

# PL/SQL 변경 영향 분석기 “리뉴얼”

김용기, 이광근  
서울대학교 프로그래밍 연구실

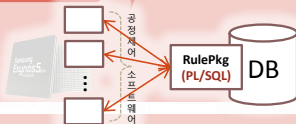
## 연구 배경 현장 개발자가 “잘 이해”하고 “실제 활용”할 수 있게 분석기를 개선하자

### 삼성반도체의 성공 비결은? 자동화 시스템

- 반도체 업계 최고의 생산 공정 제어 시스템
- PL/SQL언어로 작성된 판단 프로그램(RulePkg)

### 소프트웨어 변경 실패가 가장 큰 걱정...

- 변경 영향이 사전에 완전히 파악 안됨
- 예상 못한 오류로 인한 생산/품질 피해 심각



그런데, 분석기를 만들어줘도 쓰지를 못하네...

- 실제 코드는 매우 복잡하여 분석 정확도가 떨어짐
- 애매한 분석 결과가 많고 분석 로그는 개발자가 이해 불가

누구나 “쉽게” 쓸 수 있게 분석기를 개선해보자!

- 사용성 개선 → 더 많은 피드백
- 더 많은 피드백 → 정확도/사용성 개선

## 기존 성과 “잘 동작”하는 분석기

### 심볼릭 실행 기법을 이용한 효과적인 분석기

「2013년 삼성/ROPAS 산학 과제」

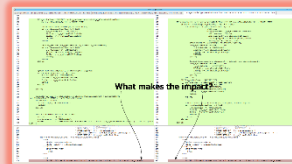
- OCaml로 작성 : 약 7,600라인, 개발기간 6개월
- 충분한 속도 : 40,000 라인을 5분 안에 분석
- 외부입력 불필요 : DB나 외부 입력 없이 효과적 분석

## “쓰기 힘든” 분석기 문제점

### 방대한 분석 결과 / 애매한 분석 내용 / 어려운 로그

- 실제 분석 대상 코드는 매우 복잡 (100개 이상의 예상 영향점, 3000줄 이상의 분석 로그)  
많은 분석 결과를 효과적으로 탐색할 수 있는 수단 부족  
분석기 설계자만 이해할 수 있는 분석 로그...

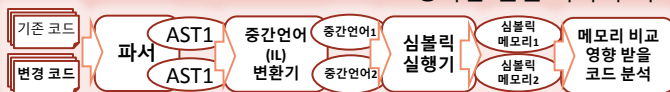
- 애매한 분석 결과:  
여러 변경점이 같은 예상 영향 코드를 가리킬 경우  
정확한 원인 파악이 어려움



### 심볼릭 실행 표현식과 조건문 그대로 기호형태로 모아서 계산해나가기

```

1 y="abc";          y: "abc"
2 z="cde";          z: "cde"
3 x = y || z;       x: "abc||"cde"
4 if(c){
5   a=1;
6 } else {
7   a=2;
8 }
    
```



구현 기술

### 결과 비교 함수의 결과값(리턴값, DB업데이트 등)을 시작점으로 거꾸로 탐색하며 비교해 나감

```

1 if (c) { 2 if (c) {
2   x = 1; 3   x = 1;
3 } else { 4   x = 2;
4   x = 2; 5   }
5 }
6 y = x + 1; 6 y = x + 1;
7 return y; 7 return y;
8
9 x1: 1, c
10 x2: 2, c
11 x3: 1(x1,x2), true
12 x3: 1, true
    
```

원했던 것 : **족집게 분석기**

VS

적용 결과 : **애매한 분석기**

## 연구 계획 사용성 → 정확도 순으로 개선 : 쉽게 쓸수 있는 정확한 분석기 만들기

### 분석 결과를 쉽게 알아볼 수 있게

- 사용성 개선이 선결과제  
분석기 개발자 ↔ 현업 개발자 간 상승효과
- 정확도 문제 해결도 반드시 필요  
명백히 영향 받는 코드를 빠뜨리는 결함 해결  
더 복잡한 분석 기술이 필요할 수도...
- 현업 개발자와 긴밀하게 협조  
개발자의 눈높이에 맞게 사용성/정확도 개선  
중요하지 않거나 가짜 영향지점을 걸러낼 필요



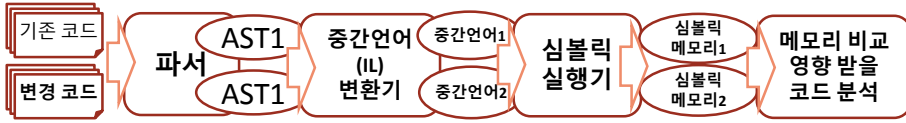
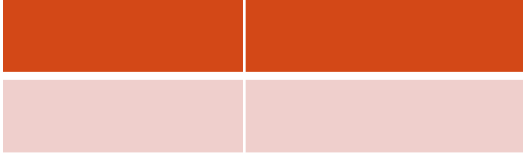
### 사용성 개선 아이디어

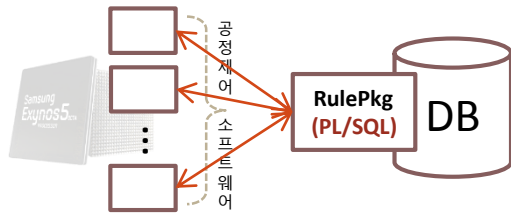
#### - 심볼릭 실행 과정 시각화



#### - 심볼릭 값 실시간 평가 UI







분석기 개발자  
**분석기  
사용성 개선**

분석기  
품질 향상

더 많은  
사용 유도

**선순환**

분석기 개발자  
**더 많은  
사용성 개선,  
결함 개선**

현업 개발자의  
피드백 증가

현업 개발자  
**분석 이해도 ↑  
더 많은 피드백**  
• 분석 누락 결함  
• 가짜 영향 리포트