

Automatic Error Correction for Functional Programming Assignment

1. Motivation

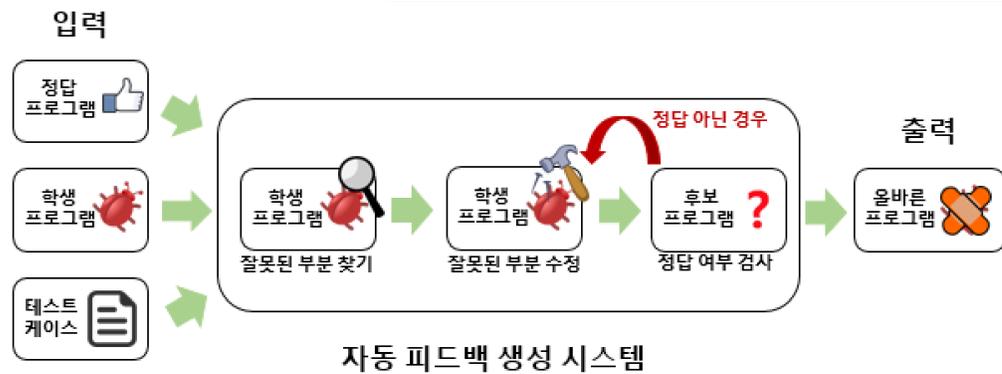
과제를 어려워하는 친구들을 보면서:
프로그래밍 과제 피드백을 자동화 할 수 없을까?



- 학생 : 점수는 받지만, 피드백이 없다.
- 교수자: 일일이 피드백을 주기 어렵다 (2~3명 vs 80 명 이상).

2. Research Goal

프로그래밍 과제 피드백 자동화

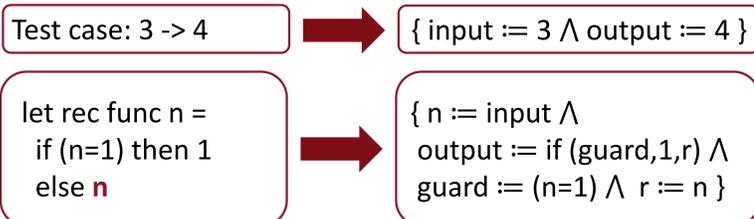


3. Algorithm

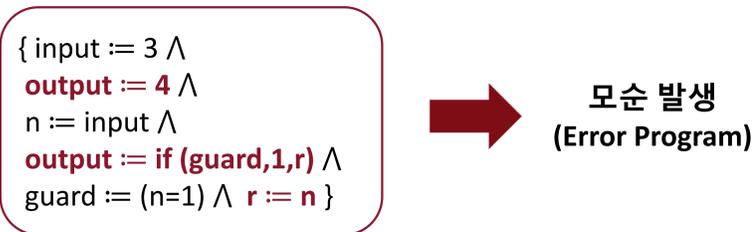
3.1 잘못된 부분 자동으로 찾기 Maximum Satisfiability(MAX-SAT) 문제로 해결

