PLRG
Programming Language Research Group

루프 민감 분석을 통한 자바스크립트 정적 분석 성능 및 정확도 올리기

박 창 희

공동 연구 : 류석영 교수님

카이스트 프로그래밍 언어 연구실

2015년 1월 27일

개 요

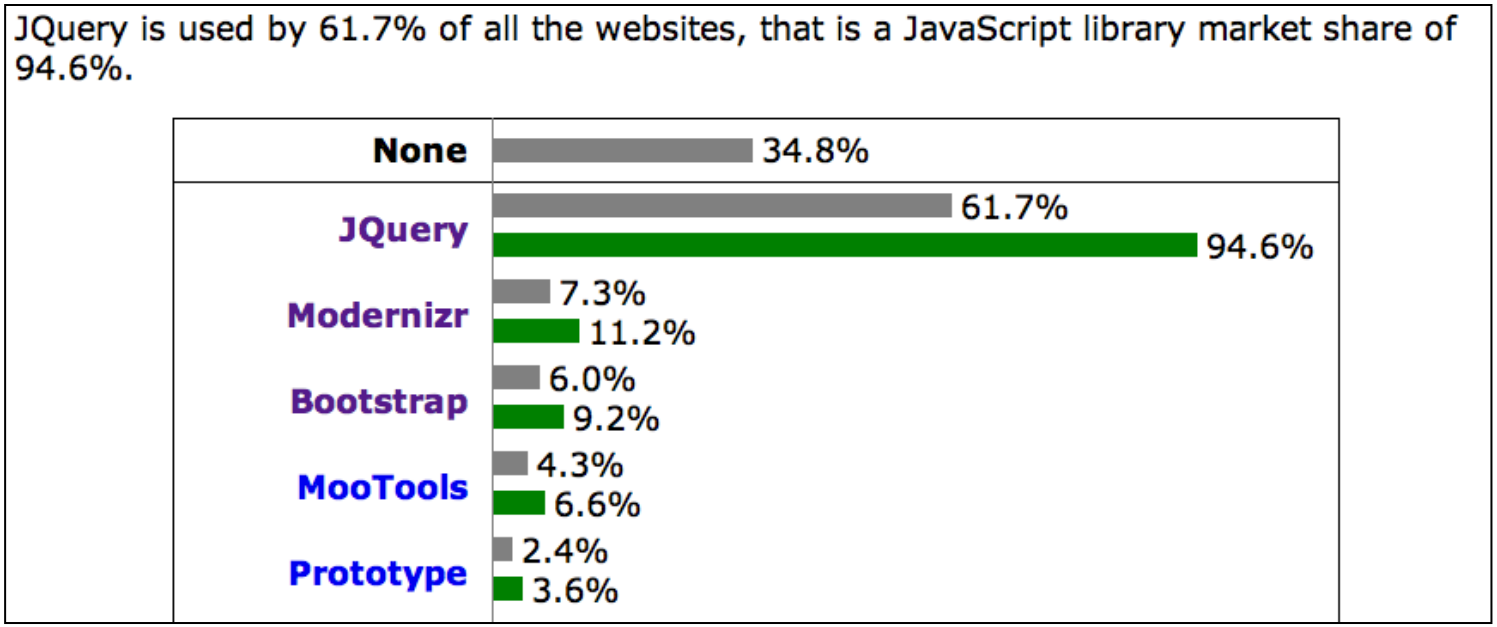
PLRG
Programming Language
Research Group

- 연구 동기
  - 자바스크립트 라이브러리 정적 분석 시 문제점
- 루프 민감 분석이란?
  - 예제 중심으로 아이디어 설명
  - 엄밀한 정의 (formalization)
  - 안전성과 정확성 증명
- 실험 결과
  - 분석 성능
  - 분석 정확도
- 결 론

연구 동기

PLRG
Programming Language Research Group

- 자바스크립트 라이브러리



출처 : http://w3techs.com/technologies/overview/javascript_library/all (2015. 1. 15)

```
<script src="jquery-2.1.1.js"> </script>
```

연구 동기

PLRG
Programming Language Research Group

- SAFE 분석기 실험 (1시간 시간 제한)

