소프트웨어공학 개론

Team Presentation #2

Testing & CFG

T4

200511306 김성훈

200710115 김철웅

200712692 진형남

201011316 김성엽

Contents

- Confrol Flow Graph
- CFG Drawing
- Statement of purpose

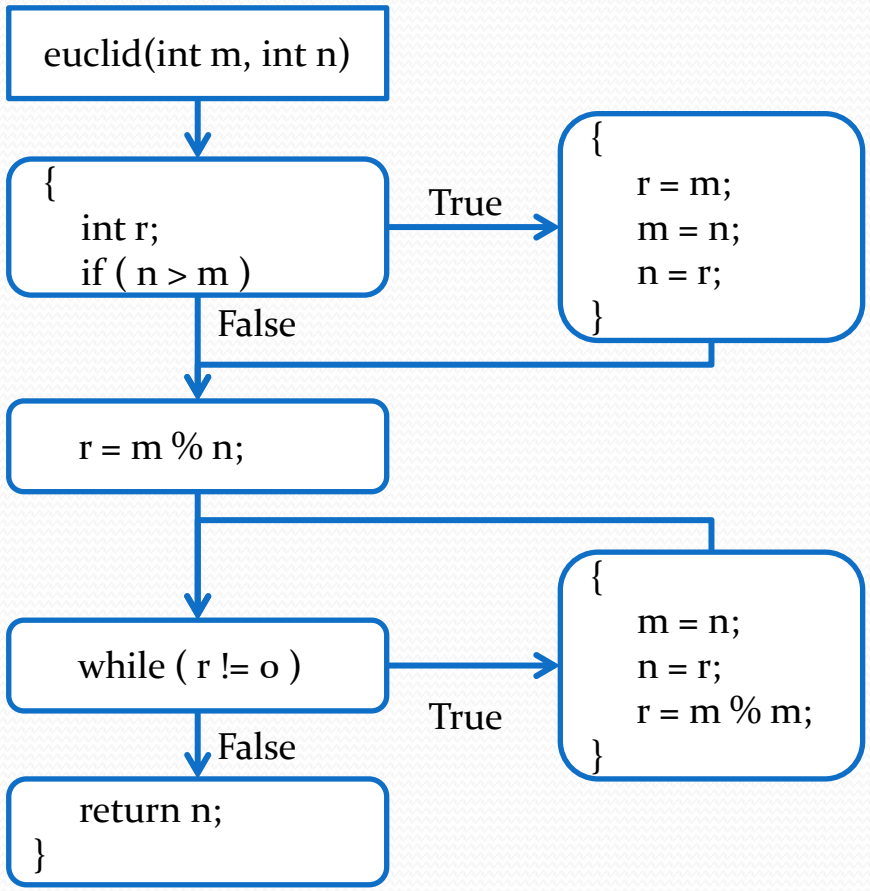
Control Flow Graph

- 제어 흐름 그래프(CFG)
  - 프로그램의 실행과 전체 프로세스를 그래프로 표현
  - 소프트웨어 모듈의 논리적 구조를 표현
  - 모듈은 하나의 입력과 출구 지점을 가짐
  - 호출 / 반환 매커니즘을 통해 구성 요소로 사용 가능
- 이용 분야
  - 프로그램 정적 분석
  - 컴파일러 최적화
  - 다른 모델의 기초
    - LCSAJ(Linear Code Sequence And Jump)

Control Flow Graph

- CFG의 예

```
euclid(int m, int n) {
        int r;
        if ( n > m ) {
                r = m;
                m = n;
                n = r;
        }
        r = m % n;
        while ( r != o ) {
                m = n;
                n = r;
                r = m % n;
        }
        return n;
}
```



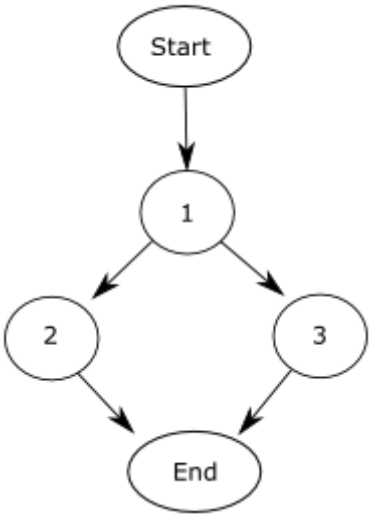
Control Flow Graph

- Graph
  - A directed graph : G(N, E)
- Node
  - 소스 코드의 영역 (Basic block)
  - Basic block : 하나의 입력점과 출력점을 갖는 최대 영역
  - 일반적으로 statement는 하나의 region으로 묶어지나 여러 개의 block으로 쪼개지기도 함
- Directed edges
  - 노드간의 제어 흐름을 표현
  - 하나의 Basic block의 끝에서 다른 Basic block의 시작으로의 프로그램 실행이 진행 가능성

