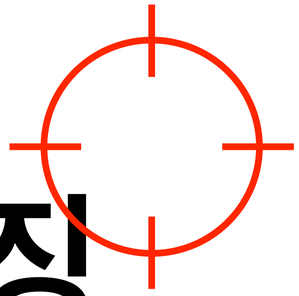
정밀 조준 퍼징



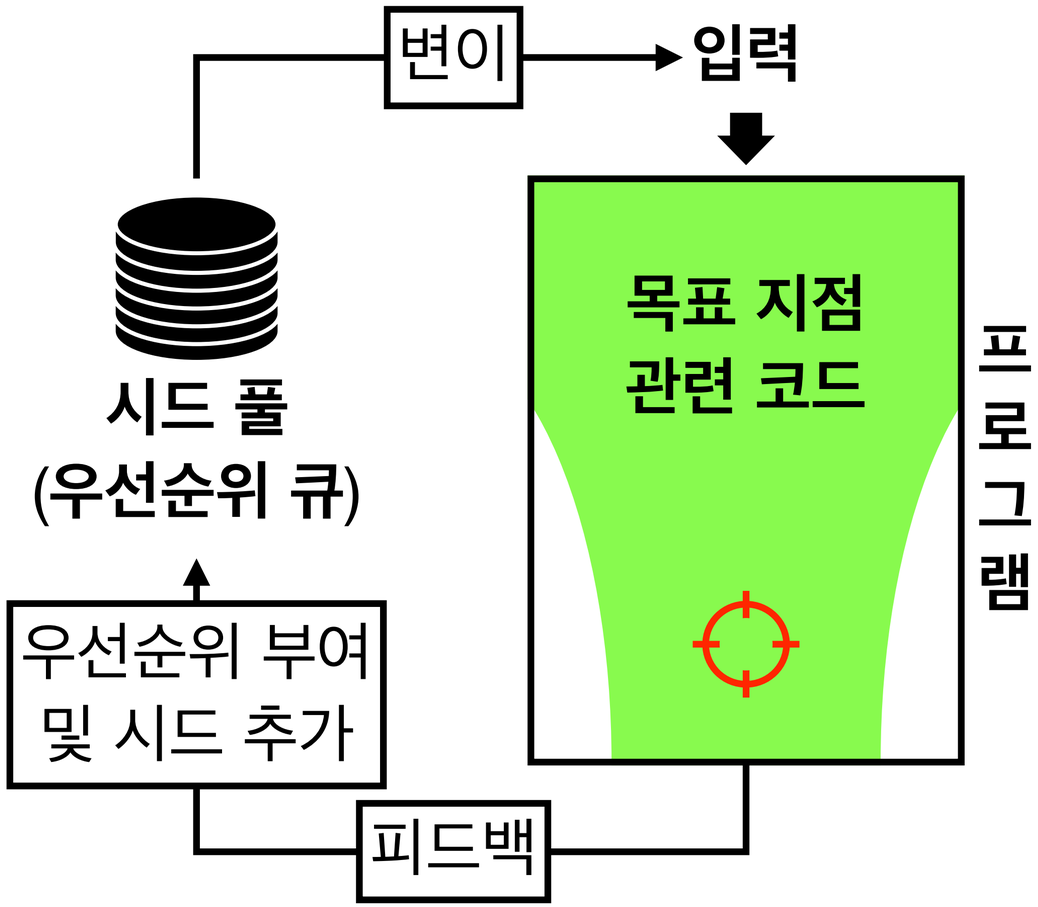
고려대학교 소프트웨어분석연구실 석사과정 허준용

SIGPL 2026 여름학교 번개 발표

지향성 퍼징

- **목표 지점**에 도달하는 입력을 효율적으로 생성하는 퍼징
  ㄴ <파일명>:<라인번호> (예: `decompile.c:2015`)
- 패치 검증, 취약점 재현, 정적 분석 경고 검증 등
- 목적: 목표 지점에서 **오류 빨리 터뜨리기**

