분석기/검산기의 자동 생성기

조성근, 오학주, 허기홍, 강동옥
서울대학교 프로그래밍 연구실

ROSAEC 워크숍
2015/01/27

ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 NRF ERC

세 줄 요약

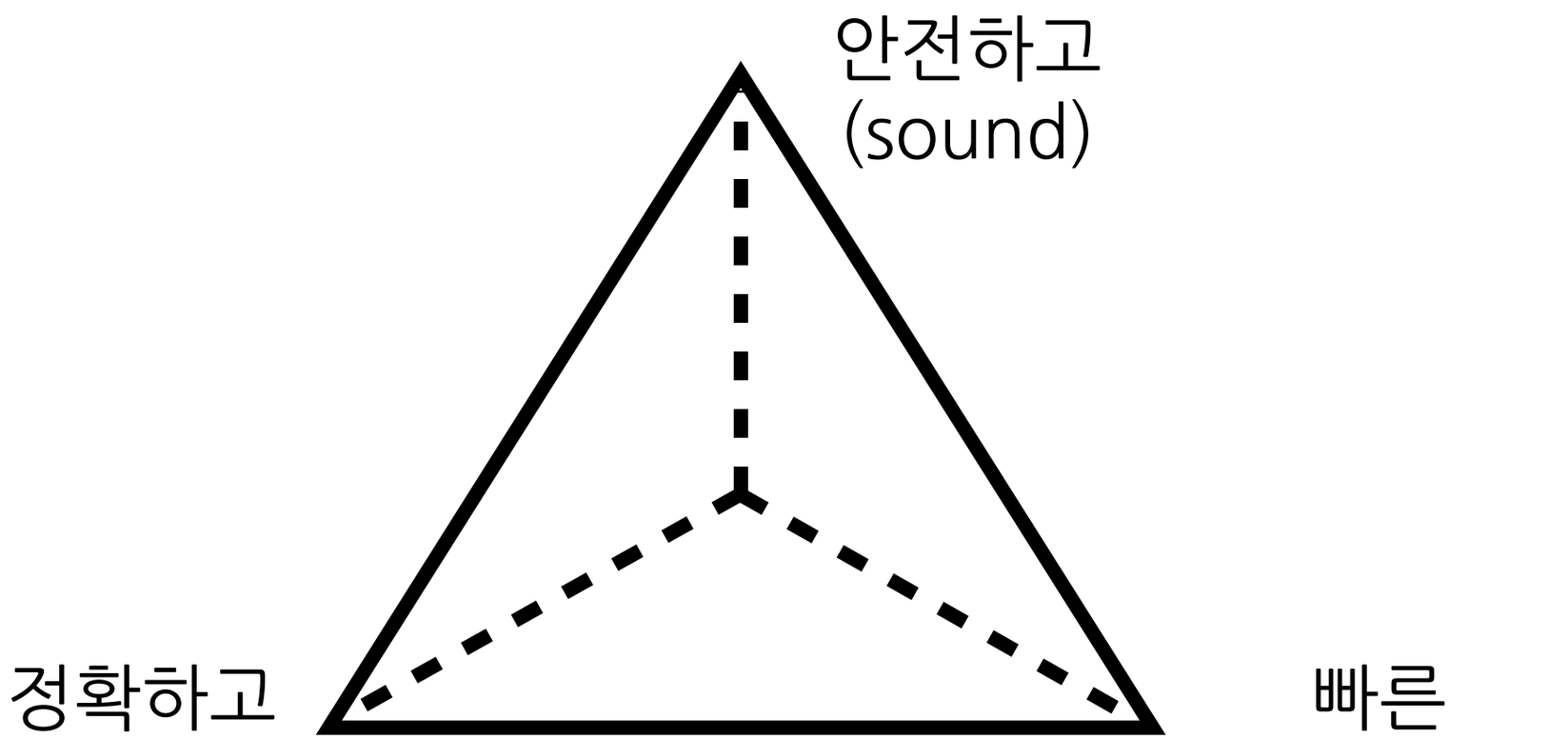
- 연구실에 분석/검증 관련 노하우가 쌓여있는데,
- 분석기/검산기를 자동으로 만들면 쉽게 공유할 수 있을 것 같은데,
- 그래서 만들다보니 괜찮더라.

