

# 1 Introduction

## 1.1 Purpose

테스팅의 방법으로는 Functional Testing과 Structural Testing이 있다. 프로젝트 specification 따라 테스트 케이스를 작성하는 Functional Testing 과는 달리 Structural testing은 프로젝트의 구조에 따라 테스트 케이스를 작성한다. 이러한 프로젝트의 구조는 여러 가지 모델로 정의 할 수 있는데, 본 프로젝트에서는 모델 중 하나인 Control Flow Graph를 다루고자 한다. C언어로 작성된 소스코드를 받아 해당 소스코드의 CFG를 그리는 도구를 개발한다.

## 1.2 Scope

### 1.2.1 Product Name

개발하는 소프트웨어의 이름은 CFG Generator로 정한다.

### 1.2.2 Product Main Functions

본 프로젝트는 주어진 C 코드를 CFG로 변환 시켜주는 것이 주요 기능이다.

## 1.3 Terms of Use

본 문서에서 사용되는 용어 정리

## 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

이 문서는 CFG Generator의 전반적인 기능을 포함하고 있다.

2장 Overall Description 에서는 기본적인 프로젝트의 기능, 사용자의 특성 그리고 개발에 관련 제약사항들을 제시하고. 3장에서는 2장의 내용을 기반으로 Functional Spec 과 Non-Functional

Spec을 제시 할 것이다.

## 2 Overall description

### 2.1 Product Perspective

#### 2.1.1 User interface

UI는 CUI 형태로 제공되며 프로젝트의 수행 방식은 Command Line 명령어 형태로 제공된다. Report는 CFG의 State의 목록과 Edge의 목록을 리스트 형태로 보여주게 된다.

명령어 예시로 다음과 같이 사용 할 수 있다.

```
# ./CG Inputcode.c result.txt
```

### 2.2 Product functions

기본적인 기능은 주어진 C Code를 CFG로 변경 시켜주는 기능을 한다.

### 2.3 User Characteristics

사용자는 기본적으로 Cygwin 환경에서 GCC 컴파일을 할 줄 아는 사람일 것이다. 또한 사용자는 Branch Coverage와 Statement Coverage를 이해 할 것이다. 그들은 CFG의 기본적인 형태를 그릴 줄 할 것이다.

### 2.4 Constraints

- 사용언어 : C
- 구동환경 : Cygwin

## 3 Specific Requirements

### 3.1 External interface requirement

#### 3.1.1 User Interfaces

3.1.1.1 CUI 형태로 Report를 제공한다.

3.1.1.2 Report는 입력된 코드로 생성된 CFG의 모든 State를 표시해 준다.

### 3.2 Functional Requirement

#### 3.2.1 실행

3.2.1.1 소프트웨어의 실행은 Command Line 명령어 형태로 제공한다. 명령어의 예시는 2.1.1을 참조하라.

3.2.1.2 잘못된 형태의 명령어를 입력했을 시에는 도움말을 출력하도록 한다.

#### 3.2.2 Report 생성 과정

3.2.2.1 프로그램의 수행 과정을 CUI로 보여준다.

3.2.2.2 C Code 입력이 성공 하였다면 성공 메시지를 출력한다. 실패 하였다면 실패 메시지를 출력하고 프로그램을 종료한다.

3.2.2.3 C Code를 CFG로 변환한다. 변환을 시작하기 전에 변환의 시작을 사용자에게 알린다.

3.2.2.4 Report를 생성한 후 생성 파일 명(report file)을 출력한 후 마친다.

### 3.3 Performance Requirement

3.3.1 코드의 크기는 100~200줄 내외의 프로그램을 대상으로 한다. Main Function을 포함하는 코드여야 한다.

3.3.2 단일 파일로 되어 있는 코드에 대하여 작동한다. 사용자가 정의한 헤더를 사용한 파일에 대해서는 작동하지 않는다.

3.3.3 포인터를 사용하지 않은 코드를 대상으로 한다.

## 3.4 Design Constraints

3.4.1 SASD 방법론에 따라 프로그램을 개발한다.