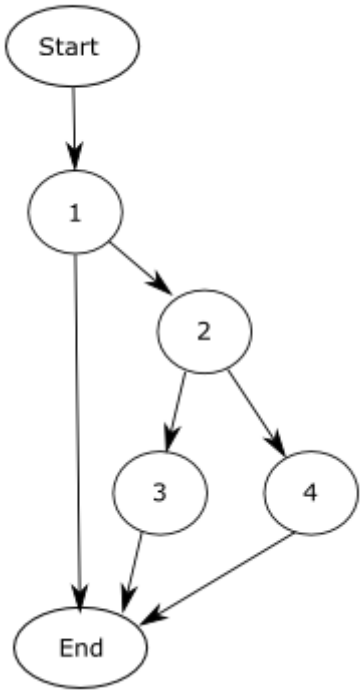
Control Flow Graph

- Loop Flows

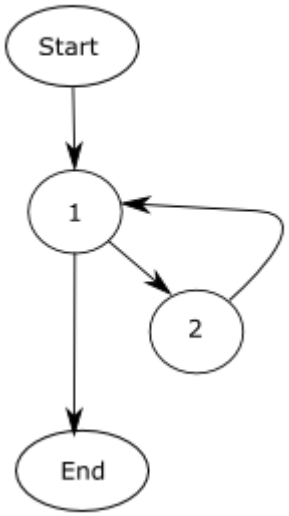
If Loop



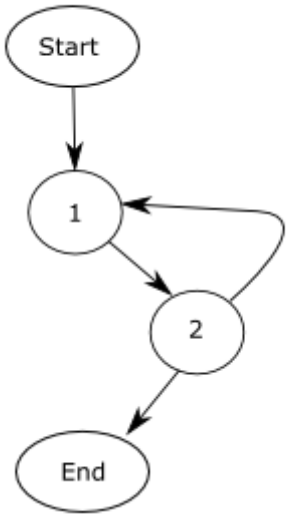
Nested If Loop



While Loop



Do While Loop



Control Flow Graph

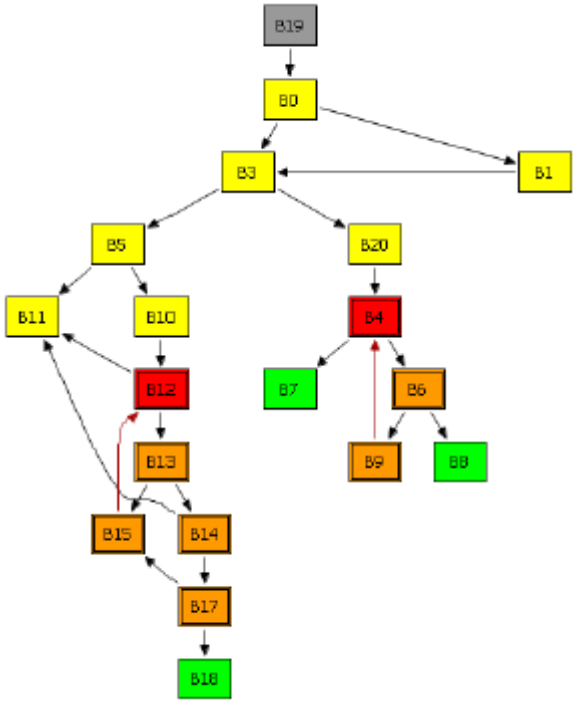
- CFG & Testing
  - Test의 철저함의 기준으로 사용 (Coverage)
    - Statement coverage
      - 모든 block 이 적어도 한 번은 수행되었는지 여부
    - Branch / decision coverage
      - 모든 edge 가 적어도 한 번은 수행되었는지 여부
    - Condition coverage
      - edge를 결정하는 조건의 모든 가능성이 수행되었는지 여부
    - Simple path coverage
      - 모든 edge의 조합들이 수행되었는지 여부