연구 목표

- 누구나 쉽게 고성능 분석기, 검산기 제작

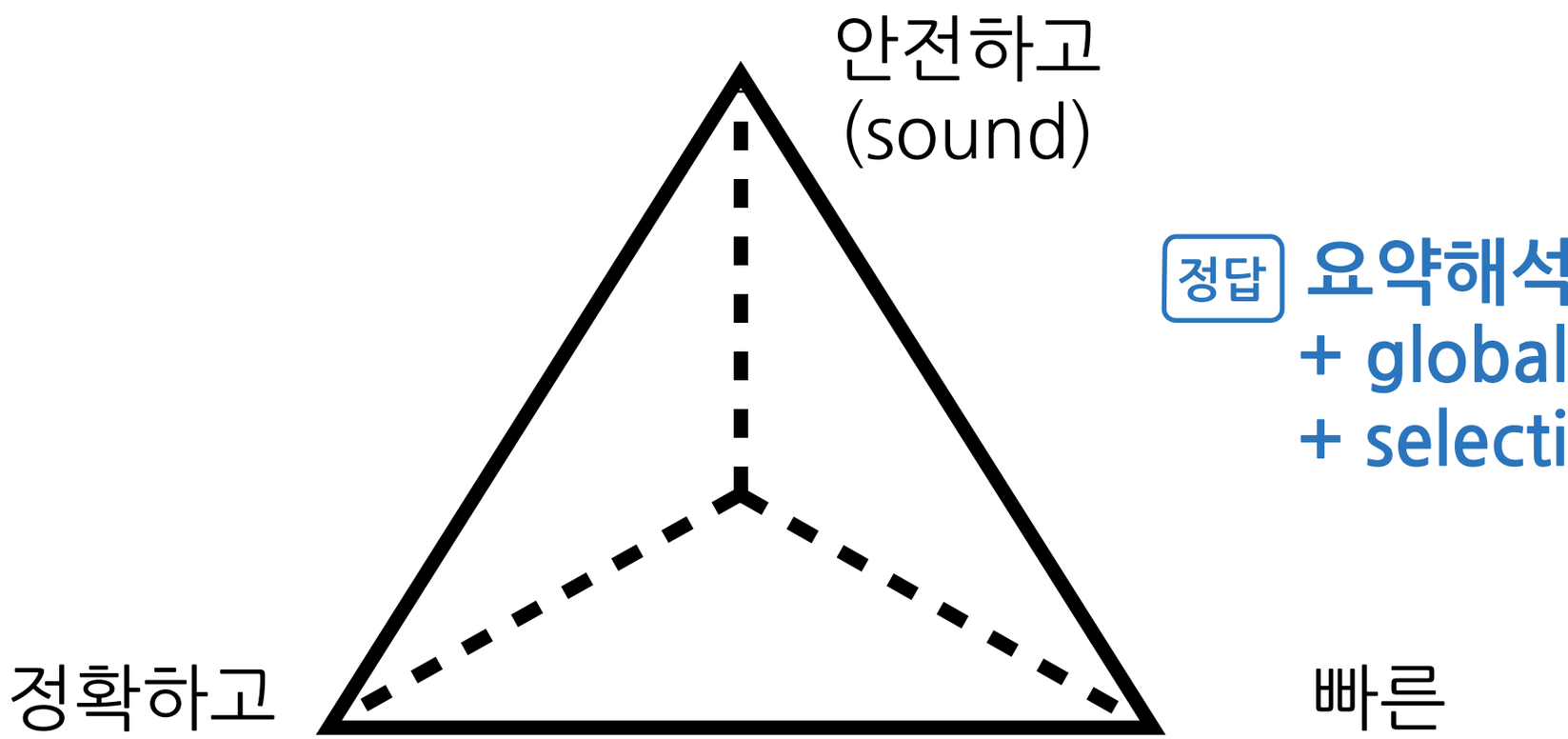
연구 목표

- 누구나 쉽게 고성능 분석기, 검산기 제작



연구 목표

- 누구나 쉽게 고성능 분석기, 검산기 제작



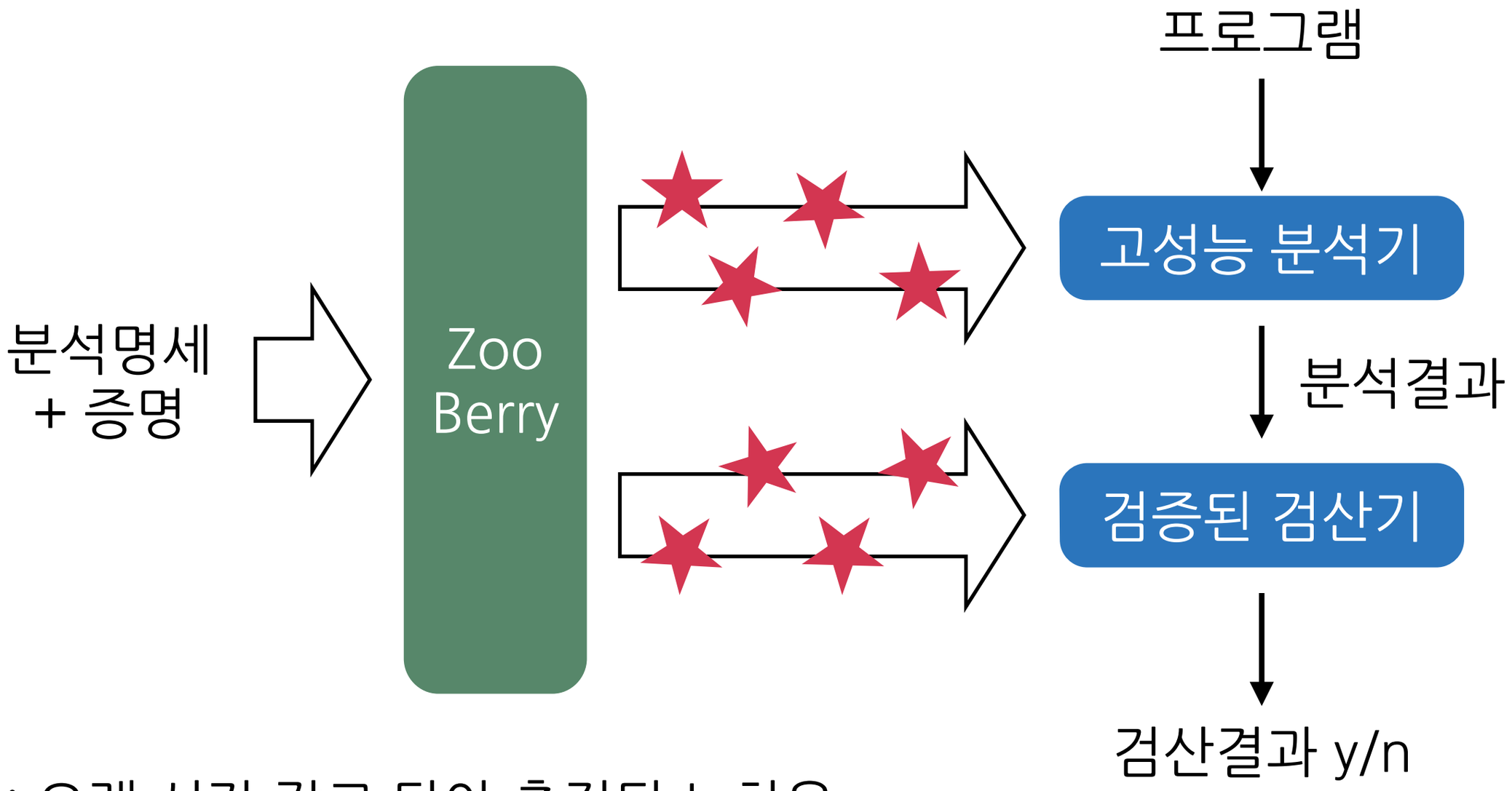
정답 **요약해석틀 / SW검증**
+ global / sparse
+ selective-X

누구나 쉬울까?

- sparse 분석
  - def/use 그래프는 어떻게 그리는지?
  - worklist order는 어떻게 정하는지?
- selective-X
  - ?
- 검산기
  - densification: 어떻게 빈 칸을 채우는지?
  - 어떻게 검산을 빠르게 하는지?

