Implementation of Interprocedural Program Slicing based on System Dependence Graph

동양대학교 Software Security Lab
노상훈.신승철

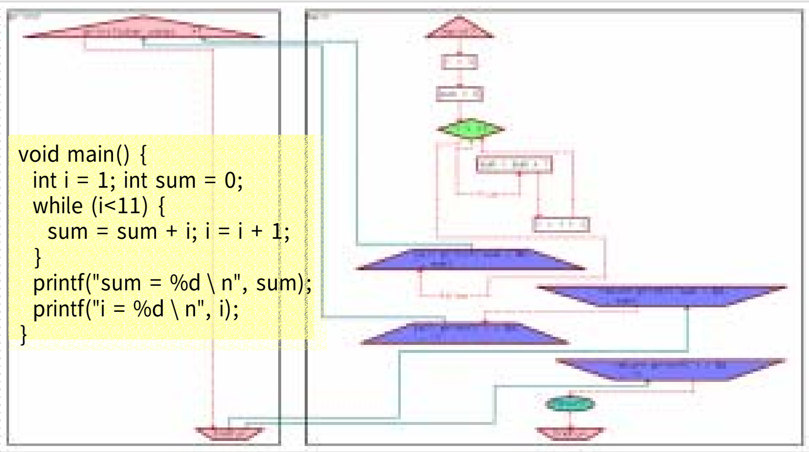
Software Security

목차

CHOPSTICK

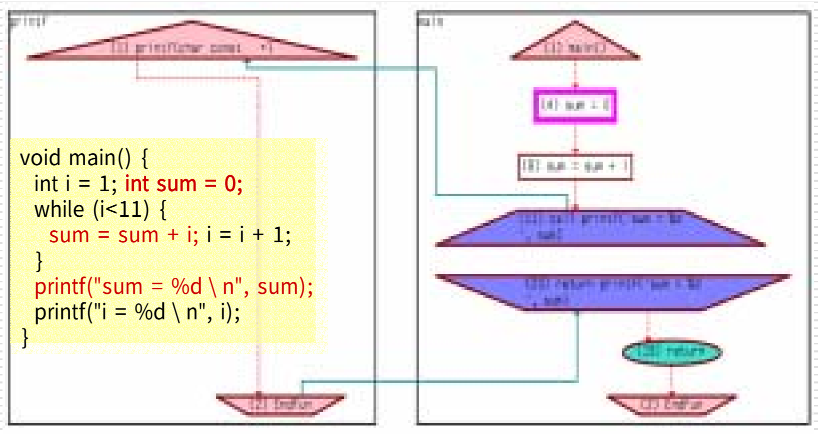
- Program Slicer for C
- Based on SDG
- Forward/Backward slicing and Chopping

Program Slicing – Source



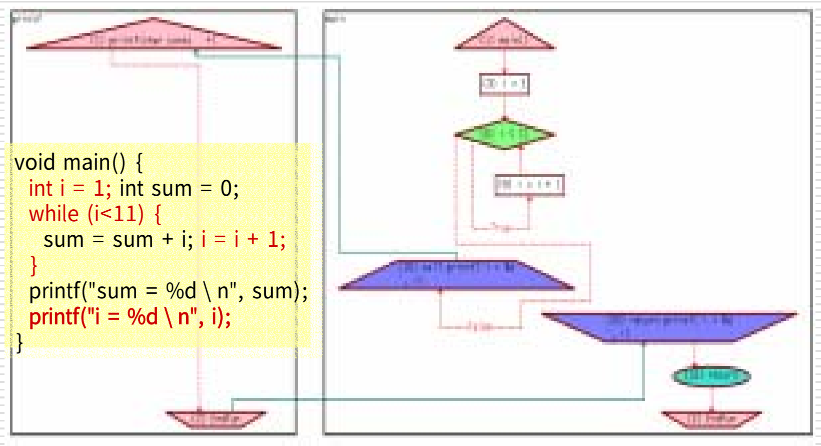
```
void main() {
  int i = 1; int sum = 0;
  while (i<11) {
    sum = sum + i; i = i + 1;
  }
  printf("sum = %d \ n", sum);
  printf("i = %d \ n", i);
}
```

