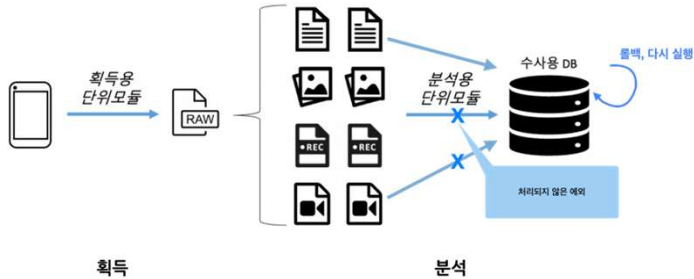


파이썬 프로그램에서 처리되지 않은 예외 분석: 디지털 포렌식 소프트웨어 중점으로

이서우

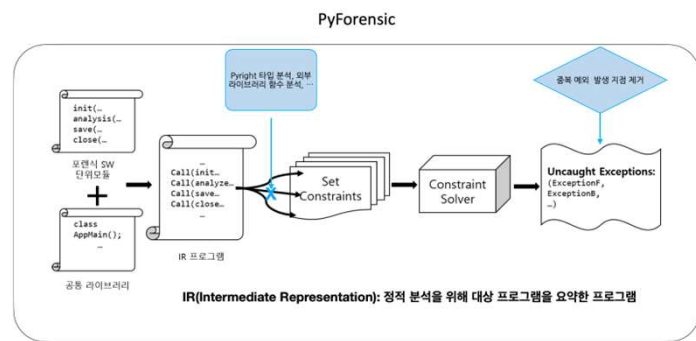
연구 배경

- 디지털 포렌식 소프트웨어
 - 디지털 증거의 누락 및 오염이 없어야함
 - 획득과 분석 부분으로 구성
 - DB와의 입출력이 빈번
 - 처리되지 않을 시 소프트웨어의 신뢰성 깨짐



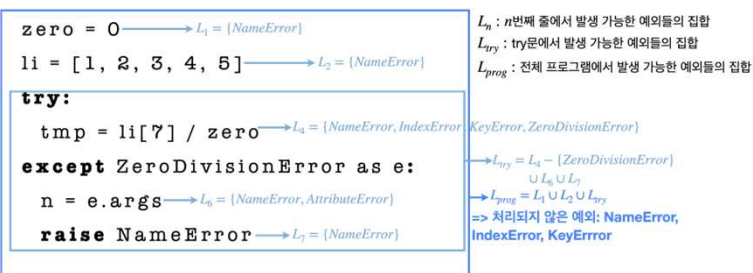
연구 목표

- 처리되지 않은 예외 분석
 - 집합 제약식 기반 분석 기법 활용
- 불필요한 경고 축소
 - Pyright 타입 분석 결과 결합
 - 중복된 예외에 대한 경고 제거



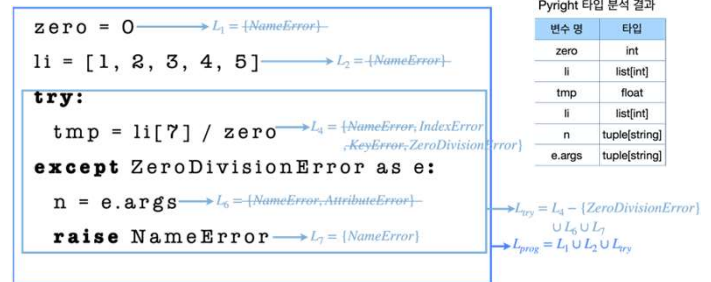
연구 핵심1: 집합 제약식 기반 분석

- 프로그램의 각 지점마다 발생 가능한 예외들의 집합 구함
 - 프로그램 지점마다 예외들의 집합에 대한 제약식 구함
 - 제약식들의 방정식의 해를 구함

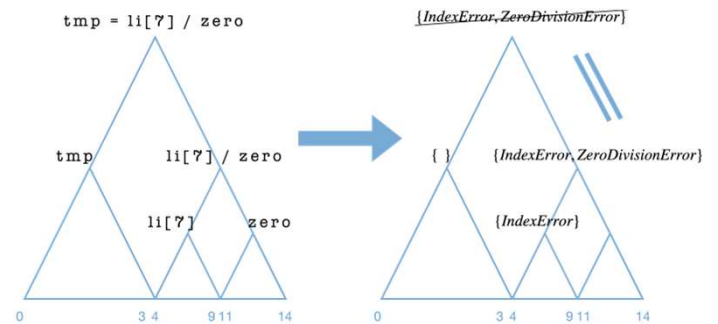


연구 핵심2: 불필요한 경고 제거

- 허위 경고 제거
 - Pyright로 각 변수 및 함수 호출 반환 값의 타입 분석
 - 타입 분석 결과에 따라 허위 경고 제거



- 동일한 예외 발생 지점에 대한 중복 경고 제거
 - 프로그램 지점 나열 후 순회
 - 중복되는 경고들 중 예외 발생 지점과 가장 가까운 프로그램 지점에 대한 경고만 남김



연구 결과

벤치마크	bench1	bench2	bench3	bench4	bench5	bench6	bench7	bench8	bench9	testing1	testing2	add_lib
불필요한 경고 제거 전	256	579	537	839	316	680	400	316	382	13	15	581
불필요한 경고 제거 후	30	83	80	191	47	139	62	47	69	5	2	161

- 벤치마크
 - 대검찰청 제공 디지털 포렌식 소프트웨어 벤치마크 9개
 - 외부 라이브러리 벤치마크 1개
 - 일반 예외 발생 파이썬 프로그램 2개
- 예외 발생 가능 패턴 3개 발견
- 대검찰청 9개 벤치마크 및 외부 라이브러리 포함 벤치마크에 대하여 평균 84%의 불필요한 경고 제거

의의

- 파이썬 코드 실행 전에 예외 발생 및 예외 발생 위치를 미리 알 수 있음