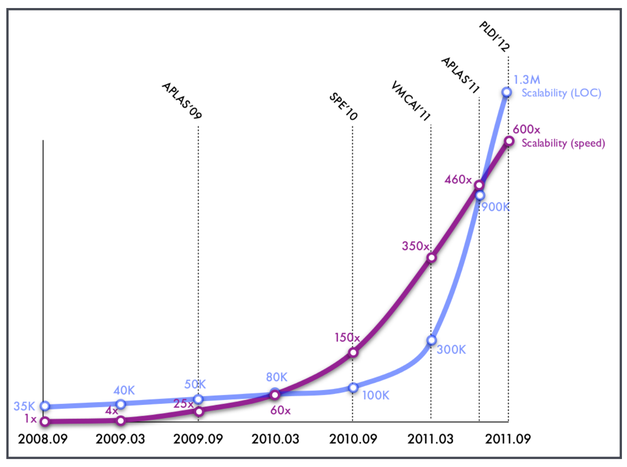
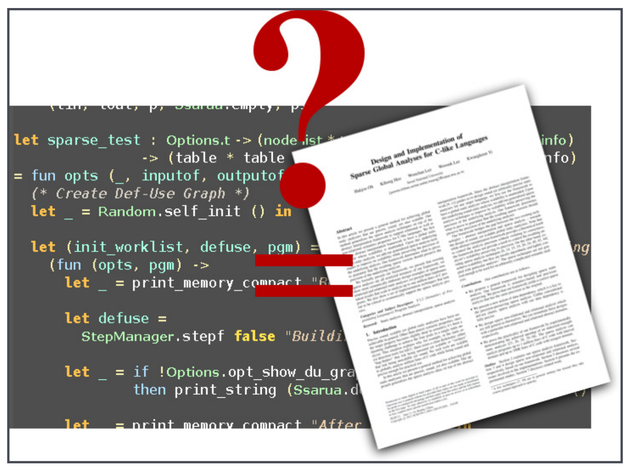
Sparse Sparrow, Verified.

Sungkeun Cho, Jeehoon Kang, Joonwon Choi, Youngseok Lee, and Kwangkeun Yi

Sparse Sparrow, 바르게 구현되었는가?





```
gamma_state sn sHat.
Proof.
  intros p sHat Hwf H sk sn H'.
  induction 1; auto.
  apply IHcon_large_f_star.
  apply gamma_state_monotone with (abs_large_f p sHat); auto.
  apply soundness_large_f with s1; auto.
Qed.

Theorem soundness : forall p sHat,
  state_wf sHat ->
  AbsState.order (abs_large_f p sHat) sHat ->
  forall n sn,
  con_large_f_star p (n,Memory.empty) sn ->
  gamma_state sn sHat.
Proof.
  intros.
  apply soundness' with p (n, Memory.empty); auto.
  apply initial.
Qed.
```

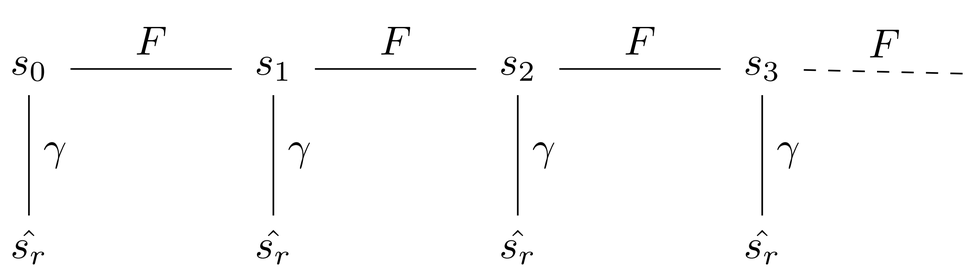
수백만줄의 코드를 분석할 수 있는 Sparrow

그 구현은 올바른가?