Program Slicing – Forward



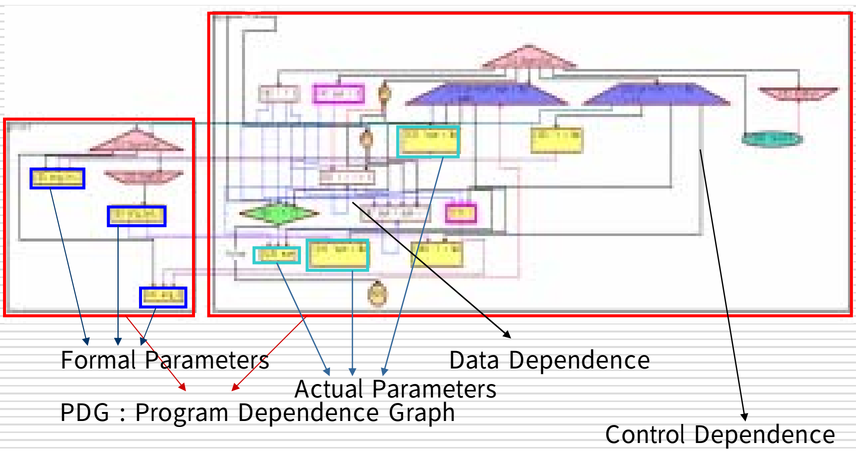
```
void main() {
  int i = 1; int sum = 0;
  while (i<11) {
    sum = sum + i; i = i + 1;
  }
  printf("sum = %d \ n", sum);
  printf("i = %d \ n", i);
}
```

Program Slicing – Backward



```
void main() {
  int i = 1; int sum = 0;
  while (i<11) {
    sum = sum + i; i = i + 1;
  }
  printf("sum = %d \ n", sum);
  printf("i = %d \ n", i);
}
```

System Dependence Graph
Program Dependence Graph



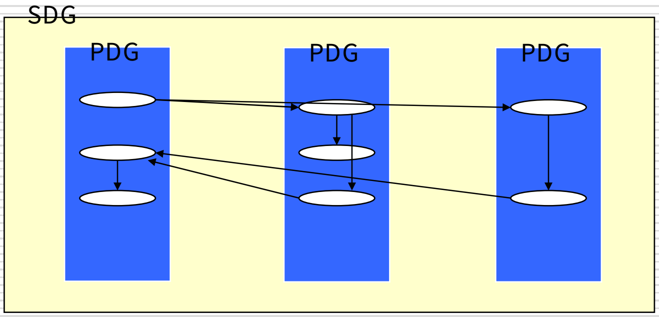
Data Structure of Graph : Past

- 일반적인 Variants type을 통한 AST의 표현 형태
- Vertex와 edge의 관계가 너무 긴밀
- 임의적인 graph 조작이 불편

```
If(BinOp (Gt,
   Lval
    (Var{...},
     NoOffset),
   Const (CInt64 (0L, IInt, None)),
   TInt (IInt, [])),
            ...
```

Data Structure of Graph : Current

- AST를 문장 단위(vertex)로 분할 하여 ID 부여, vertex에 adjacent list를 첨부
- 효율성을 위해 PDG 단위로 graph 분할



