitra V, 'Computation of Optimum ATC Using Generator Participation Factor in Deregulated System', International Journal of Advanced Research Trends in Engineering and Technology, Vol. 4, No. 1, pp. 8-11, January 2017.

[16] Ms. A. Aruna, Ms.Y.Bibisha Mol, Ms.G.Delcy, Dr. N. Muthukumaran, 'Arduino Powered Obstacles Avoidance for Visually Impaired Person', Asian Journal of Applied Science and Technology, Vol. 2, No. 2, pp. 101-106, April 2018.

[17] Mrs. S. Murine Sharmili, Dr. N. Muthukumaran, 'Performance Analysis of Elevation & Building Contours Image using K-Mean Clustering with Mathematical Morphology and SVM', Asian Journal of Applied Science and Technology, Vol. 2, No. 2, pp. 80-85, April 2018.

[18] Keziah. J, Muthukumaran. N, 'Design of K Band Transmitting Antenna for Harbor Surveillance Radar Application', International Journal on Applications in Electrical and Electronics Engineering, Vol. 2, No. 5, pp. 16-20, May 2016.

[19] B.Renuka, B.Sivaranjani, A.Maha Lakshmi, Dr. N. Muthukumaran, 'Automatic Enemy Detecting Defense Robot by using Face Detection Technique', Asian Journal of Applied Science and Technology, Vol. 2, No. 2, pp. 495-501, April 2018.

[20] Ms.Mary Varsha Peter, Ms.V.Priya, Ms.H.Petchammal, Dr. N. Muthukumaran, 'Finger Print Based Smart Voting System', Asian Journal of Applied Science and Technology, Vol. 2, No. 2, pp. 357-361, April 2018.

[21] Muthukumaran. N and Ravi. R, 'Quad Tree Decomposition based Analysis of Compressed Image Data Communication for Lossy and Lossless using WSN', World Academy of Science, Engineering and Technology, Volume. 8, No. 9, pp. 1543-1549, 2014.