

# 주관을 반영하는 사소한 수정 자동 판단 시스템

이석현, 전민석, 오학주  
소프트웨어 분석 연구실, 고려대학교

2023.07.06.

SW재난연구센터 워크샵 @머큐어 앰배서더, 울산



# 조교님, 다름이 아니라 ...



HW1 채점 관련해서 질문드립니다.

안녕하세요 조교님!

다름이 아니라 1번문항에서 조교님께 질문을 드리기 위해서  
(\* 안녕하세요 조교님! 혹시 실험해본 테스트케이스들을  
남겨두고 주석처리 해두어도 될까요?)

이렇게 주석처리를 통해 질문을 남겼으나 저의 불찰로 인해서  
주석을 완벽하게 닫지 못했습니다TT.

\*)로 주석을 닫지 못하여 ...

# 조교님, 다름이 아니라 ...



HW1 채점 관련해서 질문드립니다.

0점

```
let rec fib : int -> int
= fun n -> match n with
| 0 -> 0
| 1 -> 1
| _ -> fib(n - 1) + fib(n - 2);;
```

(\* 안녕하세요 조교님!  
혹시 실험해본 테스트케이스들을 남겨두고 주석처리 해두어도 될까요? )

# 과거의 목표: 작은 수정 자동 판별 시스템

- 주어진 수정이 작은 수정인지 판별하는 시스템



# 저자의 주관만 있는 데이터와 평가

- 나는 이거 봐주면 안된다고 생각하는데?

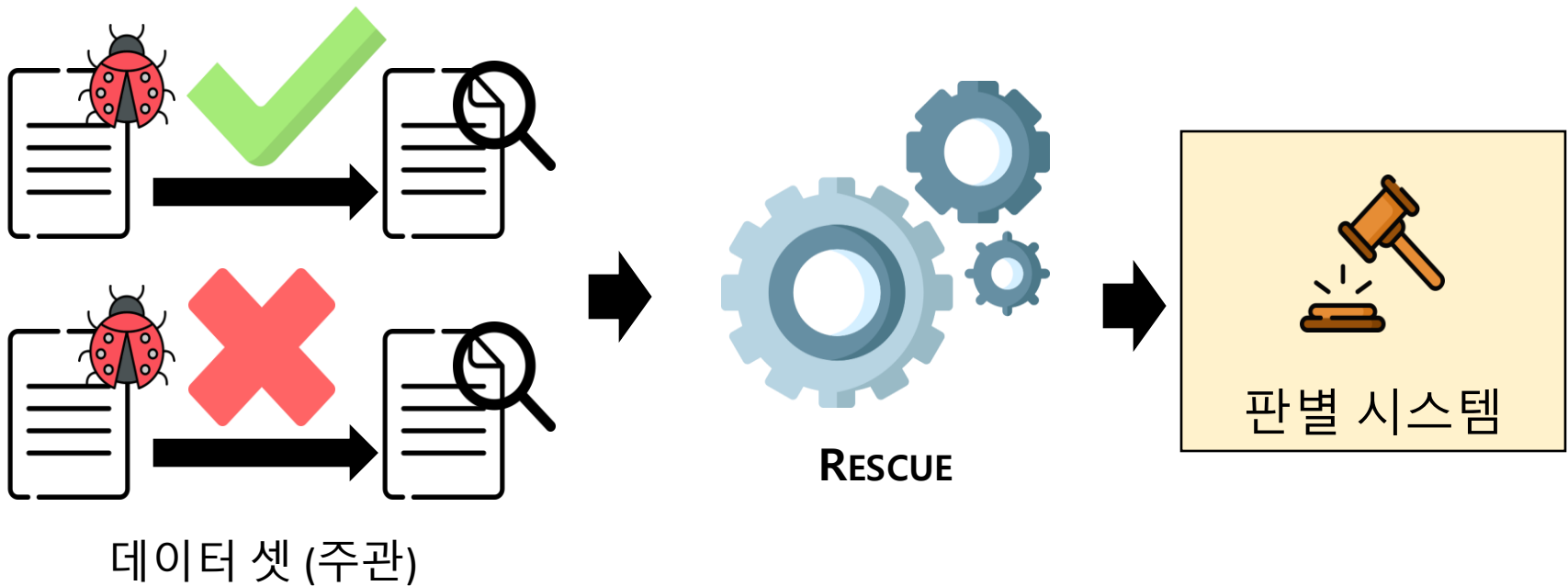
```
let min lst : int list -> int
  match lst with
  | [] -> raise (Out_Of_Constraint "")
  | [hd] -> hd
- | hd::tl -> if hd < (max tl) then hd else (min tl);;
+ | hd::tl -> if hd < (min tl) then hd else (min tl);;
```

학생이 실수로 잘못 적은게 아니라,  
몰라서 틀린거다!