CFG Drawing

- CFG 변환 과정
  - 1. Source code를 Basic block 형태로 분할
  - 2. 프로그램의 실행 가능한 흐름을 Graph 형태로 저장
  - 3. CFG Drawing
    - Positioning Algorithm
      - Block들을 기준에 따라 위치시킴
      - BFS 알고리즘, Loop 알고리즘, Hierarchical 알고리즘
    - Routing Algorithm
      - Edge들을 겹치지 않고 복잡하지 않게 연결
      - Bezier 알고리즘, Manhatten 알고리즘

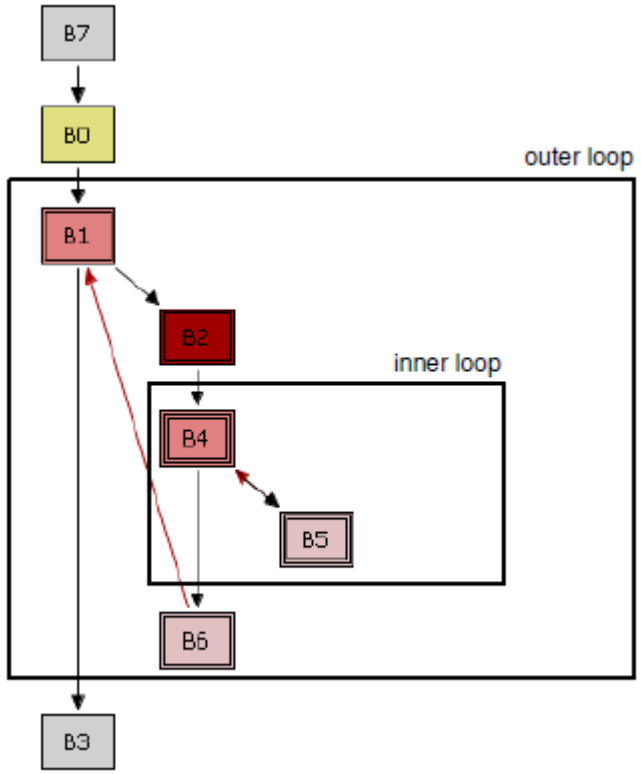
CFG Drawing

- Positioning Algorithm (1/4)
  - BFS Positioning Algorithm
    - Breath First Search, 너비 우선 탐색 이용
    - 1. Cycle을 무시하여 BFS 수행
    - 2. root로부터의 거리로 y 좌표 연산
    - 3. x 좌표 연산
      - no children : fixed value
      - children exist : sum of the widths
      - recursively : middle of children



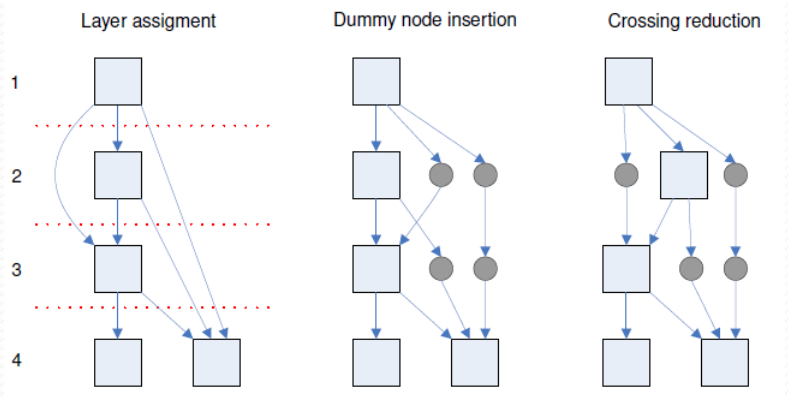
CFG Drawing

- Positioning Algorithm (2/4)
  - Loop Positioning Algorithm
    - 들여쓰기와 유사
    - 1. Loop를 작은 그래프로 묶음
    - 2. Loop의 root는 동일한 위치
    - 3. Loop의 내부는 x 좌표 증가