(1) 오류 탐색을 SAT Problem으로 변환

- 프로그램과 테스트케이스를 논리식으로 표현

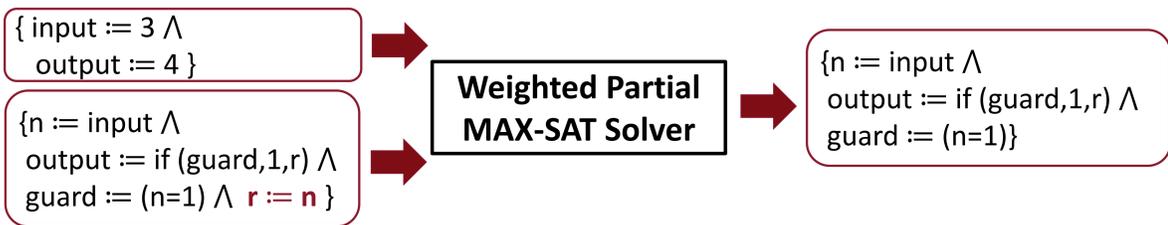


- 논리식에 모순이 생기면 오류가 존재



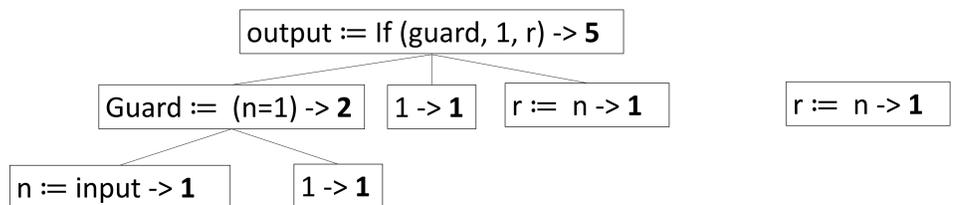
(2) MAX-SAT을 이용해 최소한의 오류를 탐색

- 모순을 만들지 않는 최대의 논리식



필수 조건을 충족시키며 모순을 만들지 않는 최대의 논리식

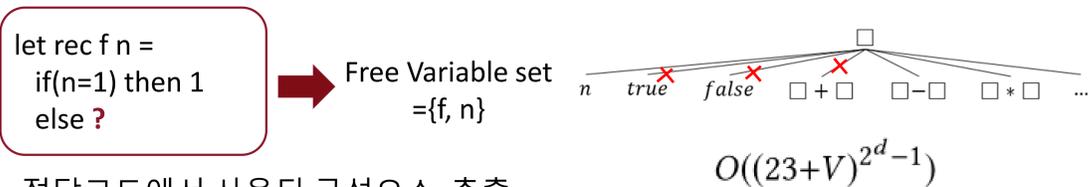
- 논리식의 Weight를 부여해 가장 큰 논리식 계산



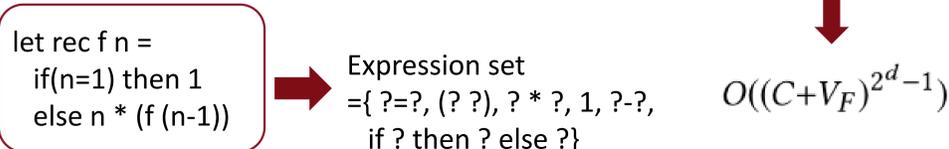
3.2 잘못된 부분 자동 교정 타입과 프로그램 구성요소를 이용한 프로그램 합성 문제로 해결

(1) 구성요소 추출

- 합성을 진행할 위치에 사용 가능한 변수를 추출

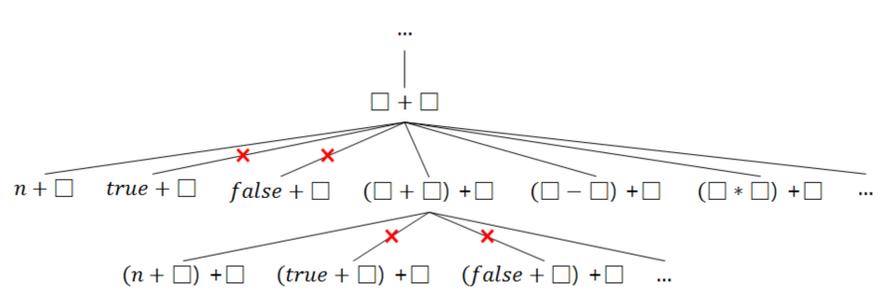


- 정답코드에서 사용된 구성요소 추출



(2) 타입 시스템을 활용

- 옳지 않은 타입을 가진 프로그램의 경우 가지치기



4. Evaluation

- ✓ 프로그래밍 언어 수업에서 사용된 문제들
- ✓ 탐색 기법을 적용하지 않은 알고리즘과 비교
- ✓ 평균적으로 380배 정도의 향상

| Test | Basic (sec) | Component (sec) | Type + Component (sec) | 속도 향상 |
|-------------|-------------|-----------------|------------------------|-------|
| List zipper | 133 | 0.488 | 0.176 | 760x |
| Prime | 0.1 | 0.016 | 0.02 | 5x |
| List map | Time out | 10.96 | 2.06 | n/a |
| Factorial | 380.064 | 6.764 | 0.936 | 400x |

실행 환경 OS Kali-linux 2016.1/ Intel Core i7-7700/2 Core/16GB DDR4

5. Conclusion

- ✓ 코드의 의미상 오류를 자동으로 찾는 새로운 알고리즘 제안
 - 오류 위치 추정 : MAX-SAT 기법 이용
 - 오류 자동 수정 : Program Synthesis 기법 이용
- ✓ 기존의 연구와 다르게 실제 언어에 대해 효율적으로 코드 생성