

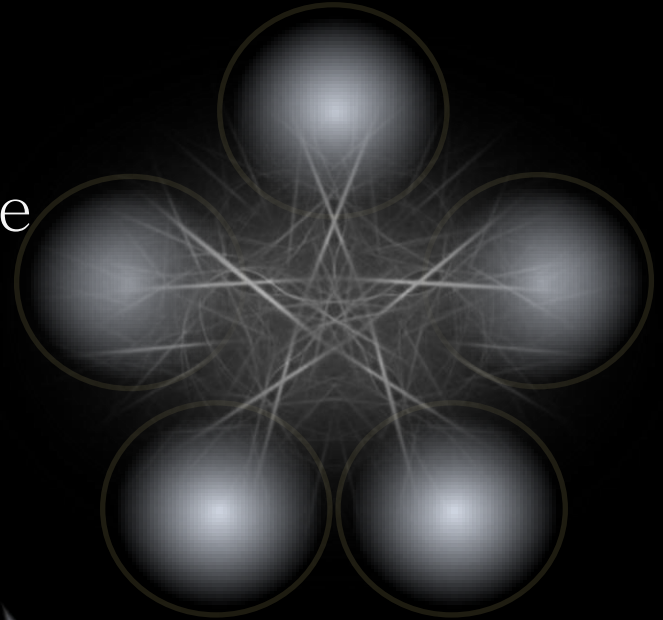
# 소프트웨어 공학개론(B)

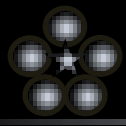
유준범 교수님  
윤상현 조교님

T10 : 발표자 200811436 안정무  
200811437 여종훈  
200811406 권성광  
200811460 최산수

# Index

- ★ Control Flow Graph
- ★ CFG Tool Execute Phase
- ★ Algorithm
- ★ Algorithm Pseudo code
- ★ Statement of Purpose



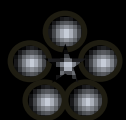


# Control Flow Graph

---

- ★ 컴포넌트나 시스템을 통해 실행될 때 이벤트의 흐름(경로)을 표현한 그래프
- ★ 방향 그래프로 이루어져 있으며 Node와 Edge로 표현이 된다.





# ★ CFG Tool Execute Phase

- ★ COBOL, C, C++, JAVA 등으로 작성된 소스 프로그램을 번역기에 의해 독립적인 중간 언어인 매개 언어로 구축한 후, 매개 언어의 저장 시스템에 저장하는 단계
- ★ 저장된 매개 언어에서 Control Flow Graph 생성에 필요한 함수 정보만을 추출하여 통합 저장 파일에 저장하는 단계
- ★ Control Flow Graph의 생성에 필요한 그래프의 노드 및 간선들의 정보들을 매개 언어로부터 탐색하여 Control Flow Graph의 정보 모형에 저장하는 단계
- ★ 알고리즘에 의해 노드의 순서를 정하고, 간선의 종류를 식별하는 단계
- ★ 완성된 Control Flow Graph를 출력하는 단계

# Algorithm

1. 블록단위로 나누어 블록리스트에 저장한다.
2. 변수들을 변수리스트에 저장한다.
3. 흐름그래프의 첫 번째 노드(노드0)를 생성, 첫 번째 노드는 변수들을 모두 Read하는 content를 가짐
4. 블록들을 모두 흐름그래프의 노드형태로 변환해 나간다.
5. 각 노드를 순서대로 간선을 붙여가며 그래프에 추가한다.
6. 입력값에 따라 출력이 나누어지는 경우, 그 안의 하위 노드들을 생성하여 추가해주고 분기형 노드로 변환하여 그 그래프에 추가해준다.
7. 4-6을 반복하여 모든 블록을 처리한다.
8. 마지막 노드를 생성하여 CFG를 완성한다.

