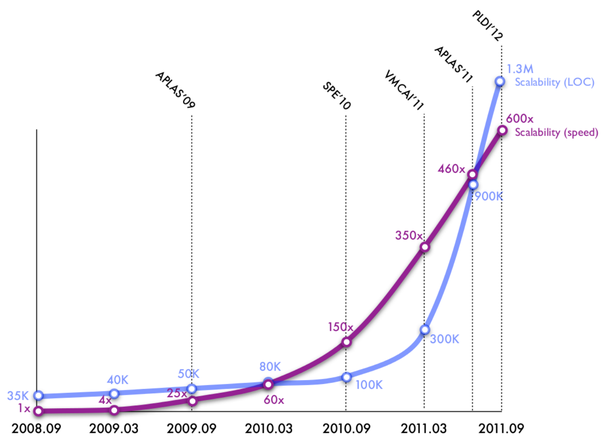
SparrowBerry: Sparrow 분석결과를 검산하고 검산 성질 검증하기

강지훈 조성근 최준원 허충길 이광근

빠르고 복잡한 우리 분석기 Sparrow, 안전할까 걱정했어요.

Sparrow The Early Bird

100만줄의 C 코드 분석하는 빠른 속도



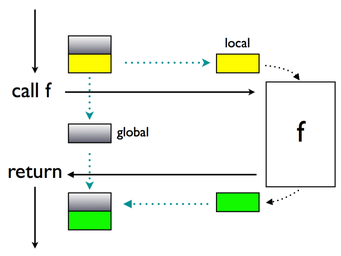
많은 최적화를 담은 복잡한 구현

3만줄 OCaml 코드

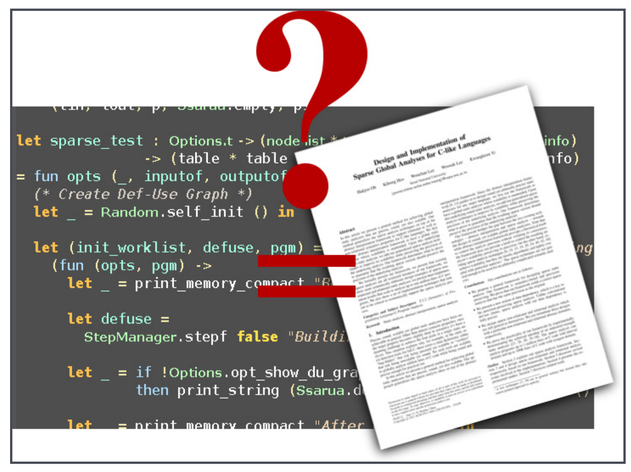
최적화 (sparse analysis, localisation, …)

baseline analysis $\hat{F} : \hat{D} \to \hat{D}$ ⟹ (sparsify) "sparse" version $\hat{F}_s : \hat{D} \to \hat{D}$

$fix\,\hat{F}$ = (still) $fix\,\hat{F}_s$



이 분석기가 디자인대로 안전할까?



이젠 걱정 말아요, 검산기 SparrowBerry를 만들었거든요. 검증도 했어요!