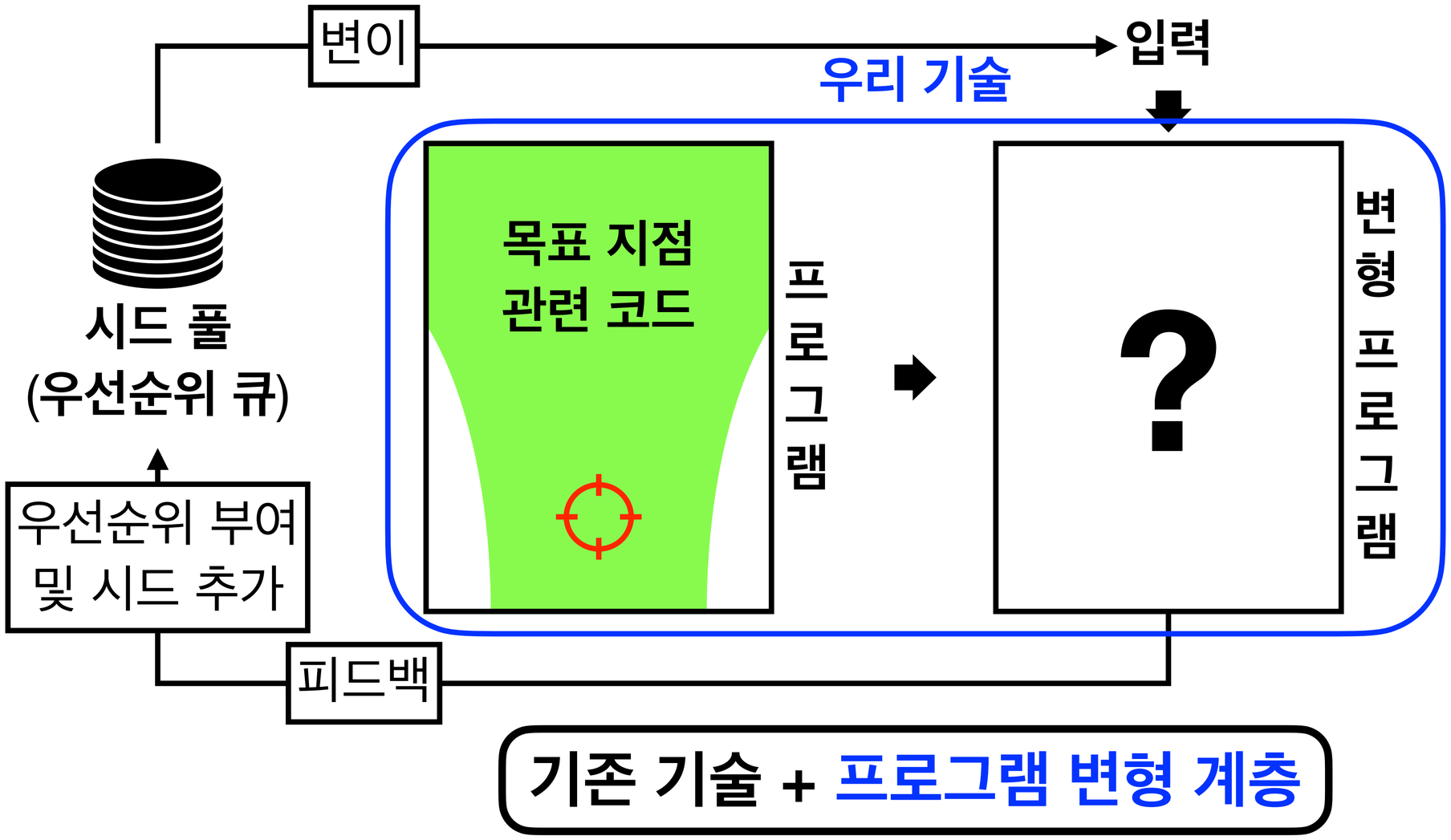
지향성 퍼징



목표 지점 관련 코드의 피드백에 따라 시드의 우선순위가 결정됨

- 목표 지점 관련 코드 예:
  - 목표 지점에 도달 가능한 코드 (AFLGo)
  - 데이터 의존 코드 (DAFL)
- 피드백 예:
  - 커버리지
  - 실행된 관련 코드와 목표 지점 간 거리

