

Demonstration of Software for Optimizing Machine Critical Programs by Call Graph Generator

Pravin Kandala^a, Jasmeen Kaur^b, Prof. T. Venkat Narayana Rao^{c*}

^a Student, Department of Computer Science, *Guru Nanak Institutions Technical Campus, Hyderabad, India*

^b Student, Department of Computer Science, *Guru Nanak Institutions Technical Campus, Hyderabad, India*

^c Professor, Department of Computer Science, *Guru Nanak Institutions Technical Campus, Hyderabad, India*

^aEmail: Pravinkandala@hotmail.com

^bEmail: Jasmeenkaur807@gmail.com

^cEmail: tvnrobby@yahoo.com

Abstract

While working with software that is complex, representation in visual forms improves the understanding and also enhances the programmers' ability to analyze the relationships between the components of a code. Placing all tools together which performs cyclomatic complexity on mission critical codes to optimize the solution is the real motto of this work. These include generation of call graphs, which are visually represented with different metrics and to assist in software coding to the programmers. The different metrics include total number of lines in the function, total number of executable lines, and number of unreachable lines. This tool accepts only C program and generates a function call graph along with function metrics providing both static and dynamic view. This software helps the developer to take decision on optimality of the program and to know the program flow, thus optimizing the program. This paper depicts the working of call graph Generator to assess the reachability and exactness of the programs.

Keywords: Call graph; cyclomatic complexity; C program; optimality; call graph Generator

* Corresponding author.
E-mail address: tvnrobby@yahoo.com.

1. Introduction

Intention behind the proposal of this system is to generate a call graph to a 'c' program. The 'C' source code is taken as an input to the system and a call graph is drawn in a hierarchical way starting from the main function of the programs and moving down until all the functions used in the program have been exhausted. The nodes in the graph at each level represent the function called by the functions in the previous level. The relationship is represented by the directed edges from one node to other. This facilitates the user to understand the program clearly. Call Graphs provides the systematic information about the programs by indicating the calling relationships between all the subroutines in the 'C' program. The Call graph generation for mission critical software is to enable the people to understand the program so that they can make an easy review, modify the code or use the defined functions in the program in other programs. The company administrative personnel would maintain a database of all the 'c' programs which are to be analyzed and store the information in another data base that is useful for them in above mentioned ways. The proposed system is a software tool that facilitates the user needs. So, it does not need any administrator to maintain. When the user clicks the icon, it prompts for the input file, a call graph is shown with all the subroutines in the program and also can provide the basic metrics like cyclomatic complexity, number of lines in the program. It also makes the user to know about the total number of executable lines in the given program and list of unused functions in a given 'C' program [1].

The system consists of following tasks:

1. The system will accept c files from the user.
2. Functions in the program are identified, and the calling functions are listed out.
3. Function metrics is calculated and call graphs are generated.

The scope of the Call Graph generator is as follows:

1. It is applicable to all types of 'C' programs.
2. It can work in Windows NT and Linux operating systems.
3. It can locate the relationship between the subroutines in the source code but doesn't define the relationship with the libraries.
4. It can distinguish between a library function and a user defined function in the source code.

Drawbacks of existing scenario in programming domain:

1. Previously normal codes are just compiled and get the result with time and space complexity directly.
2. They are parsed and complexity is indicated.
3. Dead code is not executed or shown error.
4. Displays no. of lines, but not executed lines.
5. User himself should write the efficient code carefully and execute it.

2. Proposed system

The system evaluates each line and function appropriately both dead and normal code. It displays a tree like graph, which is easier for user to identify the complexity and relation between the functions starting from the main() function. As the complexity is identified by cyclomatic complexity method, the complex function is manually deleted [4].

3. Architecture

The Figure 1 illustrates input of source code files and how they are processed into three phases to get cyclomatic complexity. Using these functions calculation of metrics, and other required measures are generated. Finally, call graphs are generated using jGraphT API.