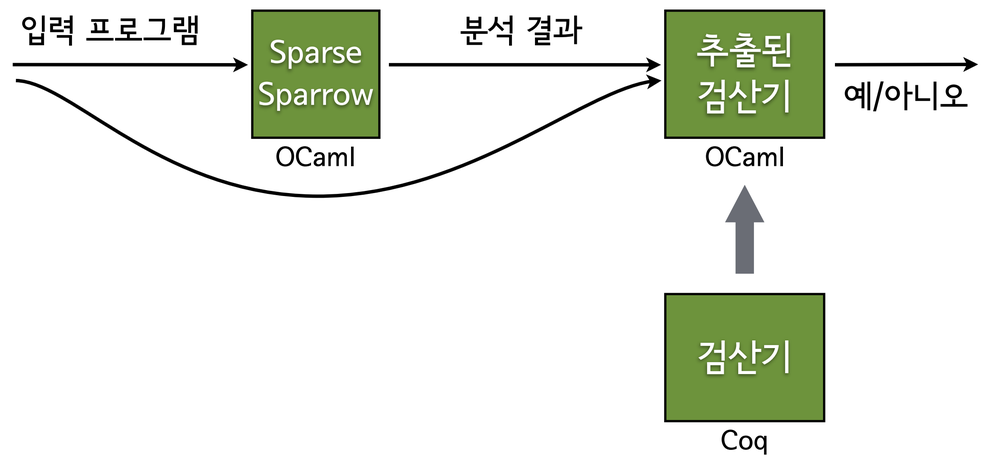
구현의 안전성을 coq으로 증명했다!

우리가 원하는 증명

$$\forall n, s_n : s_0 F^n s_n \rightarrow s_n \gamma \hat{s_r}$$

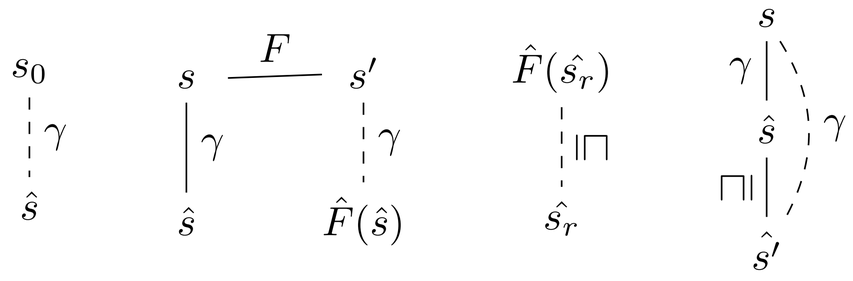


분석기 뒤에 검산기

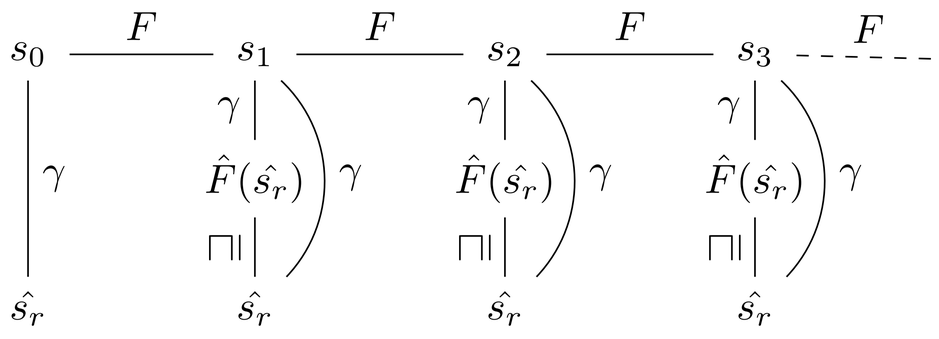


분석결과가 모든 실제 실행을 포섭하는가?

필요한 보조정리들



1) 분석 결과가 초기 상태를 포섭
2) 프로그램을 한 단계 실행해도 여전히 포섭
3) 분석 결과는 검산기 실행 의미의 한 고정점
4) 포섭 관계는 monotone 관계



보조정리들 중 1), 2), 4)는 coq에서 증명.
3)은 검산기가 입력 프로그램과
분석 결과를 받아 직접 확인

현재의 상태

```
--------------------------
Validating begins...
--------------------------

Validation passed.

Validating completes 0.sec
```

잘 알려진 벤치마크를 대상으로
분석기+검산기의 안전성과 Scalability 실험 중

1~4를 이용하여 위와 같이 증명할 수 있다!

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