Data Structure of Graph : Current

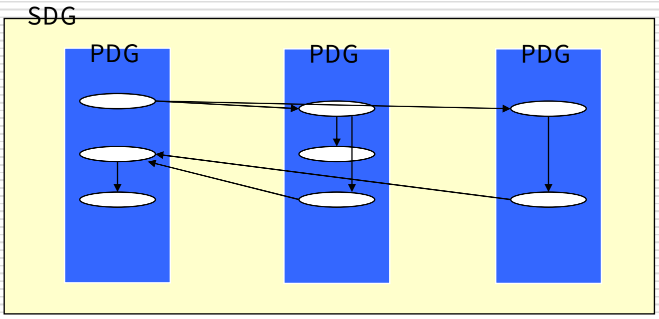
```
type cdsNodeType = {
            ntype: nodeType;
            mutable cfsucclist : cdsNodeListType;
            mutable cdsucclist : cdsNodeListType;
            ...
}

type cdsInfo = {
            funName :string;
            funAst :cdsNodeType IntMap.t;
            ...
}

let cds:cdsInfo list = ...
```

Data Structure of Graph : Current

- Vertex를 공유하는 multi graph 다루기 불편
- Interprocedural vertex를 가리키기에 불편

Data Structure of Graph : Proposal

- 상수 vertex를 통한 안전한 공유
- 각 graph는 edge만을 추가적으로 관리



Data Structure of Graph : Proposal

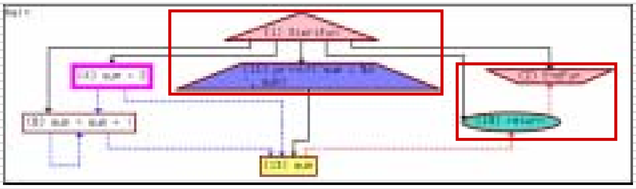
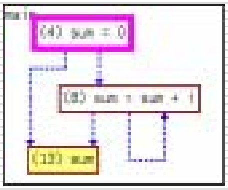
```
module NodeMap = Map.Make(Node);
module NodeSet = Set.Make(Node);

type 'a edges = 'a NodeMap.t NodeMap.t

module Graph = sig
          val nodes: NodeSet.t
          val edgemap: bool edges
          …
          end
```

Restore effective nodes

- Slice정보에 일부 필요 node가 비포함 됨
  - Function call의 parameter 분리 문제(FS)
  - Criteria 전후 무시 문제
- Slice정보에 적절한 node 복원 필요

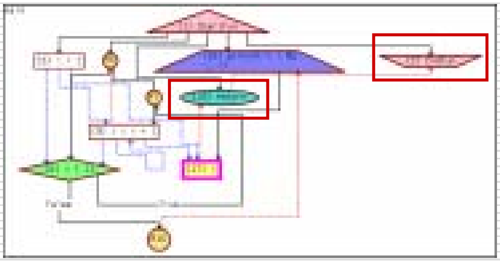
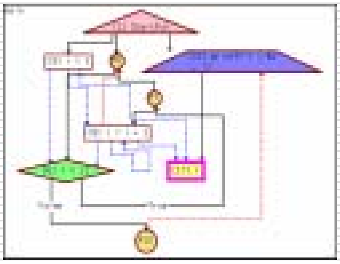
Restore effective nodes
: Forward Slicing

- Criteria 이전 node가 모두 비포함
- Function call에서 parameter만 남음
- Dependence 없을 시 exit node 비포함