분석기/검산기를 자동으로 생성하자

큰그림



★ : 오랜 시간 갈고 닦아 축적된 노하우

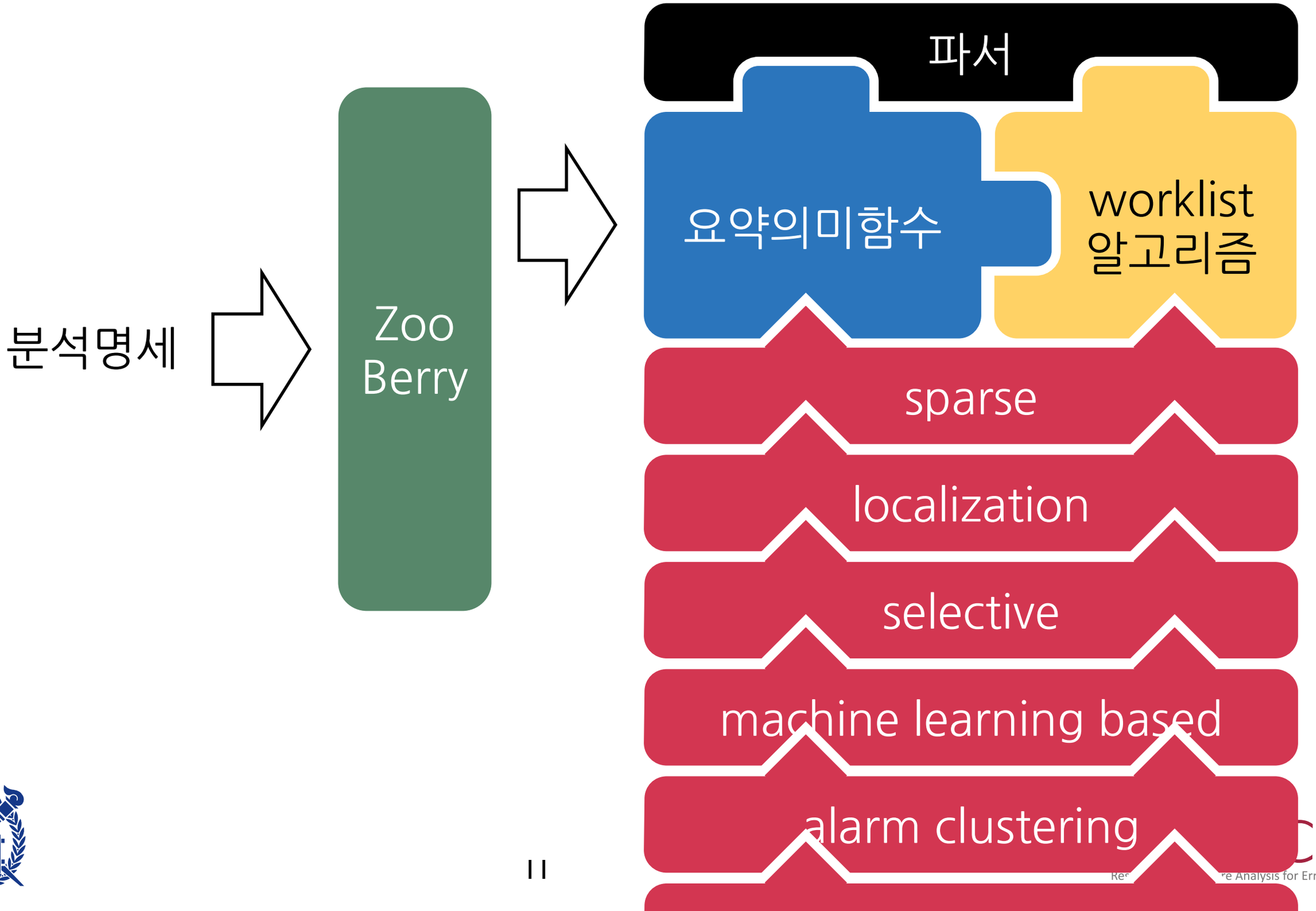
수동 vs 자동 성능비교

Pgm	LOC	수동 제작				자동 생성				**변화Δ**	
		분석기		검산기		분석기		검산기		**분석기+검산기**	
		시간	메모리	시간	메모리	시간	메모리	시간	메모리	**시간**	**메모리**
time	2K	0	3	0	4	0	4	0	3	NaN	**x1.0**
spell	2K	0	5	0	6	0	5	0	5	NaN	**x0.9**
bc	14K	4	45	10	67	3	50	14	63	**x1.2**	**x1.0**
tar	28K	6	53	21	103	6	86	28	102	**x1.3**	**x1.2**
less	24K	21	144	71	218	23	220	79	323	**x1.1**	**x1.5**
wget	35K	20	118	162	254	31	278	214	306	**x1.3**	**x1.6**
bison	56K	14	120	73	222	19	162	105	208	**x1.4**	**x1.1**
screen	45K	413	780	657	1362	772	2376	705	2224	**x1.4**	**x2.1**
합계		478	1268	994	2236	854	3181	1145	3234	**x1.4**	**x1.8**

* 단위: 시간(초), 메모리(MB)

조금 더 구체적으로

고성능 분석기 생성


분석명세
Zoo
Berry
파서
요약의미함수
worklist
알고리즘
sparse
localization
selective
machine learning based
alarm clustering

