1 Introduction

1.1 Purpose

Testing has two approaches. One is structural testing and the other is functional testing. In functional testing, experts conduct testing against project specifications. And in structural testing, experts conduct testing against structure of project. Structure of project is defined some models. In this project we use CFG (Control Flow Graph) which is one of model used in structure testing. When our program receives a c source code, the program converts the source code to CFG.

1.2 Scope

1.2.1   Product Name

The product name is 'CFG generator'.

1.2.2   Product Main Functions

Convert a c source code to CFG

1.3 Terms of Use

…

1.4 References

IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications

Software Testing , Claire Lohr

Software Testing and Analysis" by Mauro Pezzè and Michal Young, WILEY

1.5 Overview

Section 2 shows basic functions, assumptions about users, and constraints. Section 3 describe function specification and non-functional specification of the project.

2 Overall description

2.1 Product Perspective

2.1.1   User interface

UI is CUI (Character User Interface). A user uses command line in order to command our program. Report of our program shows list of 'states' and 'edges' of CFG.

e.g.) ./CG Inputcode.c result.txt

2.2 Product functions

Convert inputted c code to CFG.

2.3 User Characteristics

A user can compile with gcc on Cygwin.

The user also knows branch coverage, statement coverage and CFG.

2.4 Constraints

- Development language : C

- Execution environment : Cygwin

3 Specific Requirements

3.1 External interface requirement

3.1.1 User Interfaces

3.1.1.1 Report with a text (*.txt) file.

3.1.1.2 The report show all states and edges of CFG.

3.2 Functional Requirement

3.2.1 Execution

3.2.1.1 The program uses CUI. Refer 2.1.1.

3.2.1.2 When a user inputted unpermitted command, the program shows 'help'(that includes command syntax).

3.2.2 Report generating process

3.2.2.1 Report shows execution order of c source code.

3.2.2.2 When c source code inputted successfully, program shows "success" message. Or in error case, the program shows "error" and terminates the program.

3.2.2.3 Before the program converting CFG, shows "converting" message.

3.2.2.4 After report generating process, the program shows the name of report file.

3.3 Performance Requirement

3.3.1 C source code has 100~200 lines. It includes main function.

3.3.2 The C source code doesn't have user defined header files.

3.3.3 The C source code doesn't include pointers.

3.4 Design Constraints

3.4.1   We use SASD for developing our program.