검산 과정

분석기 → (프로그램, 분석결과) → 검산기 → 검산 성공/실패 여부

분석기 ← 프로그램

Coq으로 증명

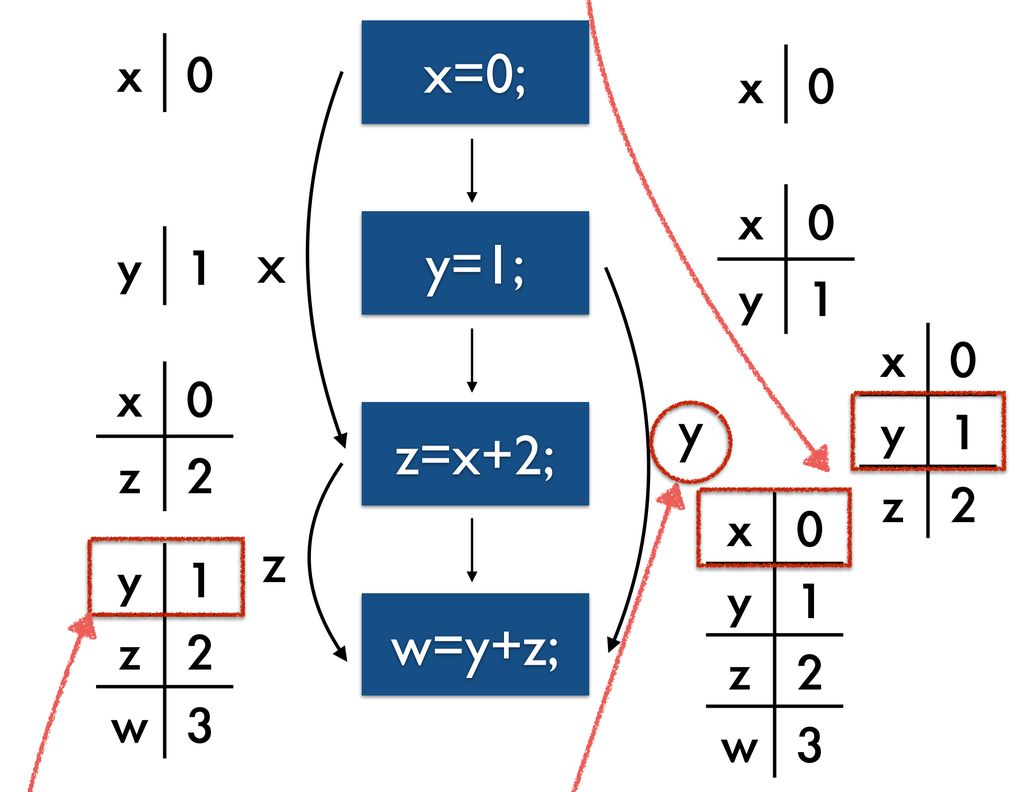
검증한 검산 성질

프로그램 p와 분석결과 A에 대해 검산 성공하면,
분석결과 A는 p의 실제실행의미를 포섭한다.

증명: 요약해석 틀의 증명 그대로.

검산기가 너무 느려서 애먹었었어요. 물론 지금은 해결 했지요!

Sparse 분석결과를 뻥튀기하느라 느렸어요



Sparse 분석결과를 뻥튀기 안하고 검산하긴 힘들어요

예를 들면 이 y는 어디서 온 y인지 알아야해요.
y=1에서 왔죠. 일반적으론,

Sparse analysis과 SSA 변환이
올바름을 증명해야 해요 :-(

그래서 절충안으로 일부만 뻥튀기해서 검산했어요

자세한 방법은? 따로 얘기해봐요!

5달만에 검산기 만들고 검증했어요.
오픈 소스 10만줄 C 코드도 문제없이 검산해요.

검산기를 이용해 Sparrow 버그를 13개나 찾았어요!

Category	Description
dependence graph	Dynamic locations were not included in a definition set when arrays are declared. Graph edges were not drawn correctly when weak-update occurs. Graph edges were not drawn correctly when an encoded library function is called. Graph edges for fields were not drawn correctly. Return edges should be definition points.
semantics	Field values should be top if the struct itself is top. Local variables should not be removed on an exit node in some cases. Field values should not be declared as dynamic values. Typing errors on abstract interval operations. 0 and null worked inconsistently in some cases. Values from address-taken locations should not be removed on exit nodes. Weak update conditions for local variables were incorrect.
parser	Functions and local variables should be treated individually, even if their names are same.

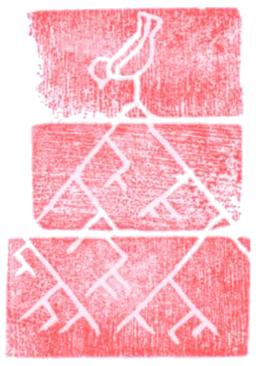
검산 실패를 이용해 분석기 버그를 찾았어요

검산 실패한 프로그램

작게 만들기 (파일을 하나씩 빼봐요) → 검산 실패하는 작은 프로그램 → 인간 (실패 원인을 분류해요)

검산기 정의를 잘못했으면 고쳐요.
분석기 오류이면 버그를 찾은거에요!

당신의 분석기도 검산기 하나 장만하세요.



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

Programming Research Laboratory
http://ropas.snu.ac.kr

SEOUL NATIONAL UNIVERSITY