SOFTWARE ENGINEERING TEAM PROJECT #2

TEAM : T8 (이자형 김은빈 오고은)

발표자 : 이자형

담당교수님 : 유준범교수님

담당조교님 : 윤상현조교님

KU 건국대학교 KONKUK UNIV.

발표순서

① . CFG 란 ?

② . CFG 생성 ALGORITHM

③ . STATEMENT OF PURPOSE

SOFTWARE ENGINEERING #T8

KU 건국대학교 KONKUK UNIV.

CFG 란?

제어흐름그래프(CONTROL FLOW GRAPH)는 프로그램의 실행이 나타내어진 그래프의 형태이다, 그리고 제어흐름그래프(CONTROL FLOW GRAPH) 와 관련된 모든 과정이 보여지는 기호의 형태이다.

제어흐름그래프(CONTROL FLOW GRAPH) 안의 그래프는 대부분 노드와 선이 연결됨으로 나타낼 수 있다. 제어흐름그래프(CONTROL FLOW GRAPH) 는 또한 EDGE와 방향을 가지고 있고 방향을 가지는 EDGE는 시작점과 끝점이 있다. 시작점은 컨트롤 플로우를 끝가지 수행해서 종료되도록 유도한다.

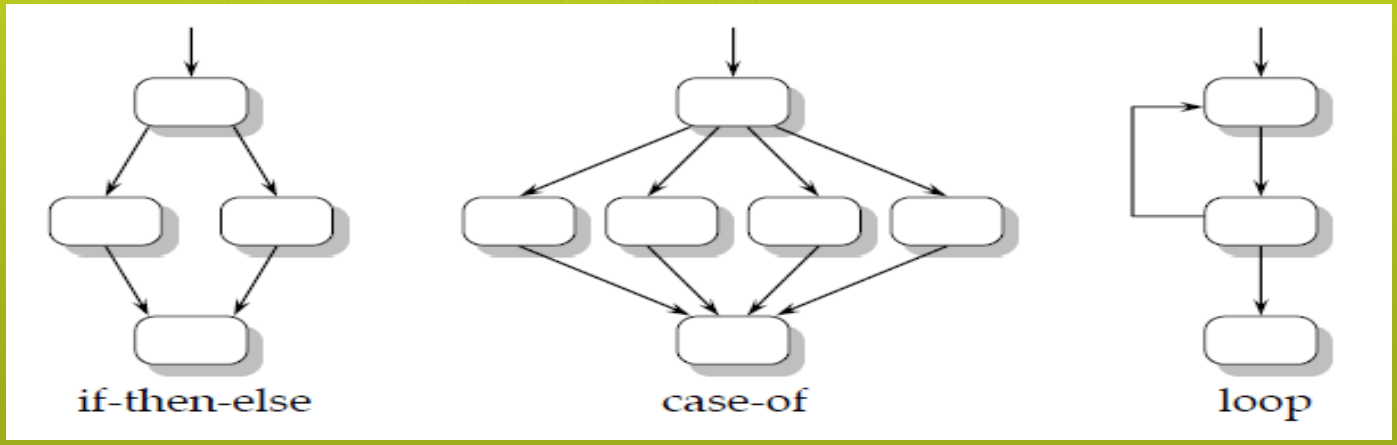
제어흐름그래프(CONTROL FLOW GRAPH) 는 통계 분석과 컴파일러 최적화툴에 사용된다. 그래프는 또한 프로그램 실행시의 REACH ABILITY 최적화를 도와준다. 제어흐름그래프(CONTROL FLOW GRAPH) 는 또한 연결이 되지 않지만 시작점과 끝점이 있는 것도 다룬다. 제어흐름그래프(CONTROL FLOW GRAPH) 는 모든 실행되는 프로그램 부분을 압축하여 설명한다.

KU 건국대학교 KONKUK UNIV.

CFG 란?

CFG = CONTROL FLOW GRAPH (제어 흐름 그래프)

- 프로그램의 제어 구조를 그래프 형태로 나타냄
- 그래프 구성 요소 : 블록(BLOCK), 분기(DECISION)
- 프로그램 제어 구조에 대한 표현 방법



KU 건국대학교 KONKUK UNIV.

CFG 란?