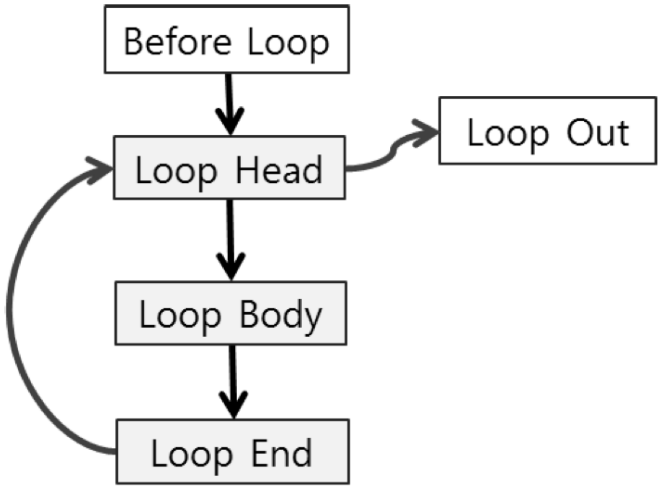
```
<script src="jquery-2.1.1.js"> </script>
```

라이브러리	소스 라인 수	기본 모드 (1-CFA)	2-CFA ~ 10-CFA	Object sensitivity
1, jQuery	9,190	X	X	X
2, Modernizr	1,406	O	O	O
3, Bootstrap	2,267	X	X	X
4, Mootools	6,595	X	X	X
5, Prototype	7,509	X	X	X

jQuery 분석 하기

PLRG
Programming Language Research Group

```
jQuery.extend = function() {
   var options, name, copy, ...
       target = arguments[0] ...
       i=1, length = arguments.length ...
   if(i === length) {
      target = this;
      i--;
   } ...
   for(; i < length; i ++) { ...
       options = arguments[i] ...
       for (name in options) { ...
          copy = options[name] ...
          target[name] = copy ...
       } ...
    }
}
jQuery.extend({expendo: ... ,
                  each: ... });
jQuery.each(...);
```



name : 모든 25개 property 이름
copy : 25개 property 뭉쳐진 값
jQuery.extend : 25개 property 뭉쳐진 값
jQuery.each : 25개 property 뭉쳐진 값

총 21개 메소드 호출이 분석됨

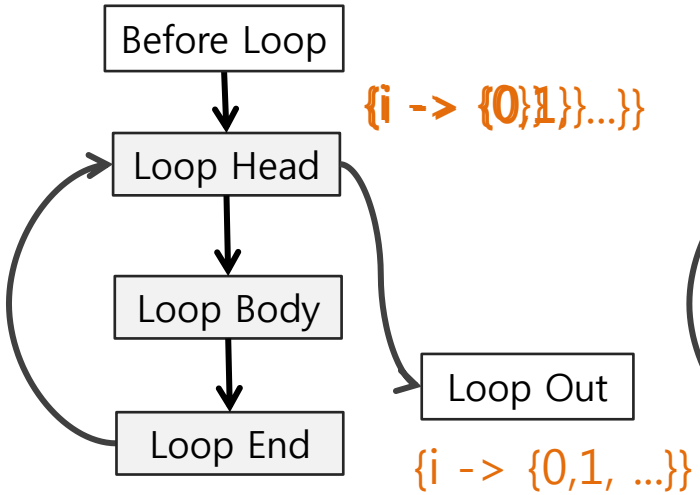
jQuery.expendo
jQuery.each
4개 필드 + 21개 메소드 =
총 25개 property

루프 민감 분석

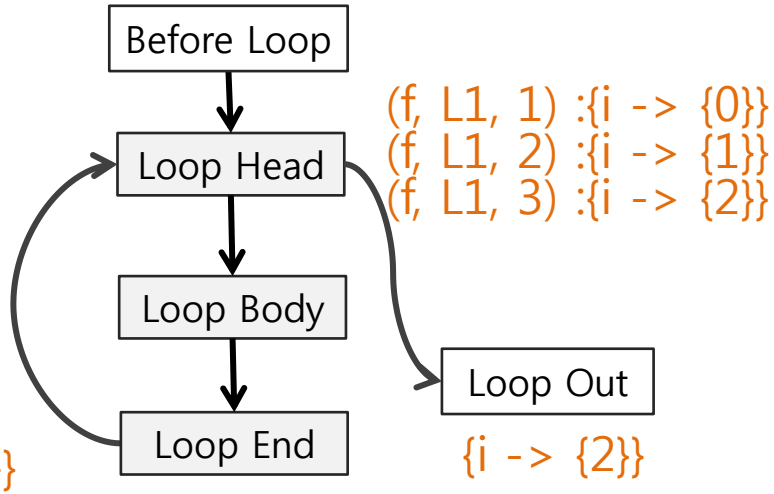
PLRG
Programming Language Research Group