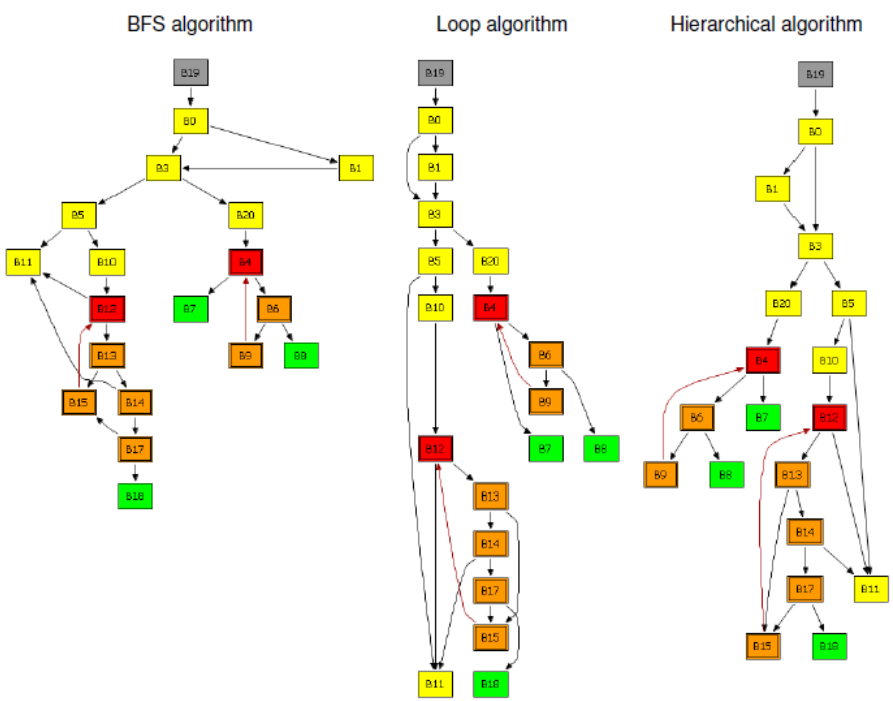
CFG Drawing

- Positioning Algorithm (3/4)
  - Hierarchical Positioning Algorithm
    - 1. Break loops
    - 2. Layer assigment : Loop 알고리즘과 동일
    - 3. Add Dummy node : edge가 연속된 계층을 연결하지 않는다면 dummy node 추가
    - 4. Crossing Reduction : edge들의 교차 개선
    - 5. Horizontal Coordinate Assignment : 노드에 x 좌표 할당



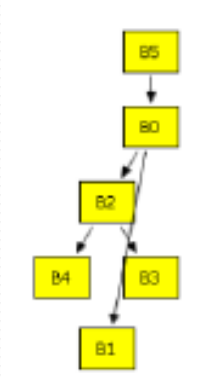
CFG Drawing

- Positioning Algorithm (4/4)
  - BFS : 길이가 짧으나 흐름을 쉽게 파악하기 어려움
  - Loop : 길이가 길지만 Loop 표현이 명확함
  - Hierarchical : Edge들의 교차가 적음

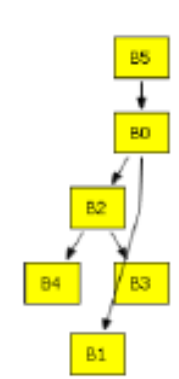


CFG Drawing

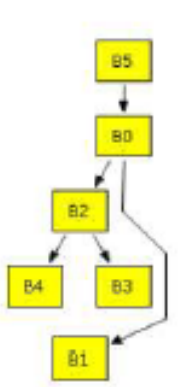
- Routing Algorithm (1/3)
  - Bezier Routing
    - 1. Initial Routing
    - 2. Evade Obstacles
    - 3. Simplify Polygon
    - 4. Create Curves



