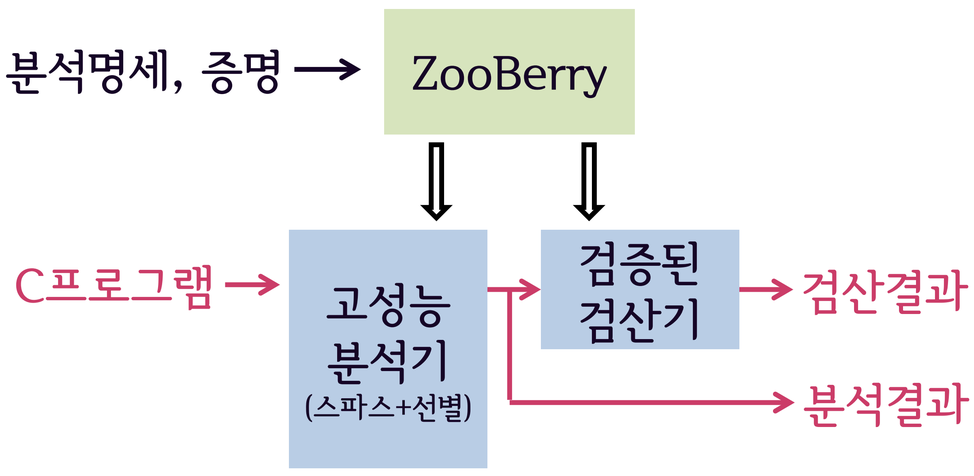
ZooBerry : 분석기/검산기 자동 생성기

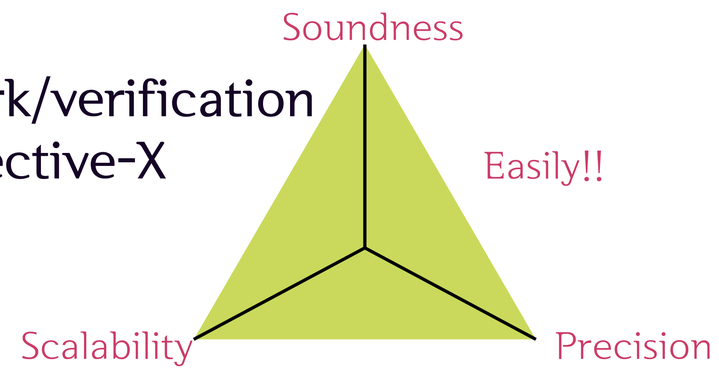
조성근, 오학주, 허기홍, 강동옥, 이광근
서울대학교 프로그래밍연구실(ROPAS)

큰그림



동기

- 누구나 쉽게 고성능 분석기, 검산기 제작.
- 고성능이라 함은,
  안전하고(sound) -> AI framework/verification
  정확하고(precise) -> global/selective-X
  빠르고(scalable) -> sparse



수동vs자동 성능비교

- 직접 제작한 고성능 분석기와 비교하여 자동 생성된 분석기도 큰 차이 없는 성능을 보임.

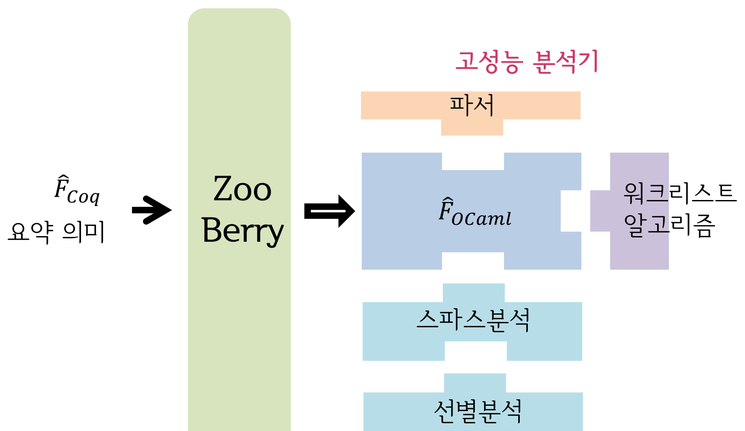
Pgm	LOC	수동 제작				자동 생성				변화Δ	
		분석기		검산기		분석기		검산기		분석기+검산기	
		시간 (s)	메모리 (MB)	시간 (s)	메모리 (MB)	시간 (s)	메모리 (MB)	시간 (s)	메모리 (MB)	시간 (s)	메모리 (MB)
time	2k	0	3	0	4	0	4	0	3	NaN	x1.0
spell	2k	0	5	0	6	0	5	0	5	NaN	x0.9
bc	14k	4	45	10	67	3	50	14	63	x1.2	x1.0
tar	28k	6	53	21	103	6	86	28	102	x1.3	x1.2
less	24k	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

- 현실:

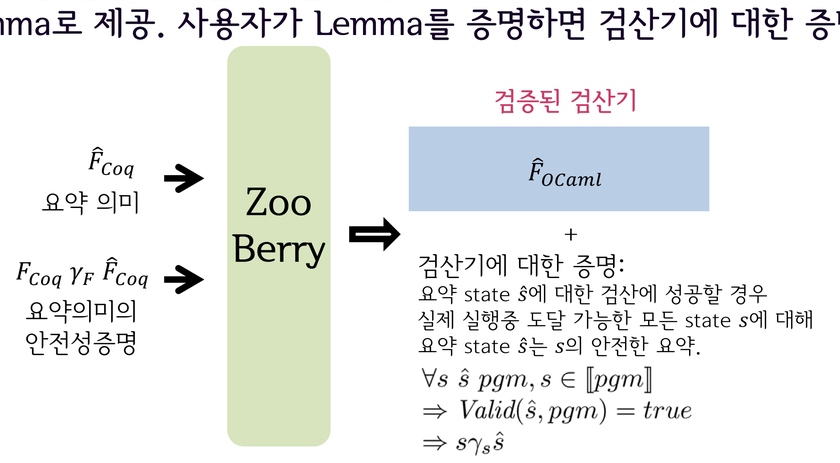
언제 다 이해하고 구현하나?
논문의 알고리즘이해 vs 실제구현?
개발 시간은?

해결책

- 고성능 분석기 생성:
  - Coq에서 사용할 수 있는 요약도메인 모듈들을 제공하여 사용자는 모듈들을 조합해 손쉽게 요약도메인 정의. 요약도메인 모듈에는 요약의미 정의에 필요한 ⊥, ⊑, =, ⊔, ⊓, widen, narrow 가 정의되어 있음.
  - 사용자의 분석기 요약 의미는 Coq으로 작성. OCaml로 Extract 수행함.
  - 파서와 워크리스트 알고리즘과 같은 분석기 기본 모듈부터 스파스, 선별분석과 같은 고난도 기술까지 철저한 abstraction layer로 모듈화. Extract한 요약 의미 모듈이 바로 끼워 들어가 고성능 분석기가 되도록.



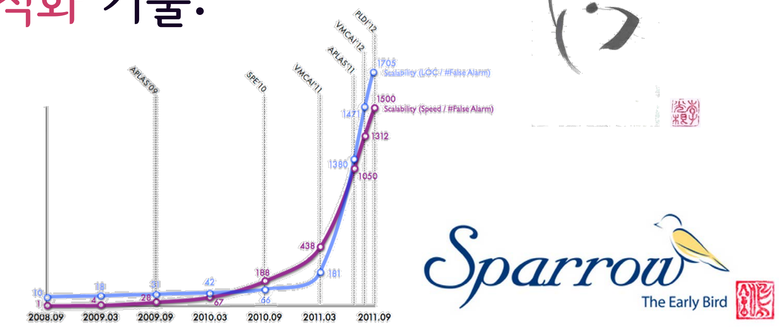
- 검증된 검산기 생성:
  - 분석기를 생성할 때 사용자가 작성한 요약의미를 그대로 사용하여 검산기 생성. OCaml로 Extract 수행함.
  - 생성한 검산기의 올바름을 증명하기 위해 사용자가 증명해야 하는 부분을 Lemma로 제공. 사용자가 Lemma를 증명하면 검산기에 대한 증명이 생성됨.



배경기술: 축적된 분석/검증 노하우

- 분석기 자동생성 기술:
  - Zoo project (VMCAI'02): 다양한 분석명세로부터 분석기를 자동 생성(abstract interpretation, set-based, data flow, model checking).
  - 분석명세를 고수준의 언어(Rabbit)로 기술.

- 분석기 최적화 기술:



Sparrow
The Early Bird

- Selective-X (PLDI'14): 분석에 도움이 되는 부분에 계산비용 집중.
- Sparse (PLDI'12): 필요한 곳에 필요한 정보를.
- Localization (VMCAI'11, APLAS'11): 함수분석에 꼭 필요한 정보만.
- 안전성을 체계적으로 조절하여 정확도를 높이는 방법 기술(2014-current).

- 최적화된 분석기를 위한 검증 기술:
  - SparrowBerry (2012-current): Sparse Sparrow에 특화된 검증된 검산기. 분석의 약 2~3배 시간으로 검산 성공.

앞으로 할일

- 미완성 부분: 앞단 Rabbit 구현 + 증명 부분.
- 분석명세 검사:
  - 요약의미함수가 단조임을 자동으로 검사 (VMCAI'02).
  - 사용자가 정의한 함수가 만족해야 하는 성질을 검사: join ⊑ widen, meet ⊑ id.
- 대상 언어 확장: 객체지향 언어, 함수형 언어.

[1] "Static Monotonicity Analysis for Lambda-definable Functions over Lattices", Andrzej Murawski and Kwangkeun Yi, VMCAI'02
[2] "Selective Context-Sensitivity Guided by Impact Pre-Analysis", Hakjoo Oh and Wonchan Lee and Kihong Heo and Hongseok Yang and Kwangkeun Yi, PLDI 2014
[3] "Design and Implementation of Sparse Global Analyses for C-like Languages", Hakjoo Oh and Kihong Heo and Wonchan Lee and Woosuk Lee and Kwangkeun Yi, PLDI 2012
[4] "Access Analysis-Based Tight Localization of Abstract Memories", Hakjoo Oh and Lucas Brutschy and Kwangkeun Yi, VMCAI 2011
[5] "Access-Based Localization with Bypassing", Hakjoo Oh and Kwangkeun Yi, APLAS 2011

Programming Research Laboratory
ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 NRF ERC
SEOUL NATIONAL UNIVERSITY