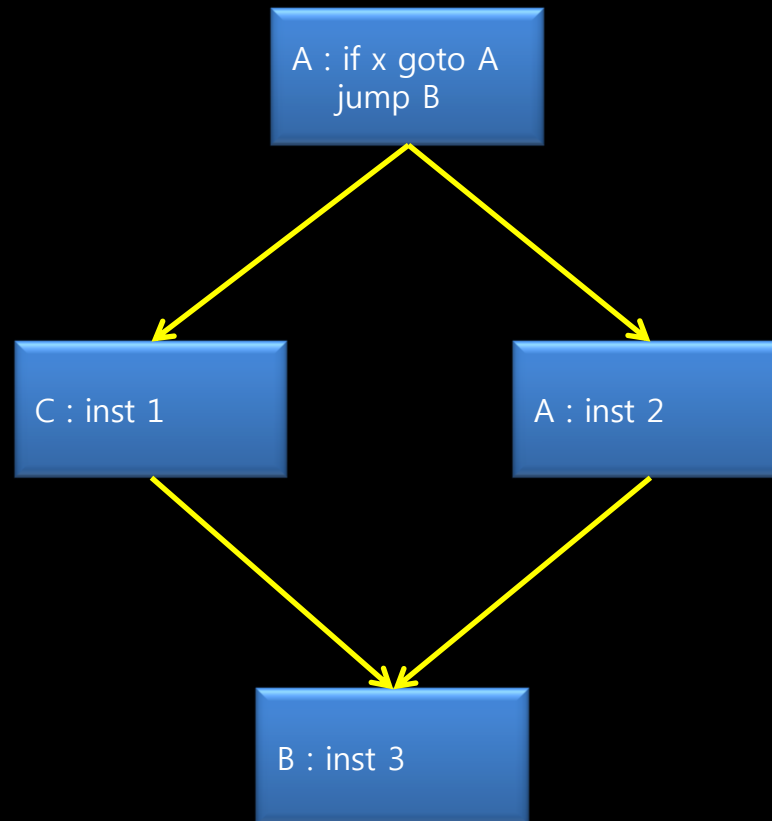


# ★ Algorithm(Cont)

if x  
  then inst 1  
  else inst 2  
inst 3

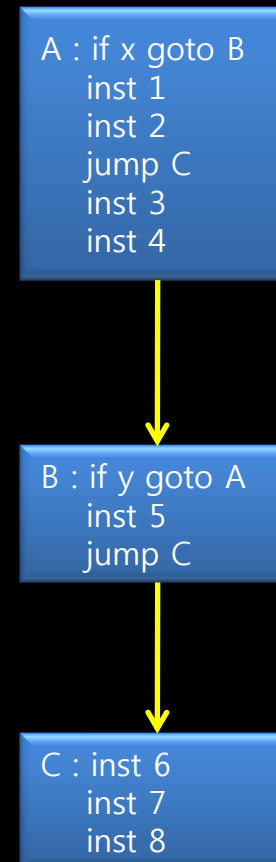


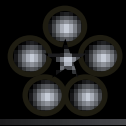
if x goto A  
  jump B  
inst 1  
A: inst 2  
B: inst 3



# ★ Algorithm(Cont)

A: if x goto B  
inst 1  
inst 2  
jump C  
inst 3  
inst 4  
B: if y goto A  
inst 5  
jump C  
C: inst 6  
inst 7  
inst 8





# Algorithm(Cont)

A: if x goto B

inst 1

inst 2

jump C

inst 3

inst 4

B: if y goto A

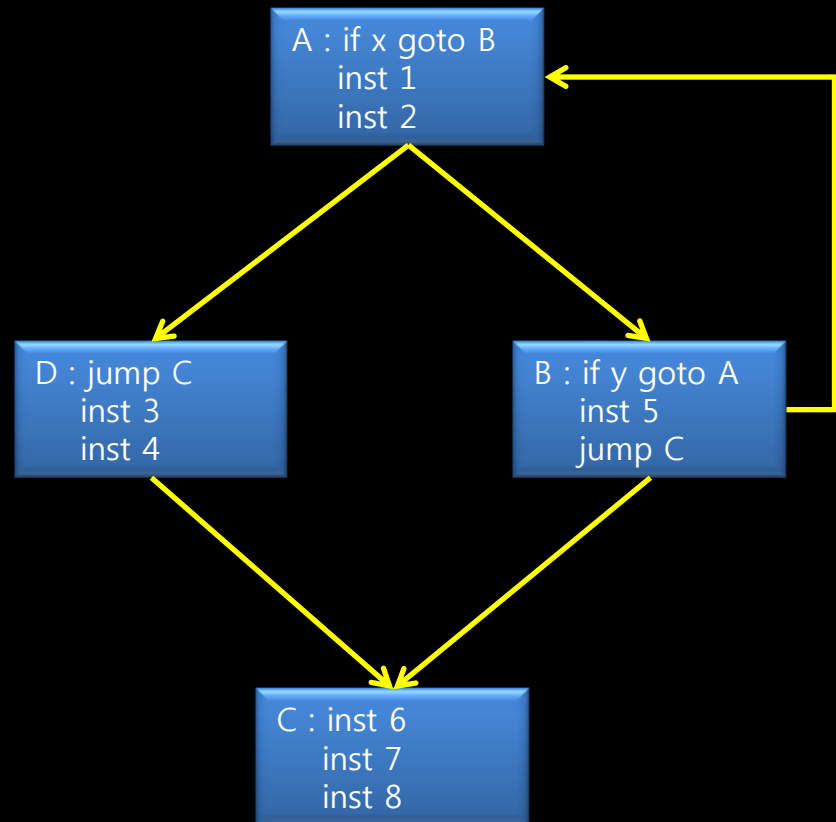
inst 5

jump C

C: inst 6

inst 7

inst 8



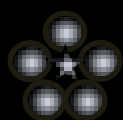




# ★ Algorithm Pseudo Code

```
block list = initial list of blocks
for each block b in block list
    remove b from block list
    branch_found = false
    for each instruction i in b
        if i is a branch
            let branch_found = true
            let countdown = branch-latency
            break
    if branch_found
        for each instruction p in b after
            decrement countdown
            if countdown = 0 break
        if countdown = 0
            split b at p
            let b' = remainder of b
            add b' to block list
            add edges from b to targets of I
            add edges from b to b'
    if not branch_found or countdown > 0
        add edge from b to fallthrough of b
```





# ★ Statement of Purpose

- ★ 소스 프로그램들의 제어 흐름 구조 정보를 분석하여 방향 그래프로 계층적 레이아웃하는 Control Flow Graph(Control Flow Graph)의 자동 생성을 위한 프로그램.
- ★ C언어로 작성된 코드를 중간 매개언어로 구축 후, 매개언어에서 CFG를 생성한다.
- ★ 생성된 CFG는 방향 그래프로 나타내어진다. (각각의 블록을 노드, 진행순서를 방향이 있는 간선으로 표현한다.)
- ★ 블록의 시작노드와 끝노드가 있다.
- ★ 분기형 노드는 하위 노드를 갖고 있으며, 조건에 따라 출력 엣지를 나누어 준다.



★ Thank you ★