- 루프의 정확도 올리는 기법
- 루프 조건이 확실한 값일 때 루프 정보를 나타내는 루프 스트링으로 각 iteration의 분석 값 구분

```
function f() {
  var i = 0;
  while(i<2) // L1
    i++;
}
f();
```



일반 분석



루프 민감 분석

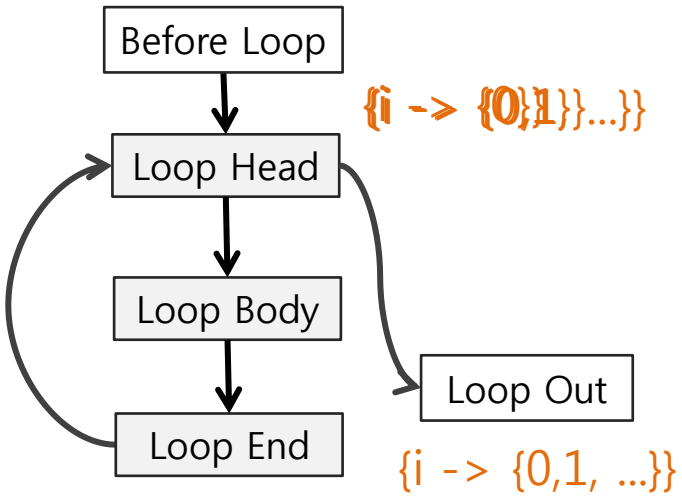
분석 도중 필요한 만큼만 루프 푸는 (loop unrolling) 효과

루프 민감 분석

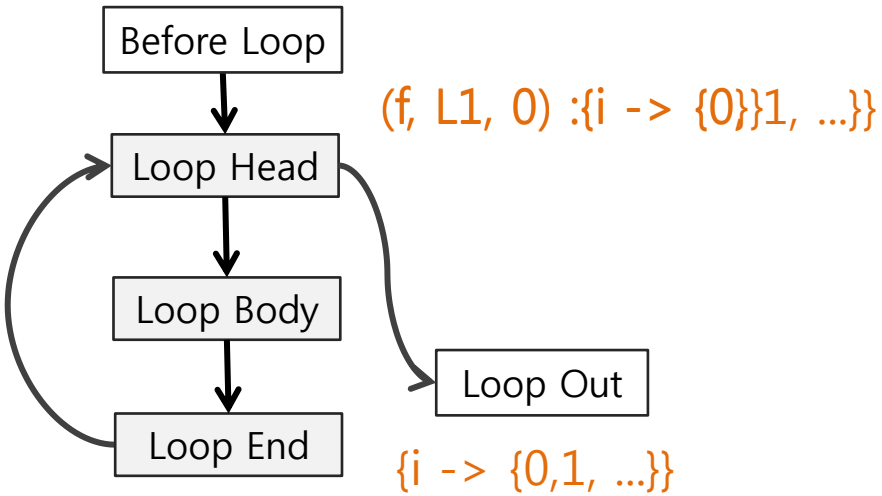
PLRG
Programming Language Research Group

- 루프 조건이 확실하지 않을 때는 iteration 구분하지 않음

```
function f() {
  var i =0
  while(i<Math.random()) // L1
    i++;
}
f();
```



