대체 Python 구현체의 표준 라이브러리에서 기능 오류 찾기

2026. 2. 2.

STAAR 2026 겨울 정기 워크숍

임정섭, 이석현, 오학주

목차

문제 정의

대체 Python 구현체의 표준 라이브러리 API에서 기능 오류 찾기

문제 정의

대체 Python 구현체의 표준 라이브러리 API에서 기능 오류 찾기

What? CPython이 아닌 Python 인터프리터

- CPython보다 성능 우수
- Python 프로그램을 다양한 환경에서 실행
  (Java/JVM, Rust, 마이크로컨트롤러)

Why?

- IoT, 암호화폐 지갑, 의료기기 등 산업 활용 (MicroPython)
- 전반적인 Python 생태계에 기여

문제 정의

대체 Python 구현체의 표준 라이브러리 API에서 기능 오류 찾기

What?

Why?

- 모든 Python 구현체의 필수 구성 요소
- 경험적으로 가장 많은 버그가 나오는 영역* [Wang et al., 2022]
- Python 구현체에서 명세 및 구현이 자주 수정되는 영역

* CPython 기준

문제 정의

대체 Python 구현체의 표준 라이브러리 API에서 기능 오류 찾기

What? 대체 Python 구현체가 CPython과 의미론이 상이

- Python 표준 라이브러리의 형식 명세가 부재
- 대체 구현체가 CPython 호환성 추구 (e.g., 공식문서, 버그 리포트)

Differences between PyPy and CPython

This page documents the few differences and incompatibilities between the PyPy Python interpreter and CPython. Some of these differences are “by design”, since we think that there are cases in which the behaviour of CPython is buggy, and we do not want to copy bugs.

이 페이지에 나열되지 않은 차이는 PyPy의 버그로 간주됩니다.

🔗 PyPy documentation

Undefined variables shouldn't be deleted in the function scope #4910

Closed #4979

xiaxinmeng opened on Apr 21, 2023

예상 출력은 UnboundLocalError인데, RustPython은 정상 값을 내보냅니다.

🔗 RustPython GitHub Issue #4910

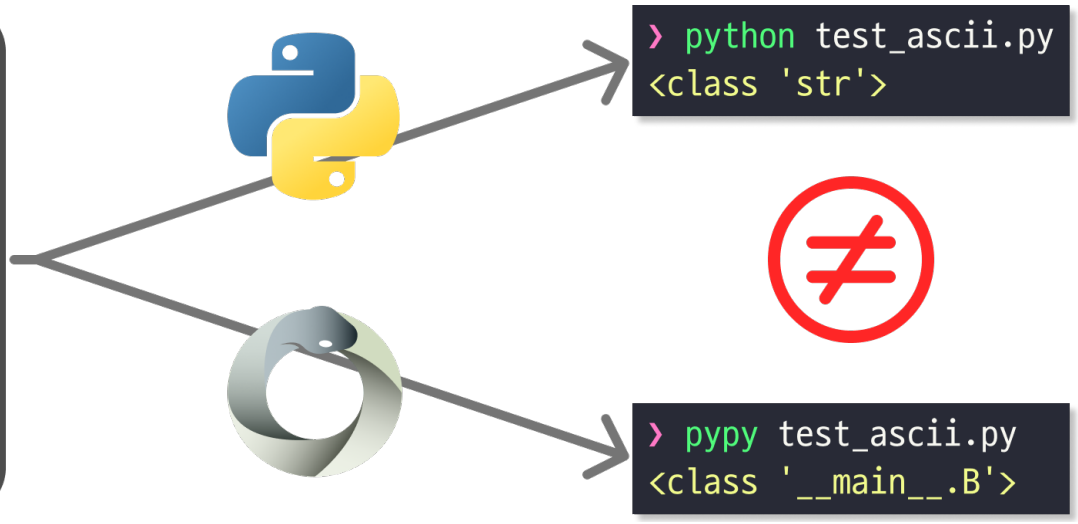
문제 정의

대체 Python 구현체의 표준 라이브러리 API에서 기능 오류 찾기

```python
# test_ascii.py
class A:
    def __repr__(self):
        return B("abc")

class B(str):
    pass

result = ascii(A())
print(type(result))
```



선행 연구

대체 Python 구현체의 표준 라이브러리 API에서 기능 오류 찾기

DyFuzz
[QRS'23]

PyRTFuzz
[CCS'23]