우리 기술

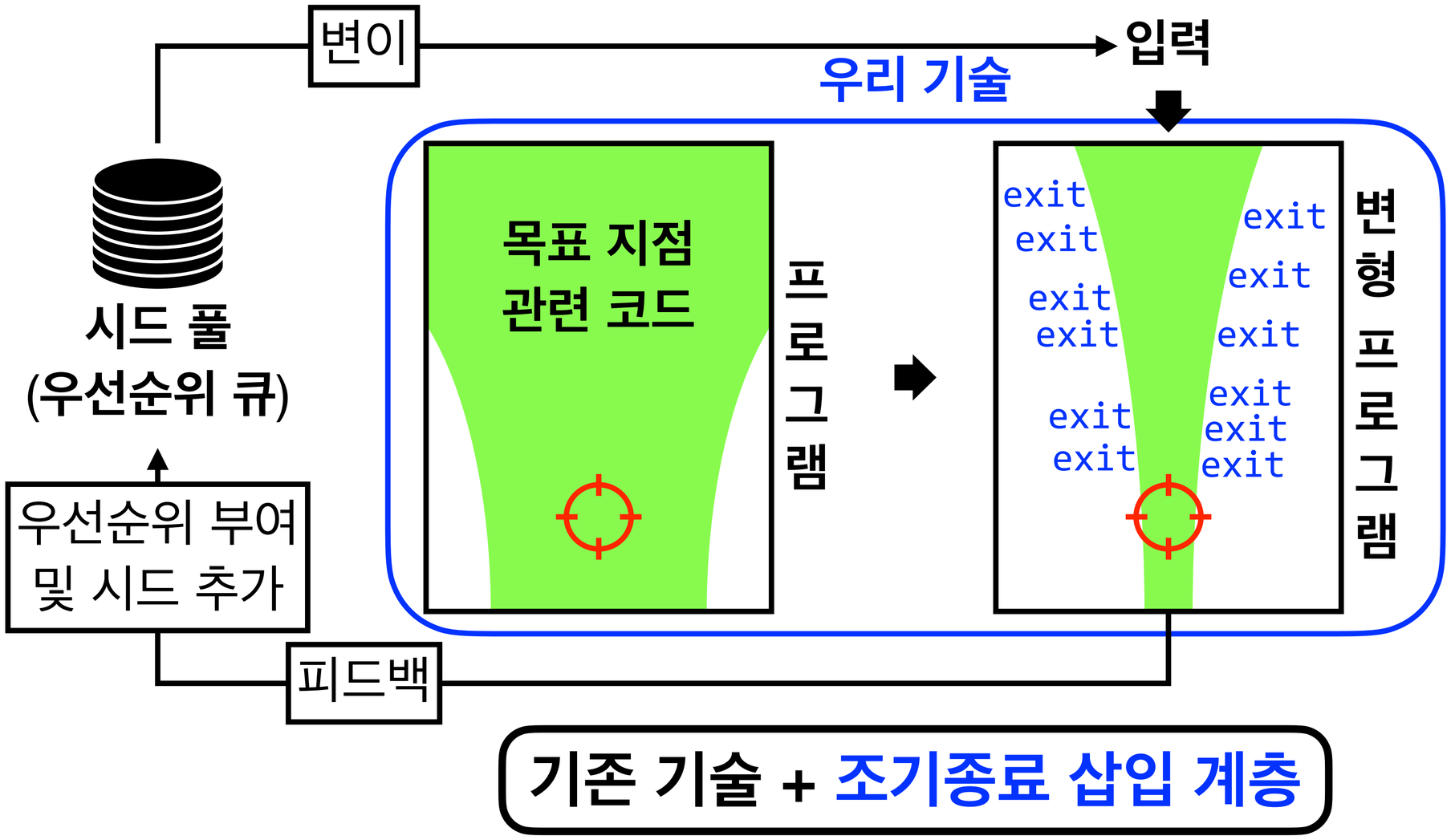

변이
입력
우리 기술
시드 풀
(우선순위 큐)
목표 지점
관련 코드
프로그램
?
변형 프로그램
우선순위 부여
및 시드 추가
피드백
기존 기술 + 프로그램 변형 계층

우리 기술 평가

- 평가 방식: 오류 노출 시간을 여러 번 측정 → 중앙값 (단위: 초)

	DAFL	DAFL + 우리 기술
CVE-2017-9988 (1시간, 40번 반복)	중앙값 X (절반 이하 성공)	2,392
CVE-2020-6628 (12시간, 40번 반복)	34,491	21,002 **1.64x**