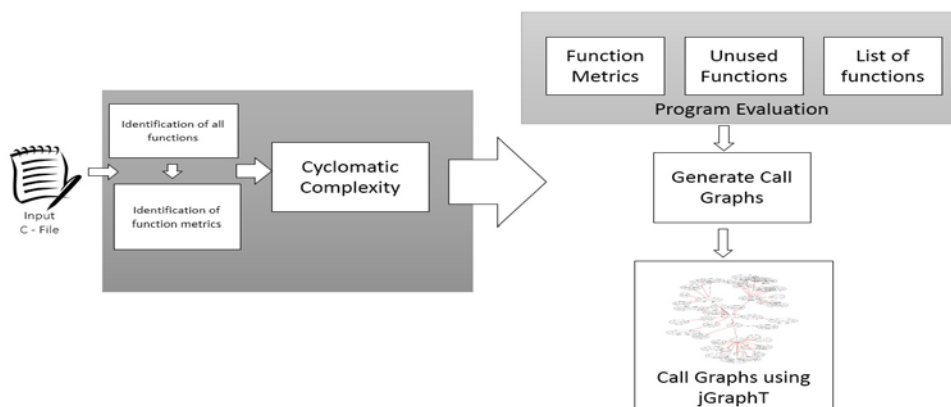


Figure 1 : Architecture diagram of optimizing programs using call graphs.

4. Implementation

The user will be prompted to select a 'C' file. Then the file is split into stream of tokens and the tokens are passed into another file. Then these tokens are checked for function prototype. The functions obtained are shown to the user for his confirmation. Then the function calls for each function is obtained and stored as function objects [2]. The function objects have different set of parameters that are used to find the metrics for each function. The metrics of every function is presented to the user in the graphical manner using tables. The user has the option to print the data presented in the table for future verification. The call graph is generated using jgrapht library package, where in the nodes of the graph indicates the functions and the links indicate the function calls [3].” Custom Look And Feel” package is used for better presentation of the user interface. In the system, data stream is first searched for function prototype, if found the function counter is incremented and then function name is saved. This is done until end of the stream. If no function prototype is found then, it is sent to further process or revert as per instructions.

5. Results

The system prompt the home page to initiate the task with option to read and select input file as shown in figure 2a, 2b 2c.



Figure 2 a, b, c: Basic Screen Snap shots for system interface.

The following is the graph form representation of the input program shown in figure 3 with links and sub links.

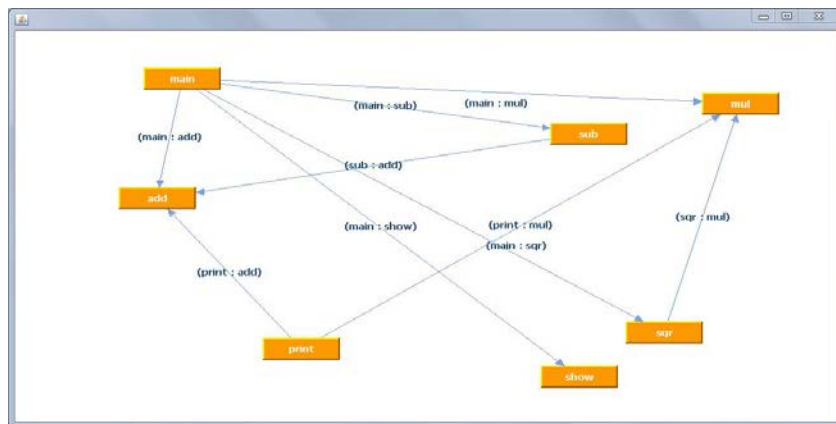


Figure 3: Screen shot for the output for the generation of call graphs after computing all the values.

The functions metrics is calculated based on the graph form and metrics are generated in the list with details of Function name, called function names and number of lines of code in each function as shown in figure 4. Table I and II has showcased the analysis and table II states the procedure status, risk of function usage and bad fix probability of the function.

Table 1. The results of the Cyclomatic Complexity for sample code

FUNCTION NAME	CALLS FUNCTION	NUMBER OF LINES
main	add() sqr() sub() mul() ...	9
add	no function	5
sub	add()	3
mul	no function	5
sqr	mul()	3
print	add() mul()	4
show	no function	3

Table 2: Analysis of cyclomatic complexity values, its analysis, Risk and probability of bad fix

FUNCTION NAME	CYCLOMATIC CO...	ANALYSIS	RISK	BAD FIX PROBABILI...
main	0	simple procedure	low	5%
add	0	simple procedure	low	5%
sub	0	simple procedure	low	5%
mul	0	simple procedure	low	5%
sqr	0	simple procedure	low	5%
print	0	simple procedure	low	5%
show	0	simple procedure	low	5%

6. Conclusion

This paper summarizes the overall working of system focusing on assistance extend to the programmers during large project coding. The problems encountered during the project development can be understood and shall be modified through this method. The work done in this paper can be a vital software analytic tool, which helps in understanding and analyzing the C program. This is a simple tool has no complexities which would assist the programmer easily understand the functionalities of the program through call graphs. These graphs are helpful while modifying, testing, documenting, explaining and maintaining the code with a facility of automatically generated call graphs.