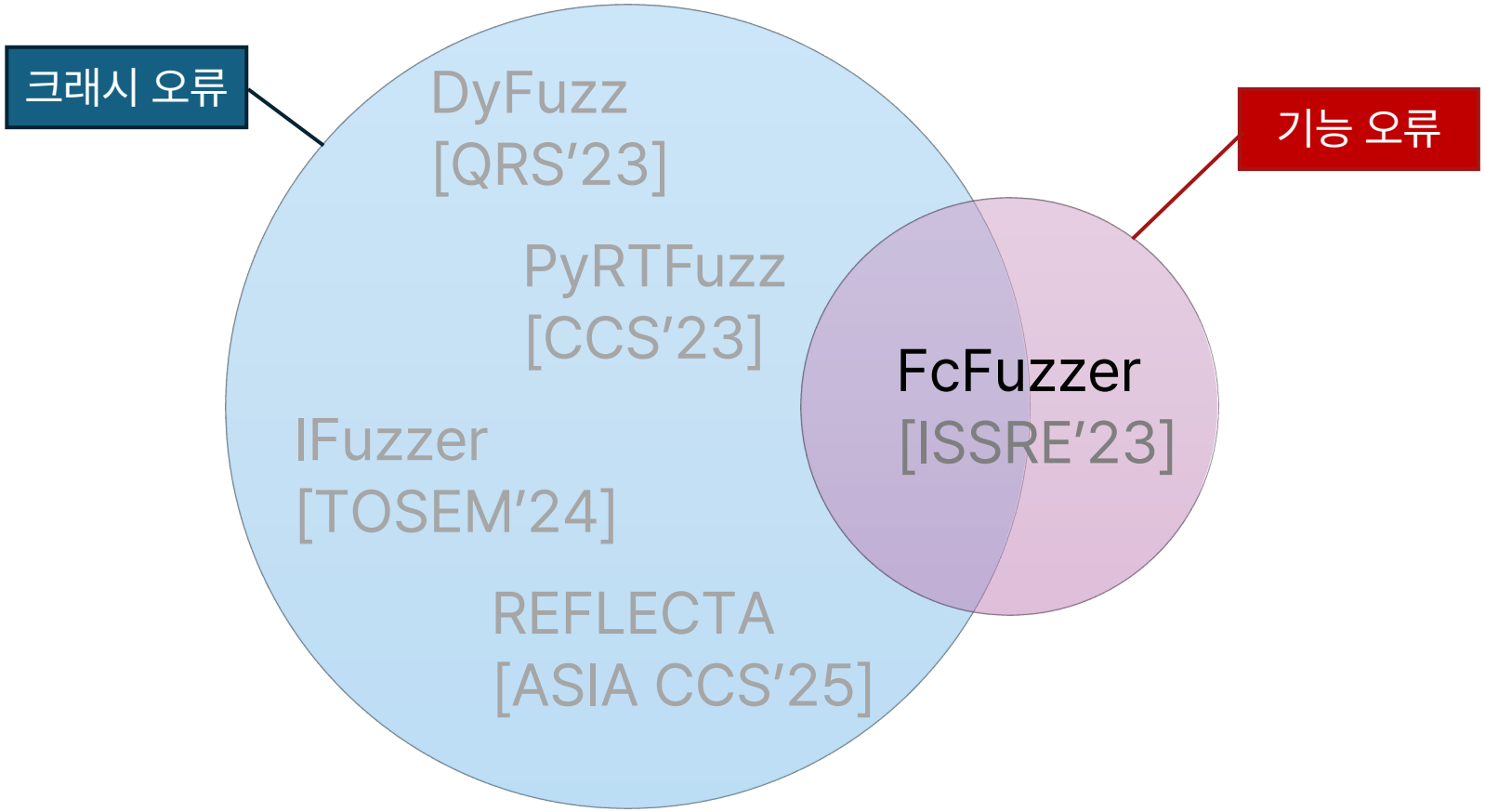
FcFuzzer
[ISSRE'23]

IFuzzer
[TOSEM'24]

REFLECTA
[ASIA CCS'25]