우리 기술

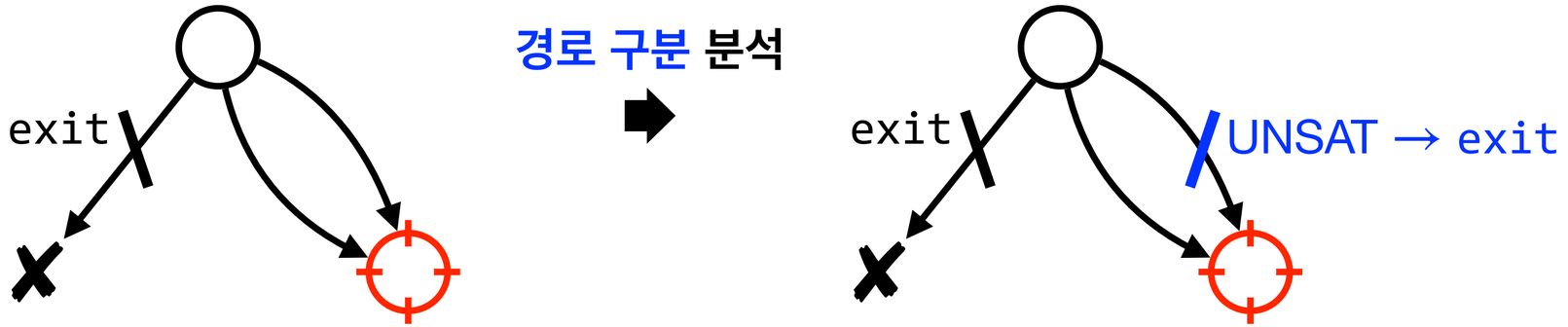

변이
입력
우리 기술
시드 풀
(우선순위 큐)
우선순위 부여
및 시드 추가
피드백
목표 지점
관련 코드
프로그램
exit
변형 프로그램
기존 기술 + 조기종료 삽입 계층

기존 조기종료 삽입 기술

- SDFuzz (USENIX Security’24)
  - **실행 흐름 그래프**상의 목표 지점 도달 불가능 분기
- BEACON (S&P’22)
  - 정수 구간 분석으로 목표 지점 도달의 **전제조건 추론** → 위반 시 조기종료