Restore effective nodes
: Backward Slicing

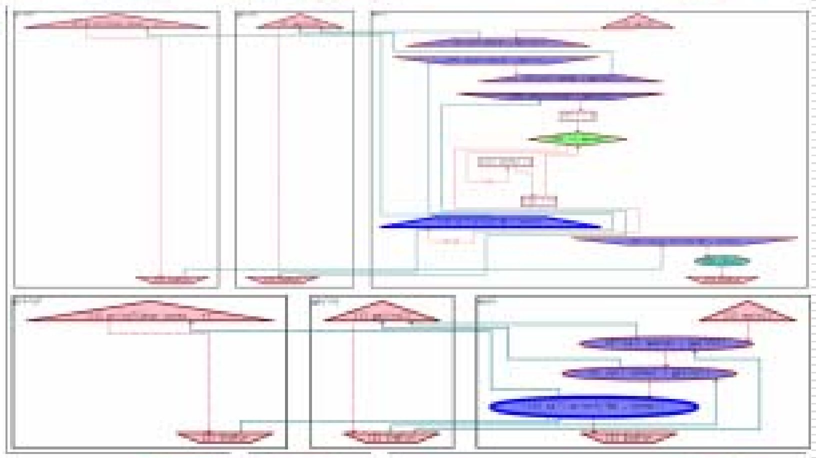
- Criteria 이후 node가 모두 비포함

Application: Flower

- 계산 정보 보다 함수 호출 정보가 중요
- Functional Information Flow Graph
- Verification

Application: Flower

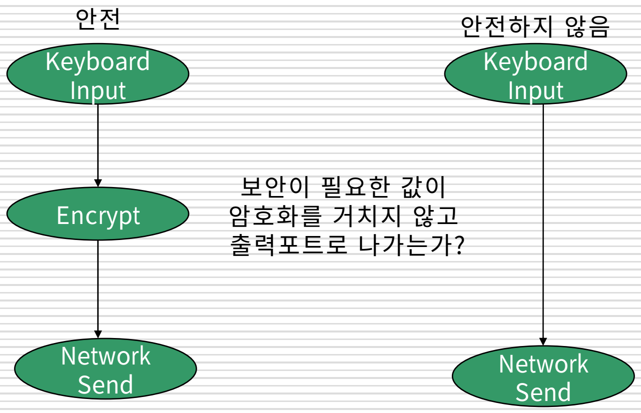


Future work

- Machine code 대상의 분석기 구축
- 경험 바탕의 개선된 Framework 구축
- 실제 사용 가능한 Applications 구현

Question?

Application: Flower - Verification



보안이 필요한 값이
암호화를 거치지 않고
출력포트로 나가는가?

Control Dependent

- N이 M의 CFG successors를 postdominates하되, 전부를 postdominates하지는 않는다. (not strictly postdominate)
- N으로의 제어 흐름이 M의 실행 결과에 종속된다.

Slicing Algorithm: Backward

- Criteria로부터 parameter-out dependence를 <u>제외한</u> dependence를 역방향으로 따라가며 node를 수집
- 수집된 node로부터 parameter-in dependence와 call dependence를 <u>제외한</u> dependence를 역방향으로 따라가며 node를 수집

Slicing Algorithm: Forward

- Criteria로부터 parameter-in dependence와 call dependence를 <u>제외한</u> dependence를 순방향으로 따라가며 node를 수집
- 수집된 node로부터 parameter-out dependence를 <u>제외한</u> dependence를 역방향으로 따라가며 node를 수집

참고문헌

- [1] M. Weiser. Program Slicing. IEEE Transactions on Software Engineering, vol. 10, no. 4, July 1984, pp. 352-357.
- [2] Horwitz, S., Reps, T., and Binkley, D., Interprocedural slicing using dependence graphs. ACM Transactions on Programming Languages and Systems 12, 1 (January 1990), 26-60
- [3] Kumar, S. and Horwitz, S., Better slicing of programs with jumps and switches. In Proc. of FASE 2002: Fundamental Approaches to Softw. Eng., (Grenoble, France, April 8-12, 2002).
- [4] Reps, T., Horwitz, S., Sagiv, M., and Rosay, G., Speeding up slicing. In SIGSOFT '94: Proceedings of the Second ACM SIGSOFT Symposium on the Foundations of Software Engineering, (New Orleans, LA, December 7-9, 1994), ACM SIGSOFT Software Engineering Notes 19, 5 (December 1994), pp. 11-20
- [5] M. J. Harrold and G. Rothermel. Syntax-directed construction of program dependence graphs. Technical Report OSU-CISRC-5/96-TR32, The Ohio State University, May 1996.