선행 연구

대체 Python 구현체의 표준 라이브러리 API에서 기능 오류 찾기



FcFuzzer [ISSRE'23] 요약

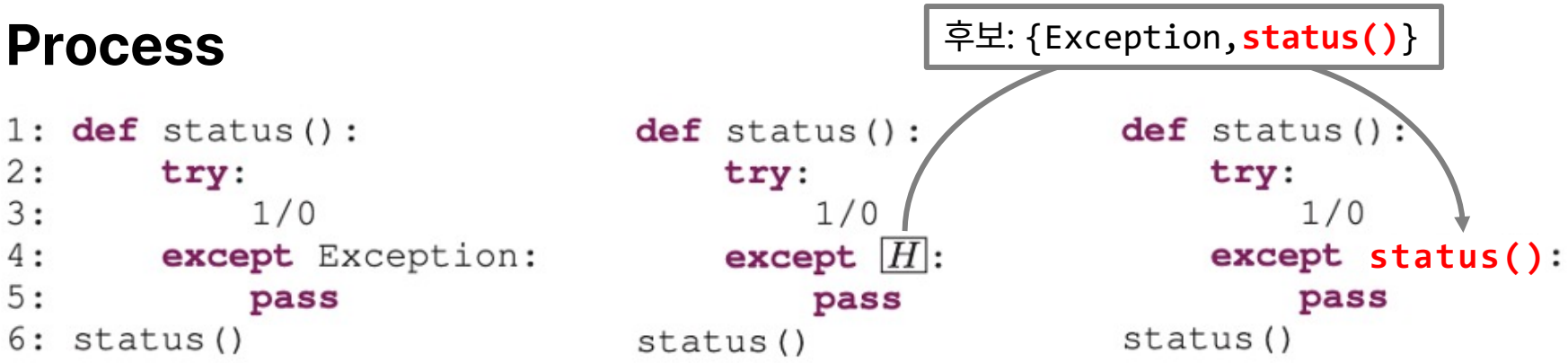
Problem Python 구현체 전체에서 크래시 & 기능 오류 찾기

Key Idea 프로그램에 구멍을 뚫고 랜덤으로 채우는 SPE(스켈레톤 프로그램 열거) 기법에서, 함수 호출을 고려하자.

Contributions

1. 함수 호출까지 탐색하여 스켈레톤의 구멍을 채우는 알고리즘
2. 제안 기술을 Python 인터프리터에 적용

Process



(a) Seed (b) Skeleton (c) Test program

목차

아이디어 직관

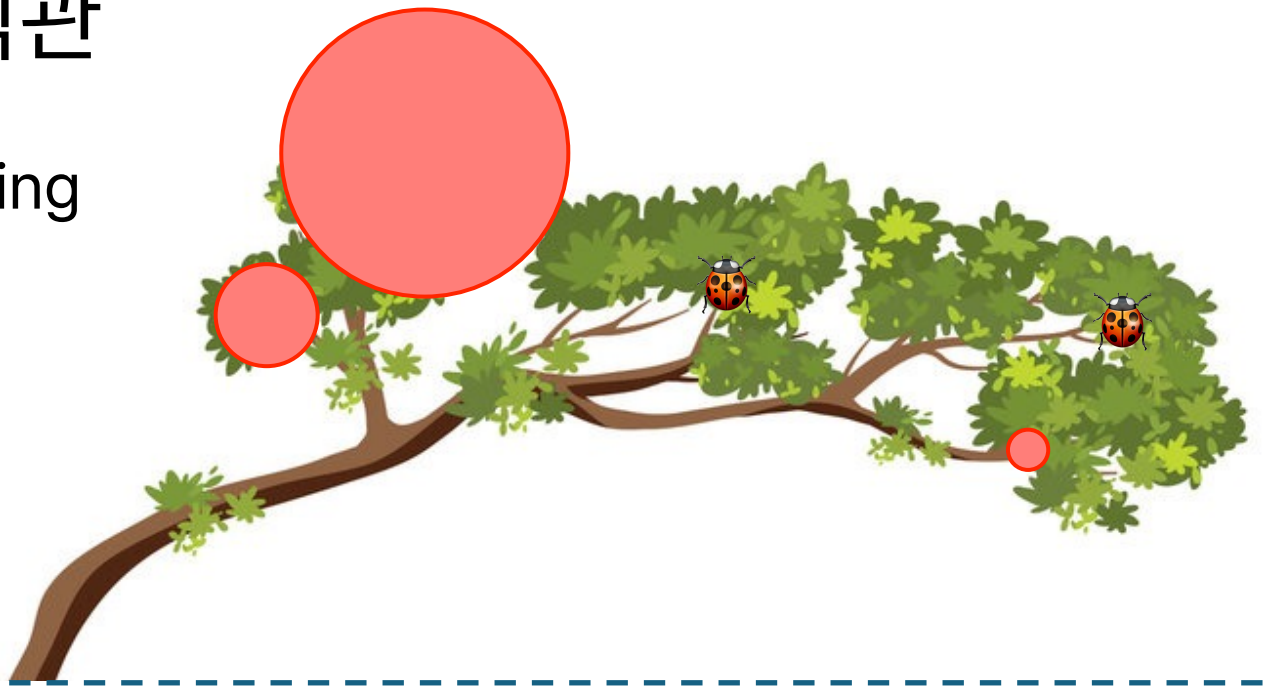
나무가지를 탐색하여 벌레를 잡자!

