이진 탐색을 이용하여
정교한 정적 분석에 효율 더하기

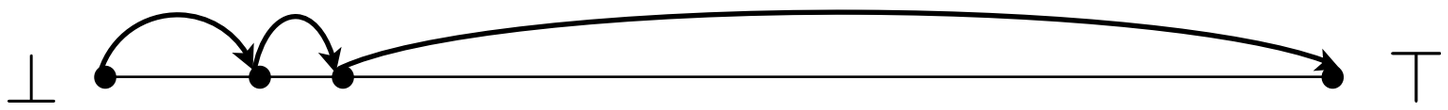
제 12회 소프트웨어 무결점 연구 센터 워크샵

서울대학교 프로그래밍 연구실
김솔

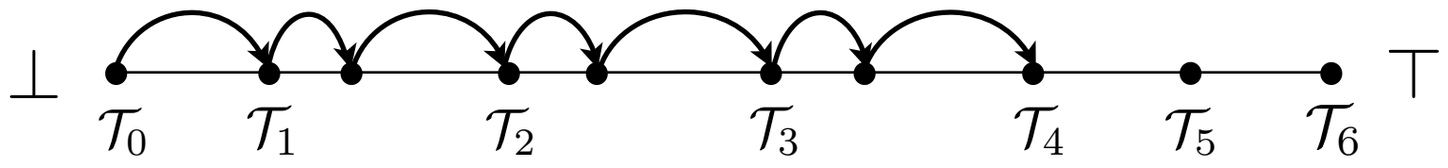
정교한 분석에 효율 더하기

```
1: int i = 0;
2: while (...)
3:   ...
4:   i++
```

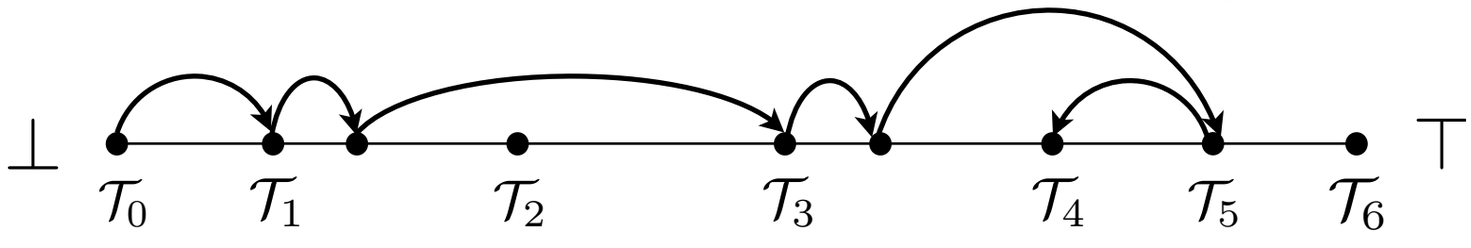
기본적인 방법 (기본적인 고정점 계산 가속 기법, conventional widening)



더 정교한 방법 (단계 고정점 계산 가속 기법, thresholded widening)



더 정교하며 효율적인 방법 (이진 탐색 단계 고정점 계산 가속 기법
, thresholded widening with binary search)



정교한 분석에 효율 더하기

```
1: int i = 0, buf[30];
2: while (...)
3:   if (i > 20)
4:     break
5:   i++
6: buf[i]
```

i의 단계 값이 {0, 5, 10, 12, 24, 150, 200, 300} 일때

기본적인 방법 (기본적인 고정점 계산 가속 기법, conventional widening)

`i : [0,0] -> [0,1] -> [0,+oo) -> [21,+oo)`

더 정교한 방법 (단계 고정점 계산 가속 기법, thresholded widening)

`i : [0,0] -> [0,1] -> [0,5] ->* [0,10] ->* [0,12] ->* [0,24] -> [21,24]`

더 정교하며 효율적인 방법 (이진 탐색 단계 고정점 계산 가속 기법
, thresholded widening with binary search)

`i : [0,0] -> [0,1] -> [0,12] ->* [0,150] ->* [0,24] -> [21,24]`

더 정교한 분석이 필요한 이유

```
1: int i = 0, buf[30];
2: while (...)
3:   if (i > 20)
4:     break
5:   i++
6: buf[i]
```

```
1: int i = 0, buf[6] = {1,2,3,1,2,3}
2: while (...)
3:    i = buf[i]
```

더 정교한 분석이 필요한 이유

기본적인 분석

흐름에 민감한 분석
(flow sensitive)

+

선택적으로 문맥에 민감한 분석
(selective context sensitive)

더 정교한 분석

흐름에 민감한 분석
(flow sensitive)

+

선택적으로 문맥에 민감한 분석
(selective context sensitive)

+

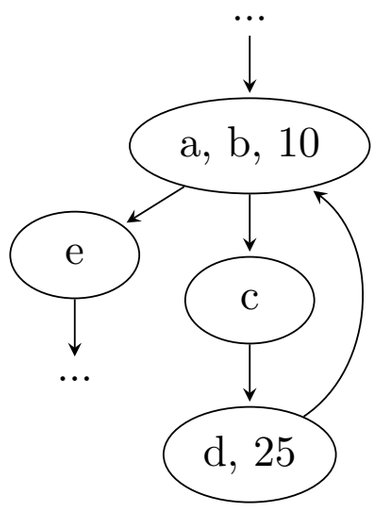
단계 고정점 계산 가속 기법
(thresholded widening)

5.71% 경보 ↓ **5.15% 정보 ↑**

단계 값은 어떻게 모으나

예비분석

단계값 모으기



+

변수	결과
a	$[-\infty, 100]$
b	$[-10, \infty]$
c	$[0, 55]$
d	$[-\infty, 7]$
e	$[16, 76]$
...	...

$\rightarrow \quad \mathcal{T}(\mathrm{a},\ \mathrm{b},\ \mathrm{c},\ \mathrm{d}) = \{0, 7, 10, 25, 55, 100\}$

본분석

실험 결과

프로그램	라인	단계 고정점 계산 가속 기법			이진 탐색 단계 고정점 계산 가속 기법		
		경보 ↓	정보 ↑	시간 (초) ↑	경보 ↓	정보 ↑	시간 (초) ↑
archimedes	7K	168	34	3.85	160	36	1.15
gnuchess	11K	62	60	39.27	43	11	17.80
bc	13K	8	3	133.25	4	7	10.29
tar	20K	13	0	159.23	10	3	27.10
make	27K	28	35	432.28	22	35	85.56
grep	28K	8	65	21.71	10	68	4.01
wget	35K	33	52	88.60	0	312	57.19
a2ps	64K	261	263	416.41	182	312	88.29
python	430K	N/A	N/A	∞	5	5	4,121.50

감사합니다.

참고

[1] Patrick Cousot and Radhia Cousot, “Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints.” in *Proceedings of ACM Symposium on principles of Programming Languages*, January 1977. pp. 238-252.

[2] Lies Lakhdar-Chaouch, Bertrand Jeannet, and Alain Girault, “Widening with Thresholds for Programs with Complex Control Graphs.” in *Springer, Heidelberg (LNCS, vol. 6996)*, 2011. pp. 492–502.

[3] Hakjoo Oh, Wonchan Lee, Kihong Heo, Hongseok Yang, and Kwangkeun Yi, “Selective context-sensitivity guided by impact pre-analysis.” in *Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, June 2014. p. 49.

[4] Sparrow http://ropas.snu.ac.kr/sparrow