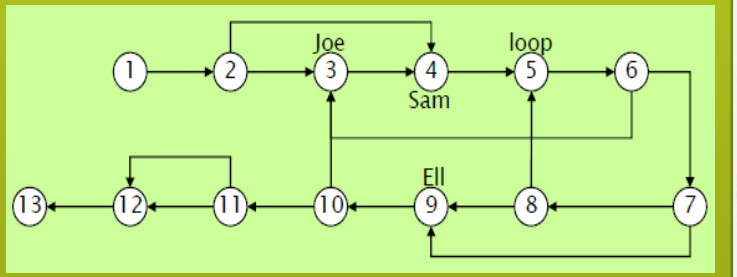
EXAMPLE))

JOE, SAM :

```
INPUT X,Y
1 Z := X+Y
1 V := X-Y
2 IF Z>=0 GOTO SAM
3 Z:= Z-1
4 Z:= Z+V
5 FOR U=0 TO Z
5 V(U), U(V) = (Z+V)*U
6 IF V(U) = 0 GOTO JOE
7 Z : Z-1
8 IF Z = 0 GOTO ELL
9 U := U+1
9 NEXT U
```

ELL :

```
9 V(U-1) : = V(U+1) + U(V-1)
9 V(U+U(V)) := U+V
10 IF U = V GOTO JOE
11 IF U > V THEN U := Z
12 Z := U
13 END
```



KU 건국대학교 KONKUK UNIV.

CFG 생성 ALGORITHM

▶CFG 만다는 알고리즘 4단계

① EDGE 인식

② 기본적인 블록 건설

③ 지연되는 것 해결

④ 데이터 충돌 해결

SOFTWARE ENGINEERING #T8

KU 건국대학교 KONKUK UNIV.

CFG 생성 ALGORITHM

① EDGE 인식

처음에 우리는 명령에 부합하는 NODE를 그래프에 나타내는 CFG를 만들 것이다. 두번째로 연이은 상태의 NODE의 집합들을 BASIC BLOCK으로 합칠 것이다. 결과적으로 보통형태의 CFG를 얻을 것이다.

EDGE를 알아보는 알고리즘에 의해 수행된 분석은 프로그램의 모든 PATH를 통해 이동한 것과 같다.

② 기본적인 블록 건설

CFG를 구조화 시키기 위한 알고리즘이 주어지면 프로그램언어의 집합체들은 PATH들에 의해 지연 되고 만다. 그것들의 알고리즘생성은 우리들것과는 다른 양상을 보인다. 왜냐하면 PATH는 프로그램에 남아 지연시키고 있기 때문이다.

프로그램은 필요한 제어흐름에서 입력된 베이직블록으로부터 간단히 분리되어있다. 우리의 프로그램을 발전시키려는 접근은 PATH의 지시는 지연 되지않고 자리는 프로그램분석을 좀더 잘 처리할 수 있게 된다.

SOFTWARE ENGINEERING #T8

KU 건국대학교 KONKUK UNIV.

CFG 생성 ALGORITHM

③ 지연되는 것 해결
CFG가 만들어 졌으면, 그것은 비교적으로 다른 설명의 지연들보다 간단하다.
만약에 새로운 위치에서 지연이 발생하면, 움직인 지시는 새로운 곳에 삽입된다.
만약에 새로운 위치에서 지연이 발생되지 않는다면, I는 적절한곳에 삽입된다.(고려해라 P(위치)에있는 I를)
만약에 지연된 SLOT들이 충분히 크고 길다면, I는 계승하는것을 전파 시킬 것이다.

④ 데이터 충돌 해결
명령이 프로그램안의 다른 위치로 이동할때, 충돌을 방어할수 있는 자리로 옮기는것이 중요하다.

KU 건국대학교 KONKUK UNIV.

STATEMENT OF PURPOSE

위 프로그램은 **C언어 소스코드**를 받아들여, 그 소스코드를 **분석**하고, 그것을 토대로 **CONTROL FLOW GRAPH를 생성하는 알고리즘** (CFG CONSTRUCTION ALGORITHM)을 가동해 알맞은 **CONTROL FLOW GRAPH를 그리는 것**이 목표이다.

CONTROL FLOW GRAPH를 생성하는 알고리즘(CFG CONSTRUCTION ALGORITHM)은 EDGE인식, 기본적인 블록건설, 지연되는 것 해결, 데이터 충돌 해결로 총 **4단계**로 이루어지며 이는 **블록과 분기**를 이용해 **프로그램 제어구조를 그래프형태**로 알맞게 그려내는 것을 목표로 한다.

KU 건국대학교 KONKUK UNIV.

END ^.^

조교님 아잉 ♥

SOFTWARE ENGINEERING #T8