- 나무가지: 표준 라이브러리 CPython 소스코드의 모든 분기
- 벌레: 특정 분기로 실행되는 프로그램에서 발생하는 기능 오류

기능 오류는 CPython 상 다양한 분기에 존재한다.

아이디어 직관

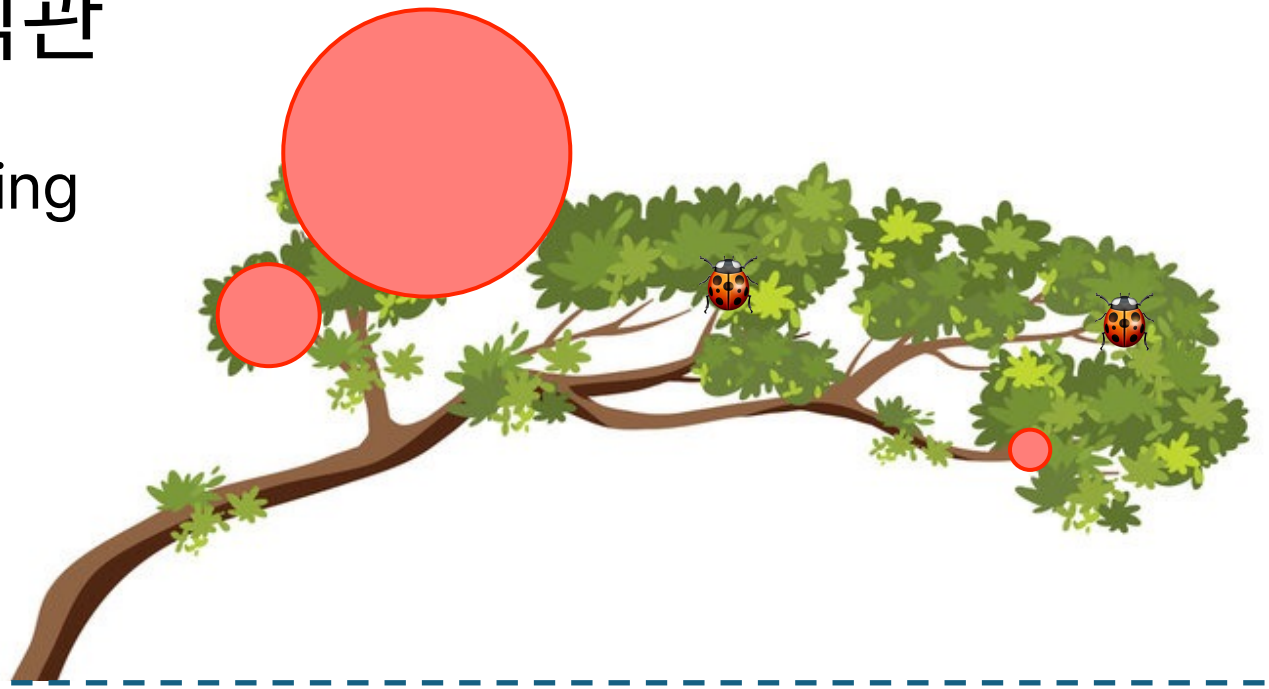
Blackbox Fuzzing
(FcFuzzer)