# 목표: 주관을 반영하는 작은 수정 판별 시스템

- 작은 수정에 대한 주관이 반영된 판별 시스템 만들기



# (\* 포스터 세션에서 만나요 \*)

**주관을 반영하는 사소한 수정 자동 판단 시스템**  
 고려대학교 이석현, 전민석, 오학주

## 1. 학생들의 실수와 주관적인 사소함

**Case 1: 잘못된 작성된 주석으로 인한 컴파일 오류**

```
--- ocaml/case1/original.ml 2022-12-27 06:18:18.606634681 +0000
+++ ocaml/case1/revise.ml 2022-12-26 02:26:21.377147951 +0000
@@ -3,5 +3,3 @@
| 0 -> 0
| 1 -> 1
| _ -> fib(n - 1) + fib(n - 2);;
-
- (* 조교님, 이렇게 사용한 테스트 케이스를 주석으로 남겨 놔도 될까요? *)
```

**Case 3: 함수명 수정**

```
--- ocaml/case3/original.ml 2022-12-28 08:10:25.093633517 +0000
+++ ocaml/case3/revise.ml 2022-12-29 03:28:37.767792788 +0000
@@ -228,7 +228,7 @@
let solve : type_eqn -> Subst.t
= fun eqnlist -> unifyall eqnlist Subst.empty

-let typecheck: exp -> typ
+let typeof: exp -> typ
= fun exp ->
  let new_tv = fresh_tyvar() in
  let eqns = gen_equations TEnv.empty exp new_tv in
```

**Case 2: 명세와 다른 잘못된 오류 메시지**

```
--- ocaml/case2/original.ml 2022-12-28 07:28:03.736839156 +0000
+++ ocaml/case2/revise.ml 2022-12-28 07:28:00.056878153 +0000
@@ -88,7 +88,7 @@
let v2 = eval e2 env in
(match v1, v2 with
| Int n1, Int n2 -> Int (n1 + n2)
| _ -> raise (Failure "UndefinedSemantics"))
-
- -> raise UndefinedSemantics
+
+ | SUB (e1, e2) ->
  let v1 = eval e1 env in
  let v2 = eval e2 env in
  (* 15개의 추가적인 같은 함수 *)
```

**Case 4: 분기 문 삽입**

```
--- ocaml/case4/original.ml 2023-02-20 05:52:13.750169931 +0000
+++ ocaml/case4/revise.ml 2023-02-20 05:52:26.506004690 +0000
@@ -8,4 +8,5 @@
= fun list ->
  match list with
  | [] -> 0
  | [n] -> n
  | hd::tl -> max_function hd (max tl);;
```

## 2. RESCUE: 주관을 반영한 사소한 수정 자동 분류 시스템

Accept, if  $Dist(a, b, S) < \theta$

사소한 수정 분류기

학습 데이터 셋 (주관)

✓ Case 1, Case 2, Case 3

✗ Case 4

$S = Ident$

$\theta = \infty$

지난 가을 학기의 경우

✓ Case 1, Case 2

✗ Case 3, Case 4

$S = \emptyset$

$\theta = \infty$

더 엄격한 교육자

제한된 입력( $S$ )을 허용하는 수정 거리

$$Dist(a, b, S) = \begin{cases} |a| & |b| = 0 \\ 1 + Dist(a, t(b), S) & |a| = 0, b_0 \in S \\ \infty & |a| = 0, b_0 \notin S \\ Dist(t(a), t(b), S) & a_0 = b_0 \\ 1 + Dist(t(a), b, S) & b_0 \in S \\ 1 + \min \begin{cases} Dist(t(a), b, S) \\ Dist(a, t(b), S) \\ Dist(t(a), t(b), S) \end{cases} & \text{otherwise} \end{cases}$$

## 3. 평가

**1. RESCUE가 주관을 잘 반영하는지?**

- FatFinger: 173개의 사례 (자동 수정 기술 수정 포함)
- 75개의 사소한, 98개의 중대한 오류 수정
- 5개의 유명한 문자열 유사도 알고리즘과 비교

	Minor	Major	Total
RESCUE	92%	97%	95%
Levenshtein	82%	69%	72%
Jaccard	79%	69%	70%
Cosine	79%	69%	70%
Bag	80%	69%	70%
Compression	61%	86%	75%

**2. RESCUE를 사용하기 쉬운지?**

- FatFinger에서 20개의 학습 데이터 선별

	Minor (64)	Major (89)	Total (153)
RESCUE	97%	99%	98%
Levenshtein	97%	46%	67%
Jaccard	97%	47%	70%

**3. RESCUE가 바뀐 기준에도 잘 동작하는지?**

- 지난 가을 학기의 조교 보다 더 엄격한 조교가 FatFinger 분류
- 21개의 사소한, 152개의 중대한 오류 수정

	Minor	Major	Total
RESCUE	94%	97%	97%
Levenshtein	3%	96%	85%
Compression	0%	99%	87%