일반 분석



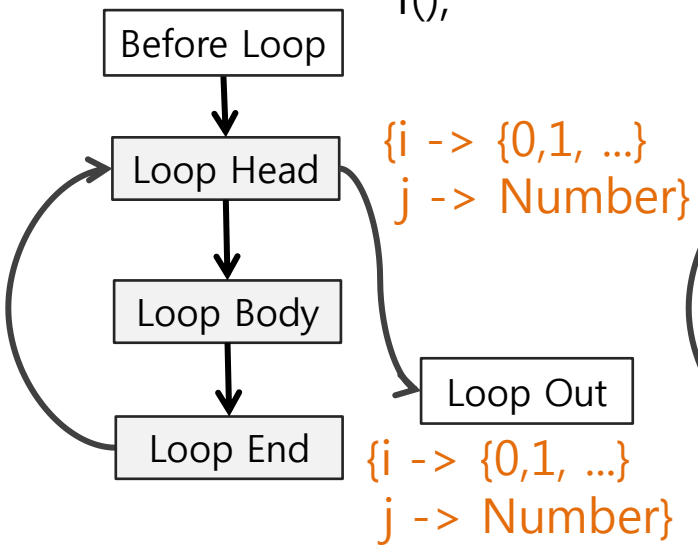
루프 민감 분석

루프 민감 분석

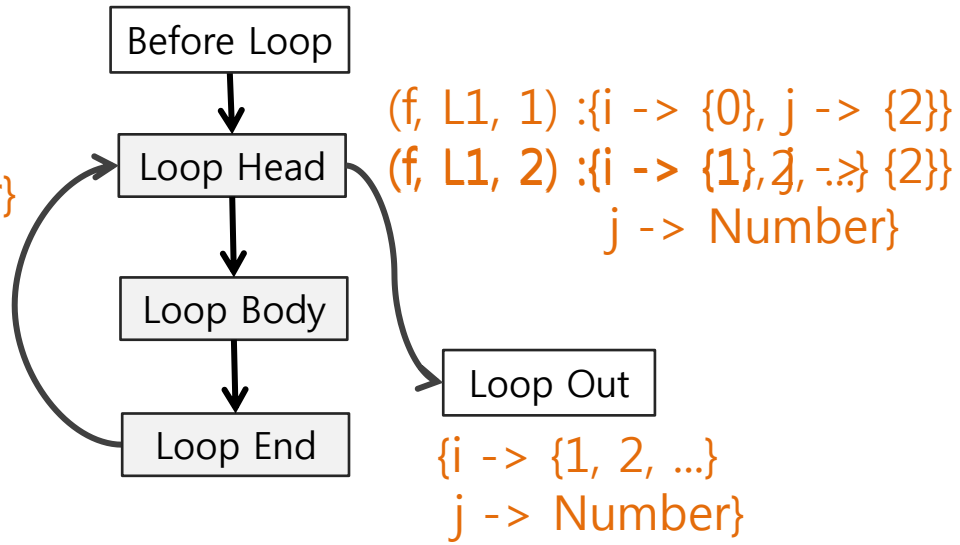
PLRG
Programming Language Research Group

- 루프 조건이 루프 도중 확실하지 않게 변할 때

```
function f() {
  var i =0, j=2;
  while(i<j) {   // L1
     if(i==1) j=Math.random();
     i++;
  }}
f();
```