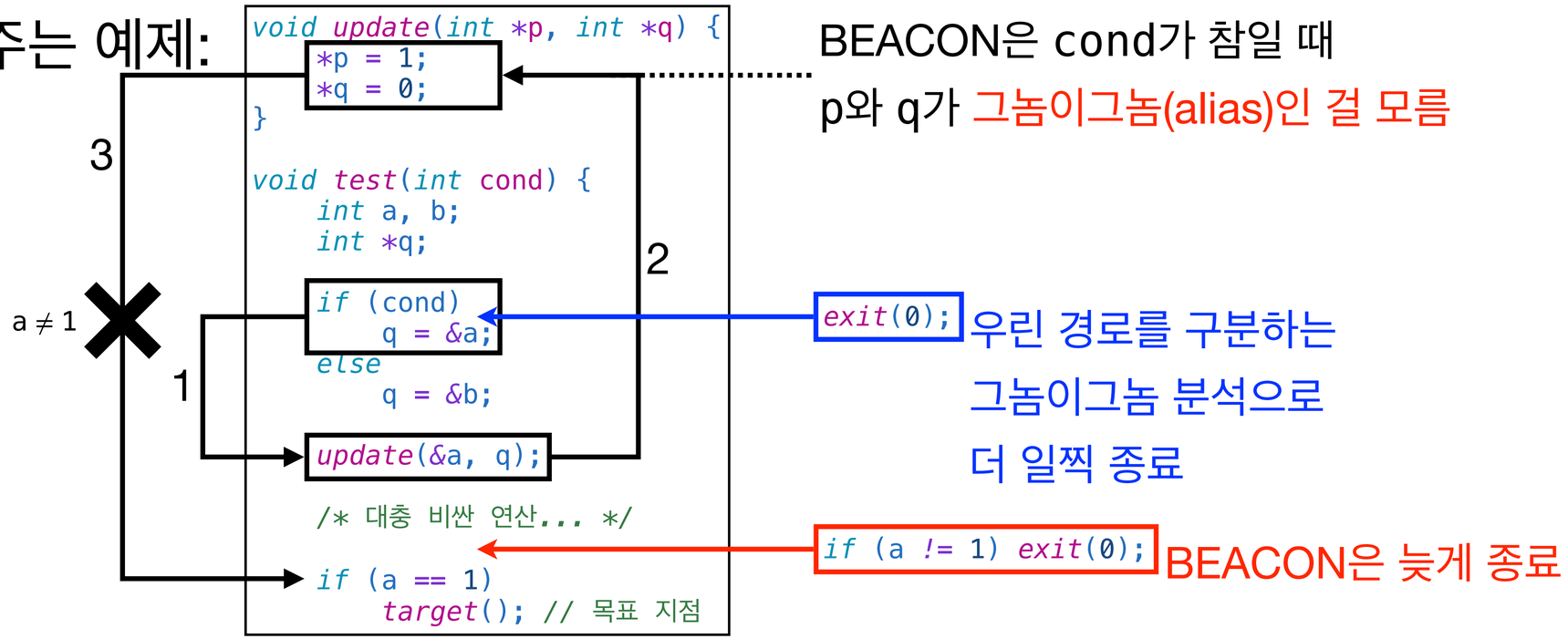
기존 조기종료 삽입 기술 vs. 우리 기술

- vs. SDFuzz (USENIX Security'24)
  - 실행 흐름 그래프상의 목표 지점 도달 불가능 분기 + α

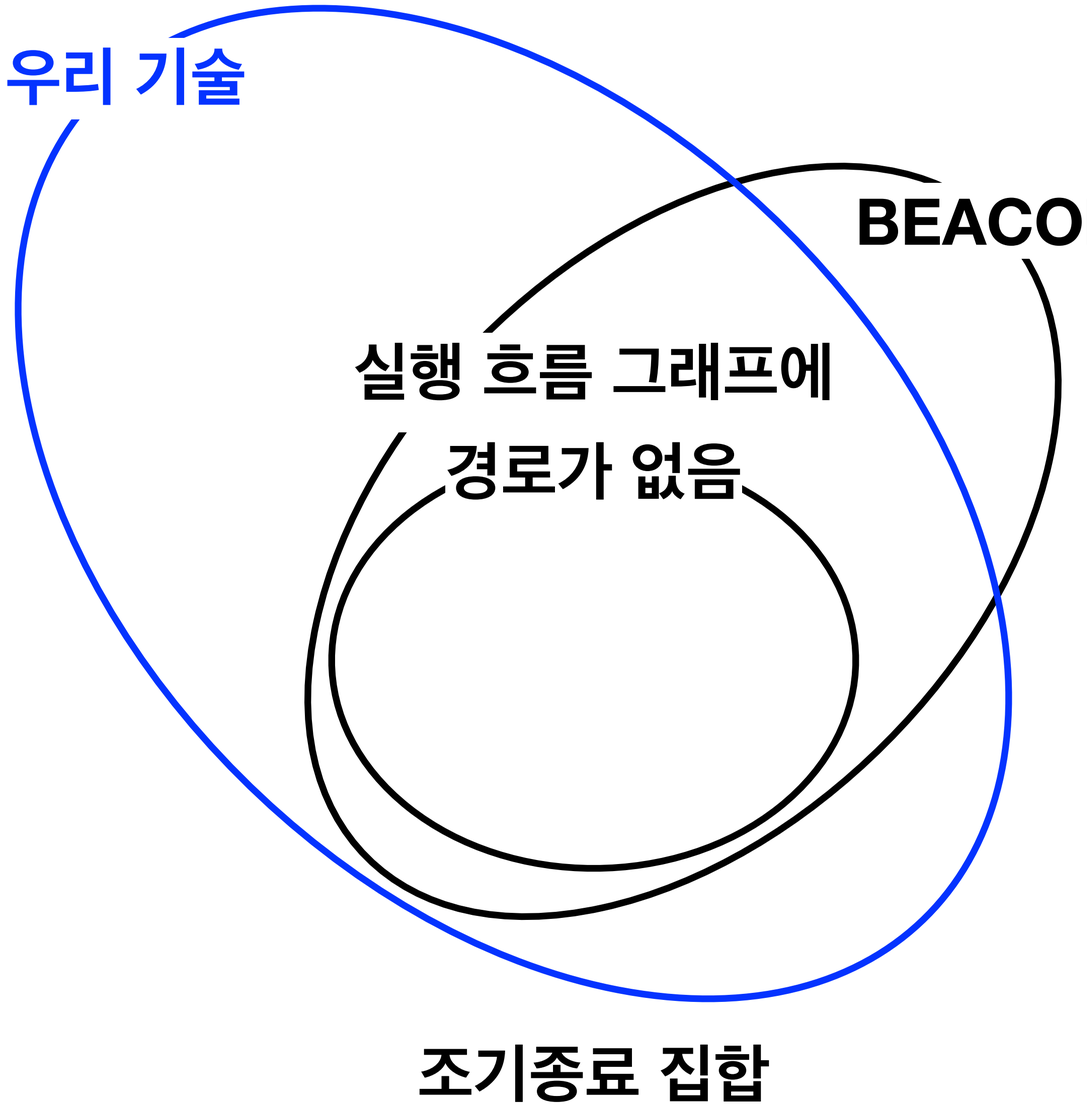


기존 조기종료 삽입 기술 vs. 우리 기술

- vs. BEACON (S&P'22)
  - 정수 구간 분석으로 목표 지점 도달의 전제조건 추론 → 위반 시 조기종료
  - 차이를 보여주는 예제:



계획



기존 기술보다 더 많이 조기종료

세 줄 요약

- 목표 지점과 관련 없는 부분에서 빨리 종료하면 지향성 퍼징 효율 좋아짐
- 기존 기술에 얹어보니 성능 꽤 좋아짐
- 더 정확한 분석으로 더 많은 곳에서 빨리 종료할 거임

세 줄 요약

- 목표 지점과 관련 없는 부분에서 빨리 종료하면 지향성 퍼징 효율 좋아짐
- 기존 기술에 얹어보니 성능 꽤 좋아짐
- 더 정확한 분석으로 더 많은 곳에서 빨리 종료할 거임

감사합니다 :)