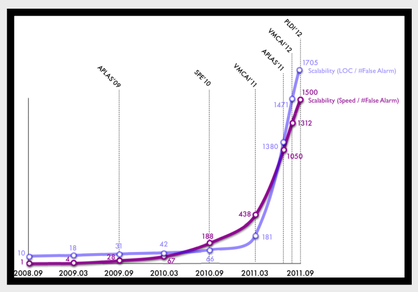
꼭 필요한 변수 관계에만 집중하는 프로그램 분석

허기홍, 이광근
서울대학교 프로그래밍 연구실

1. 동기

- Sparrow, 정확도 향상을 고민할 때
  - 100만 줄 C 코드 전체 분석 , 허나 많은 허위경보

예) make-3.76.1 2.7만 줄 1872 개 경보




```
472
473 = Buffer Overrun =
474
475 # of alarms grouped by tag : 1872
476 = Alarms =
477 1. ar.c:305:24 (ar_glob)
478     - state.1514.arname[i.1517]: offset: IBot, size: IBot
479 2. ar.c:306:29 (ar_glob)
480     - state.1514.arname[(i.1517 + 1)]: offset: IBot, size: IBot
481 3. ar.c:320:25 (ar_glob)
482     - *n.1516: offset: IBot, size: IBot
483     - names.1515[(i.1517 - 1)]: offset: [0,+oo), size: IBot
484 4. ar.c:319:44 (ar_glob)
485     - *n.1516: offset: IBot, size: IBot
```

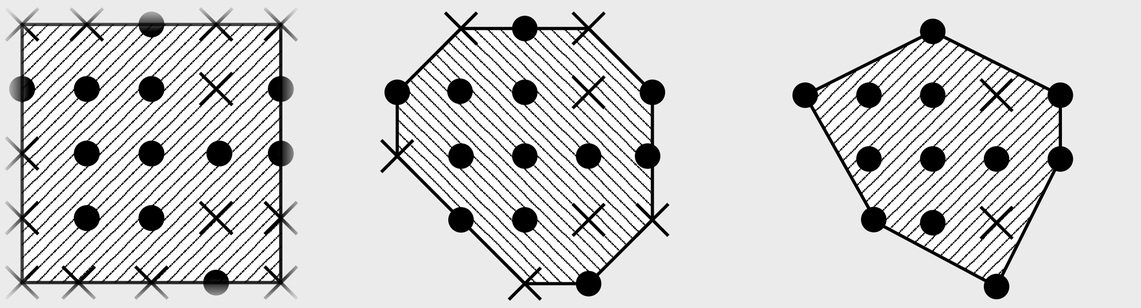
- 허위 경보 대표 원인 : **변수 관계**, 함수 문맥, 등

```
char *path = ((void *)0);
int path = xstrdup (getenv ("PATH"));  // path = {size:(-∞,+∞)}
int path_len = strlen (path);          // path_len = (-∞,+∞)
while (1) {
  for (; pos < path_len && path[pos] != ':'; pos++)
    str_add_char (file, path[pos]);    // pos < path.size?
}
```

- 기존 관계 분석 방식의 문제
  - 주먹구구식 변수 묶음
  - 때에 따라 불필요한 관계를 계산

2. 관계 분석 (Relational Analysis)

- 두 변수 값의 관계를 알아내는 분석
  요약값 공간 (abstract value domain)이 결정



$a \le x \le b$
$c \le y \le d$

$\pm x \pm y \le c$

$\Sigma_k a_k x_k \le b$

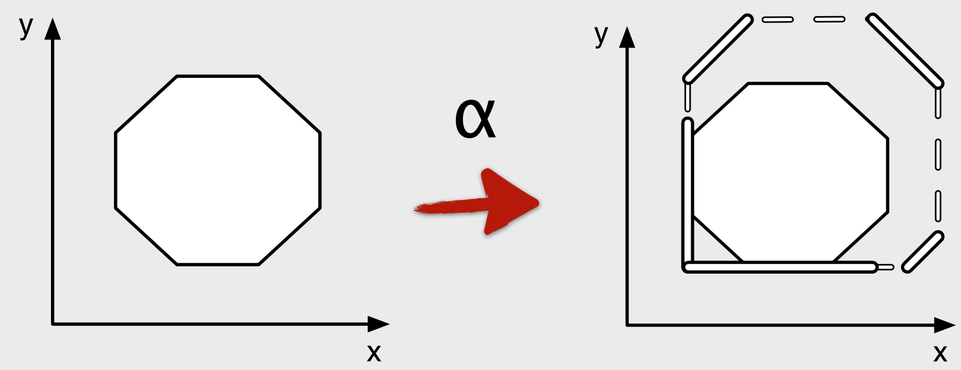
- 변수 묶음 (pack) : 관계를 알고 싶은 변수들

```
int http_parse_request(...){
   n = real_until(&data);
   data[n-1] = 0;
}
int read_until(char** data){
   while(...){
      len++;
   }
   buf = realloc(buf, len+1);
   *data = buf;
   return len;
}
```

{data, n, len, buf}

3. 집중 : 꼭 필요한 변수 관계

- 변수 묶음 계산, 체계적으로
  - 의미 기반 사전분석으로 어림 짐작
  예)

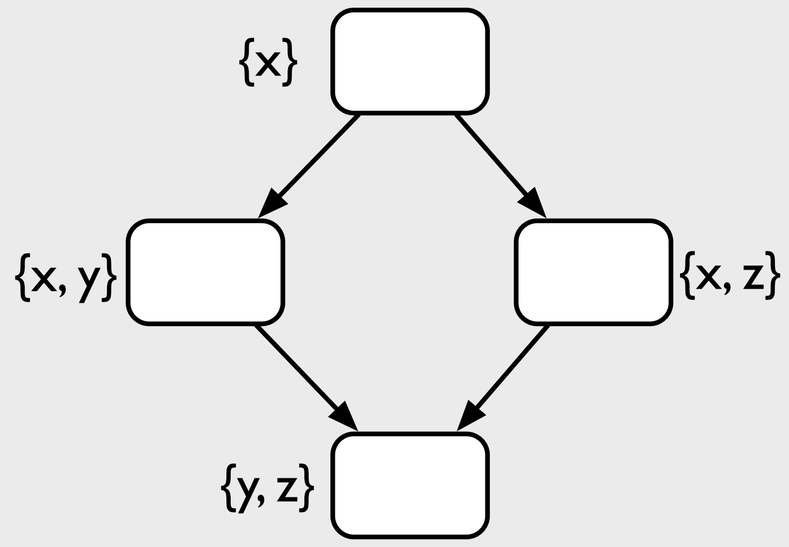


$x, y \in [1, 3]$
$x + y \in [3, 6]$
$x - y \in [-3, 3]$

$x, y \in \mathbb{Z}^+$
$x + y \in \mathbb{Z}^+$
$x - y \in \mathbb{Z}$

사전 분석에서 관계 있는 두 변수는 본 분석에서도 관계가 있다.

- 변수값 관계, 필요한 곳에서만 유동적으로



4. 향후 계획

- 사전 분석과 본 분석, 이론적으로 정리
- 실제 분석기에 적용하여 성능 확인
- 정확도 / 속도의 최적점 찾기
  - 더 / 덜 정확한 사전분석

5. 결론

- Sparrow 에 효율적인 관계 분석이 필요
- 꼭 필요한 관계에만 집중
  - 사전 분석을 통해 변수 묶음
  - 필요한 곳에서만 필요한 관계 분석

Programming Research Laboratory
SEOUL NATIONAL UNIVERSITY
ROSAEC center
Research On Software Analysis for Error-free Computing