분석명세 예

```
Definition eval (e : exp) (m : Mem.t) : Val.t :=
  match e with
  | Const c => eval_const c
  | Lval l =>
    let lv := eval_lv l m in
    mem_lookup lv m
  | UnOp u e =>
    let v := eval e m in
    eval_uop u v
  | BinOp b e1 e2 =>
    let v1 := eval e1 m in
    let v2 := eval e2 m in
    eval_bop b v1 v2
  | Question e1 e2 e3 =>
    let v1 := eval e1 m in
    let i1 := itv_of_val v1 in
    if i1 === bot then bot else
    if [0, 0] === i1 then eval e3 m else
    if ~~ ([0, 0] <== i1) then eval e2 m else
      let v2 := eval e2 m in
      let v3 := eval e3 m in
      v2 \/ v3
```

bot

interval

join

검증된 검산기 생성

분석명세
+ 증명

Zoo
Berry

densifier

요약의미함수

worklist
알고리즘

검산기의 안전성 증명

배경기술: 축적된 분석/검증 노하우

- 분석기 자동생성 기술
- 분석기 최적화 기술
- 최적화된 분석기를 위한 검증 기술

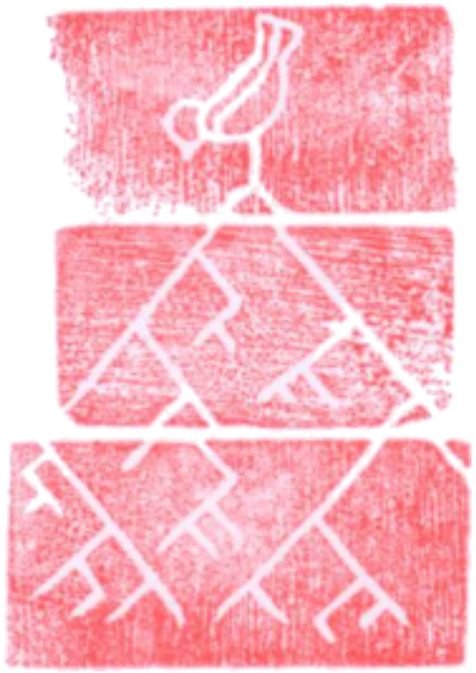
분석기 자동생성 기술

- Zoo project
  - 다양한 분석명세로부터 분석기를 자동생성
  - abstract interpretation, set-based, data flow analysis
  - Rabbit: 분석명세를 위한 고수준 언어

분석기 최적화 기술

- selective-X (PLDI'14)
  - 분석에 도움이 되는 부분에 분석비용 집중
- sparse (PLDI'12)
  - 필요한 곳에 필요한 정보를
- localization (VMCAI'11, APLAS'11)
- 그 외

최적화된 분석기를 위한 검증 기술



- SparrowBerry (2012-current)
  - Sparse Sparrow에 특화된 검증된 검산기
  - 분석의 약 2~3배 시간으로 검산 성공

앞으로 할 일

- 미완성 부분: 앞단(Rabbit), 증명부분
- 분석명세 진단
  - 요약의미함수가 단조임을 자동으로 검사 (VMCAI'02)
  - 그밖에 성질들 검사: join ⊑ widen
- 대상언어 확장

결론

- 안전하고, 정확하고, 빠른 분석기를 누구나 쉽게 만들 수 있다.

참고

[ROSAEC tech memo'15] Towards Scalable Translation Validation of Static Analyzers.
Jeehoon Kang, Sungkeun Cho, Joonwon Choi, Chung-Kil Hur, Kwangkeun Yi.

[PLDI'14] Selective Context-Sensitivity Guided by Impact Pre-Analysis.
Hakjoo Oh, Wonchan Lee, Kihong Heo, Hongseok Yang, and Kwangkeun Yi.

[PLDI'12] Design and Implementation of Sparse Global Analyses for C-like Languages.
Hakjoo Oh, Kihong Heo, Wonchan Lee, Woosuk Lee, and Kwangkeun Yi.

[VMCAI'11] Access Analysis-Based Tight Localization of Abstract Memories.
Hakjoo Oh, Lucas Brutschy, and Kwangkeun Yi.

[APLAS'11] Access-Based Localization with Bypassing.
Hakjoo Oh and Kwangkeun Yi.

[VMCAI'02] Static Monotonicity Analysis for Lambda-definable Functions over Lattices.
Andrzej Murawski and Kwangkeun Yi.

Zoo Project: http://ropas.snu.ac.kr/zoo/

Sparse Analysis: http://ropas.snu.ac.kr/sparseanalysis/

SparrowBerry: http://ropas.snu.ac.kr/sparrowberry/

감사합니다.

ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 NRF ERC