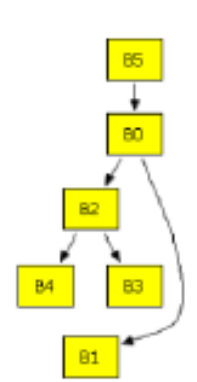
Initial Routing



Evade Obstacles



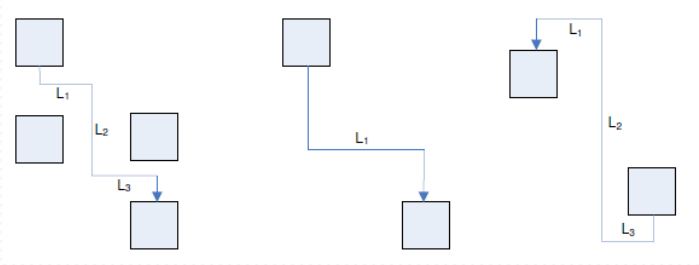
Simplify Polygon



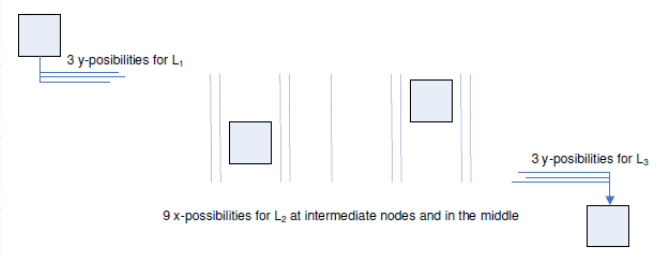
Create Curves

CFG Drawing

- Routing Algorithm (2/3)
  - Manhatten Routing (1/2)
    - 오직 직교 edge만 허용하는 경우
    - 두 노드 사이 최대 4번의 꺽임으로 연결 가능



    - 가능한 수많은 후보들 중 좋은 edge를 선택

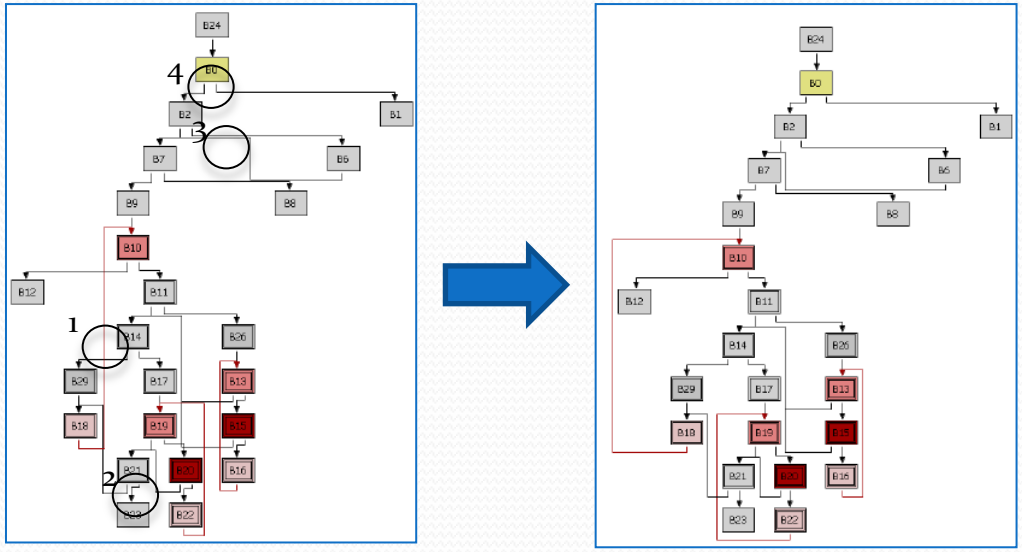


CFG Drawing

- Routing Algorithm (3/3)
  - Manhatten Routing (2/2)
    - 좋은 Edge의 기준

      1. 교차하지 않는 것　　2.꺾임이 적은 것

      3.근접하지 않은 것　　4. 동일한 수평선 상에 위치 (교차하지 않는 경우)



Statement of purpose

- 프로젝트 내용
  - 입력 받은 C 프로그램을 CFG 로 변환하여 효과적으로 출력한다.
  - 입력 : C 프로그램
  - 출력 : CFG (Control Flow Graph)

Statement of purpose

- Purpose (1/2)
  - C 프로그램을 입력받아 CFG로 출력
  - C 언어로 작성된 파일을 읽음
    - 입력 파일의 문법 오류는 고려하지 않음
  - 입력받은 소스 코드를 CFG 표현 형태로 변환
    - 분기가 발생하는 지점을 기준으로 Basic Block으로 분할
    - 프로그램의 실행 가능한 흐름을 Graph 형태로 저장

Statement of purpose

- Purpose (2/2)
  - 변환된 CFG 표현 형태를 이용하여 CFG를 출력
    - 저장된 Basic Block을 Positioning 알고리즘을 이용하여 CFG의 노드를 배치
    - 저장된 흐름을 Routing 알고리즘을 이용하여 CFG의 Edge를 연결
    - BFS / Loop / Hierarchical 알고리즘을 선택적 출력
    - Bezier / Manhatten Routing 알고리즘 선택적 출력