7. Limitations

- The call graph is static; any dynamic linking of function will not be depicted in graph.
- This will provide only the basic metrics of the program.
- If there is no main function it will show just an error message and terminates.
- The links will be increased when function calls are high. So there is a possibility for overlap. ^[9]

8. Future enhancements

The enhancements that can be made to the existing work are:

- Provision for identifying the unused variables in the program.
- A scrollable presentation of the Call Graph

- Self-organized graph can be developed.
- Dynamic call graph is more useful than static one.
- More metrics can be shown to the user.

References

- [1]. Optimizing Programs Using Call Graphs by Pravin Kandala, Jasmeen Kaur and Prof. T. Venkant Narayana Rao ,IJARES, April, 2014.
- [2] A J Sojev. "Basis Path Testing"
- [3] John Sichi, JGraphT - Project Administrator.
- [4] McCabe (December 1976). "A Complexity Measure". IEEE Transactions on Software Engineering: 315.
- [5] McCabe, Watson (1996). "Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric".
- [6] Diestel, Reinhard (2000). Graph theory. Graduate texts in mathematics 173 (2 ed.) New York: Springer. ISBN 0-387-98976-5.
- [7] Harrison (October 1984). "Applying Mccabe's complexity measure to multiple-exit programs". Software: Practice and Experience (J Wiley & Sons).
- [8] McCabe (December 1976). "A Complexity Measure". IEEE Transactions on Software Engineering: 308–320. Template: Working link
- [9] C grammar defintion for use with javaCC Contributed by Doug South.

AUTHORS BIOGRAPHIES



#1 Pravin Kandala is currently in his 4th year at the Guru Nanak Institutions Technical Campus pursuing Bachelors in Computer Science and Engineering. And he is a Microsoft Student Partner. Member at National App Review Board Microsoft. Holds Microsoft Associate, Microsoft Certified Professional and Microsoft Certified Solution Developer on Web platform and Windows store. Has developed apps on Windows 8, windows phone, blackberry platforms. Over the past four years he has been actively participating in projects, technical seminars and workshops. He avers Computer research oriented study and has been working on Cloud computing, Mobile applications, algorithms, Augmented Reality, Artificial Intelligence, Image processing. He can be reached at pravinkandala07@gmail.com.



#2 Jasmeen Kaur is currently in her 4th year at the Guru Nanak Institutions Technical campus pursuing Bachelors in Computer Science and Engineering. Over the past four years she has been actively participating in several technical presentations, seminars & workshops. Member and organizer at Hyderabad Youth Assembly. She holds

certifications for Microsoft technology associate and Microsoft certified solution developer. Has developed Windows 8 apps. Her major interest of research lies in the areas of Computer and Developing, Network Security, Computer vision, Theoretical Computer Science, Data Structures, Machine learning. She can be reached at jasmeenkaur807@gmail.com.



Professor T.Venkat Narayana Rao, received B.E in Computer Technology and Engineering from Nagpur University, Nagpur, India, M.B.A (Systems), holds a M.Tech in Computer Science from Jawaharlal Nehru Technological University, Hyderabad, A.P., India and a Research Scholar in JNTU. He has 21 years of vast experience in Computer Science and Engineering areas pertaining to academics and industry related I.T issues. He is presently working as Professor, Department of Computer Science and Engineering, Guru Nanak Institutions Technical Campus, Ibrahimpatnam, R.R.Dist., A.P, INDIA. He is nominated as an Editor and Reviewer to 38 International journals relating to Computer Science and Information Technology and has published 61 papers in international journals. He is nominated as member on international Advisory Committees of International Conferences and number of reputed professional bodies. He is currently working on research areas, which include Digital Image Processing, Digital Watermarking, Data Mining, Network Security and other emerging areas of Information Technology.