일반 분석

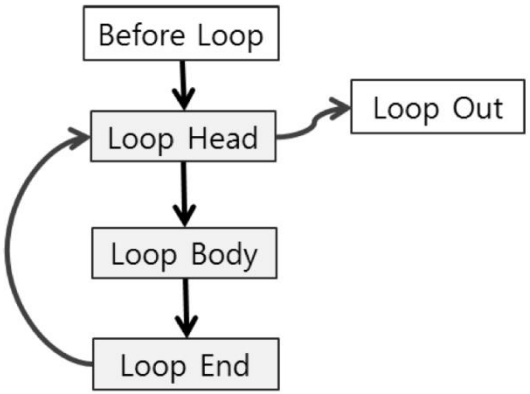


루프 민감 분석

루프 민감 분석

PLRG
Programming Language Research Group

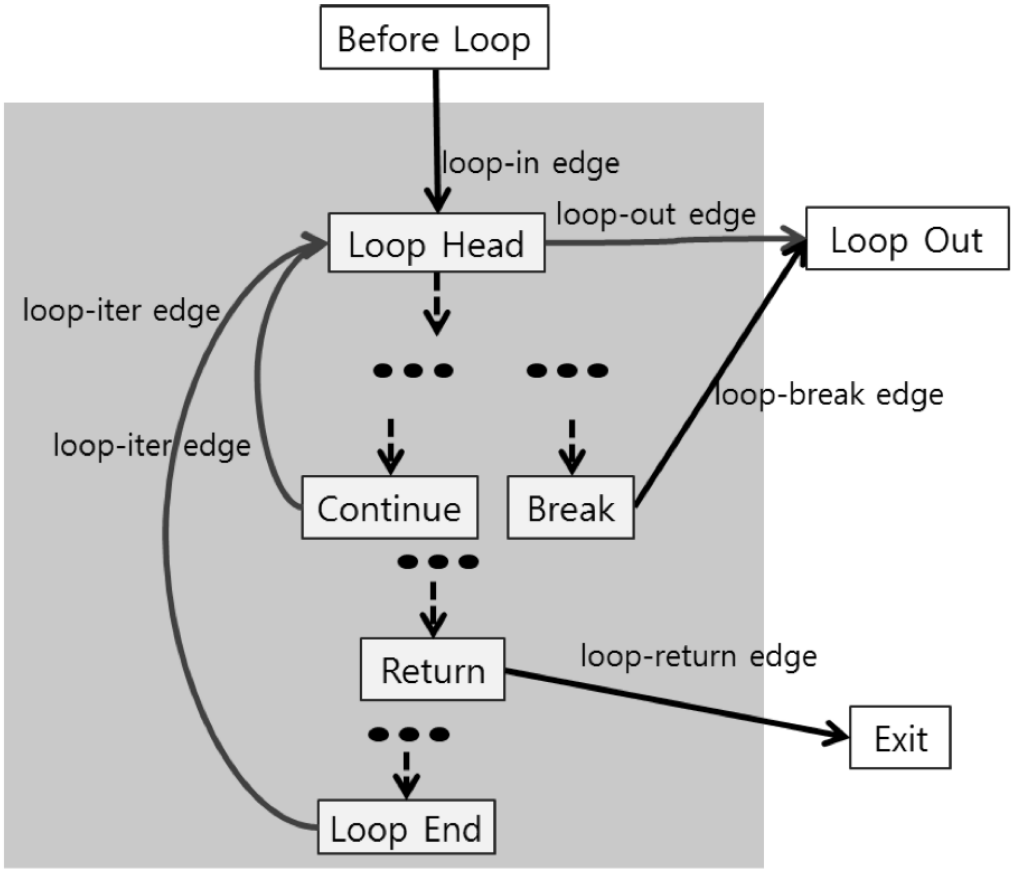
- 요약 해석 (AI) 틀에서 엄밀하게 정의
- 요약 해석 틀에서 잘 정의된 k-CFA 분석 도구가 있다고 가정
- k-CFA 분석 도구를 확장하여 루프 민감 분석 도구 정의
  - 프로그램: 흐름 제어 그래프(CFG)로 표현
  - 모든 루프는 while(exp) ... 로 변환 가능하고 아래의 CFG 형태를 따름



루프 민감 분석

PLRG
Programming Language Research Group

- 루프 관련 CFG 노드 및 엣지



루프 민감 분석

PLRG
Programming Language Research Group

- Loop-iter 엣지 분석 전이 함수 (transfer function)

$$\hat{F}_{\mathsf{lsa}}(\hat{\kappa}_l)(\langle e, \langle \pi, \psi \rangle\rangle) \quad \text{if} \quad loopIterEdge(e) \ \wedge \psi = \langle \pi, target(e), m \rangle \oplus \psi_1 \ =$$

$$\begin{cases} [\![stmt(e)]\!](\hat{\tau}_1) \sqcup [\![stmt(e)]\!](\hat{\tau}_2) & \text{if} \ \ 2 \le m \le j-1 \ \wedge \ \forall \langle \pi_2, n_{lhead_2}, m_2 \rangle \in \psi_1 : m_2 \neq 0 \ \wedge \\ & \hat{\tau}_1 = \hat{\kappa}_l(\langle origin(e), \langle \pi, \langle \pi, target(e), m-1 \rangle \oplus \psi_1 \rangle\rangle) \ \wedge \\ & \hat{\tau}_2 = \hat{\kappa}_l(\langle origin(e), \langle \pi, \psi \rangle\rangle) \ \wedge \ check(stmt(e), \hat{\tau}_1) = \top_b \\ [\![stmt(e)]\!](\hat{\tau}) & \text{if} \ \ 2 \le m \le j-1 \ \wedge \ \forall \langle \pi_2, n_{lhead_2}, m_2 \rangle \in \psi_1 : m_2 \neq 0 \ \wedge \\ & \hat{\tau} = \hat{\kappa}_l(\langle origin(e), \langle \pi, \langle \pi, target(e), m-1 \rangle \oplus \psi_1 \rangle\rangle) \ \wedge \\ & (check(stmt(e), \hat{\tau}) = \hat{\mathsf{true}} \ \vee \ check(stmt(e), \hat{\tau}) = \hat{\mathsf{false}}) \end{cases}$$

