CS840 Project 4:

Cyclomatic Complexity

In C++ Programs

Samuel Gluss

Jozo Dujmovic

4-1-2016

Abstract – Cyclomatic complexity for a program is defined as the number of logic conditions in a program plus one. Cyclomatic complexity is directly correlated to cognitive complexity, which means it is an effective barometer for how challenging a piece of code is to understand. Cognitive complexity is not always obvious to determine, as it depends on the experience and ability of the developer, as well as how abstruse the code is. However, because cyclomatic complexity is relatively easy to determine objectively, it is possible to use it as a tool to compare programs which are similar in terms of their conceptual difficulty level. The resulting value returned can give some insight into how difficult the code may be to follow through its various control statements. A program with no loops or controls is a simple vertical path from top to bottom, whereas a program with many loops and controls may take a convoluted path which is confusing to the observer, having as they do only five or so registers in their brains.

# I. Introduction:

This experiment will be conducted by examining seven C++ programs, five of which are from a programming languages class assignment, and two of which are benchmarks from a previous project. These programs should give a good representation of the average C++ module, although none of them implement object oriented principles beyond utilizing a random generator.

The parser is a Python script which loads a programs source code into memory, then strips out all non-normal code such as comments, preprocessor directives, strings and characters. It does this with a state machine that only adds code to the output if it is deemed to be in a ‘normal’ code segment. This ensures that none of these are accidentally matched by the regexes which are then used to match any for and while loops, if statements, and case switches. Inside each match for an if, for, or loop condition, the match.count() function is used to find the number of logical and (&&) and logical or (||) present in each of these bodies. Once the entire source file has been parsed, these values are added up to obtain the cyclomatic complexity, defined as the number of condition bodies, plus the number of Boolean operators within these bodies.

Contents

[I. Introduction: 1](#_Toc447282527)

# Citations

1. McCabe, Thomas J. – A Complexity Measure – IEEE Transactions 1976