루프 민감 분석: 안전성 및 정확성

PLRG
Programming Language Research Group

- 안전성 (soundness)
  - 기존 안전한 k-CFA 분석 도구에서 루프 민감 분석 도구로의 확장은 안전성을 유지함
- 정확성 (precision)
  - 루프 민감 분석 도구는 기존 k-CFA 분석 도구에 비해 같거나 더 정확한 분석 값을 줌
- 증명 보조 기구 Coq 으로 증명 완료

실 험

PLRG
Programming Language Research Group

- 분석 성능 비교 실험
  - ➢ 비교 대상 분석기
    - 기존 분석기: 최신 버전의 SAFE, TAJS, WALA
    - $SAFE_{lsa}$: 루프 민감 분석 확장 버전의 SAFE
  - ➢ 분석 대상 프로그램

Group (Number of programs)	
jQuery 1.0.0~2.1.0	(14)
Modernizr 2.8.2	(1)
BootStrap 3.3.0	(1)
Mootools 1.5.1	(1)
Prototype 1.7.2	(1)
BENCH	(61)
SLICE	(61)

라이브러리 로드만 하는 프로그램

```
<script src="jquery-2.1.1.js"> </script>
```

jQuery 튜토리얼 예제 프로그램

BENCH와 같지만, jQuery 라이브러리에서 사용하지 않는 부분 삭제 (slicing) 한 프로그램

실 험

PLRG
Programming Language
Research Group

- 분석 성능 비교 실험
  - 비교 방법 : 10분 동안에 분석이 끝나는지 검사
  - 결 과

Group (Number of programs)	Success			
	SAFE	$SAFE_{lsa}$	TAJS	WALA
jQuery 1.0.0~2.1.0 (14)	0	14	11	3
Modernizr 2.8.2 (1)	1	1	1	1
BootStrap 3.3.0 (1)	0	0	0	0
Mootools 1.5.1 (1)	0	1	0	0
Prototype 1.7.2 (1)	0	1	0	0
BENCH (61)	0	25	19	0
SLICE (61)	0	41	32	0

실 험

PLRG
Programming Language Research Group

- 분석이 끝나지 않는 이유: 루프 뭉치는 부분 발생
  - 정적으로 알 수 없는 값
    - jQuery.now(), new Date()
    - Math.random()
  - 이벤트 분석의 한계점
    - 가능한 이벤트 발생 시나리오 모두 고려
    - event.target : 이벤트가 처음 발생한 HTML 요소

실 험

PLRG
Programming Language Research Group

- 분석 정확도 향상
  - ➢ 결 과

Library	SAFE				$SAFE_{lsa}$			
	time (s)	MaxCALL (#)	CALL (%)	PROP (%)	time (s)	MaxCALL (#)	CALL (%)	PROP (%)
jQuery 2.1.1	timeout	23	91.67	50.00	26.40	2	99.73	83.93
Modernizr 2.8.2	53.37	49	97.24	53.33	5.45	2	97.10	100.00
BootStrap 3.3.0	timeout	23	92.31	50.00	timeout	41	95.50	75.04
Mootools 1.5.1	timeout	213	80.95	10.00	222.73	2	99.75	93.88
Prototype 1.7.2	timeout	49	94.87	15.38	34.91	2	100.00	87.72
Average	–	71	91.48	35.74	–	9	98.41	88.11

- MaxCALL(#) : 한 함수 호출 지점에서 불릴 수 있는 함수의 최대 개수, f()
- CALL(%): 전체 함수 호출 지점에서 단 하나의 함수 호출로 분석된 지점의 비율
- PROP(%): 객체의 property 접근 시 property 이름 값이 상수인 경우의 비율, obj[x]

결 론

PLRG
Programming Language Research Group

- 자바스크립트 웹 프로그램 정적 분석 시 높은 정확도는 분석 성능 향상으로 이어질 수 있음
  - 루프는 분석의 정확도를 떨어뜨리는 가장 큰 요인
- 루프 민감 분석
  - 분석 도중 루프 조건을 보고 판단하여 필요한 만큼만 루프의 iteration 분석 값을 구분
  - 전체 프로그램 분석 정확도 올림
  - 획기적인 분석 성능 향상