Ours

아이디어 직관

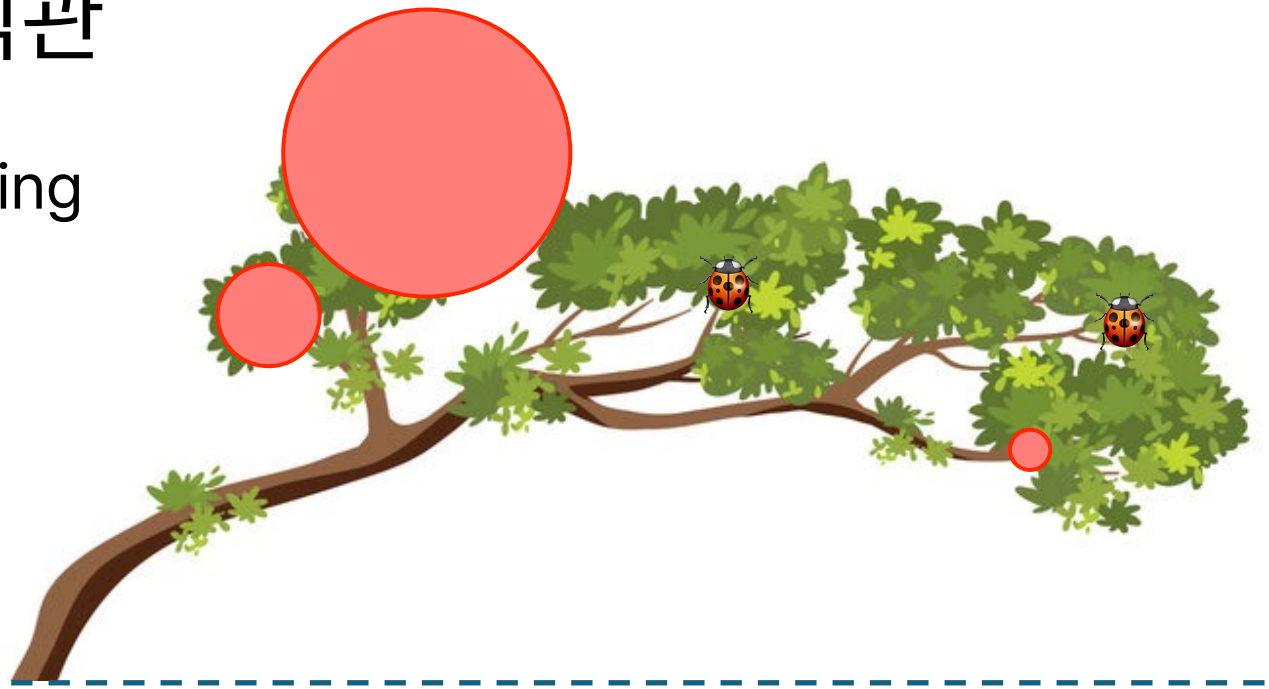
Blackbox Fuzzing
(FcFuzzer)



Ours

아이디어 직관

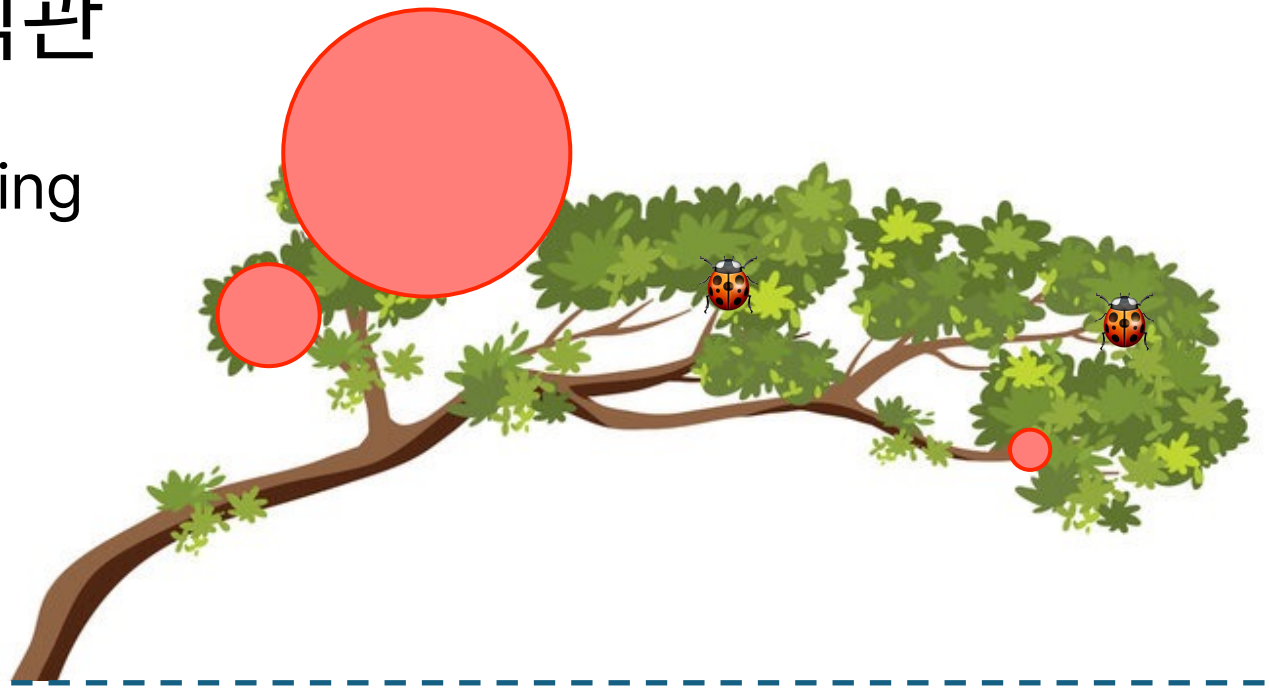
Blackbox Fuzzing
(FcFuzzer)



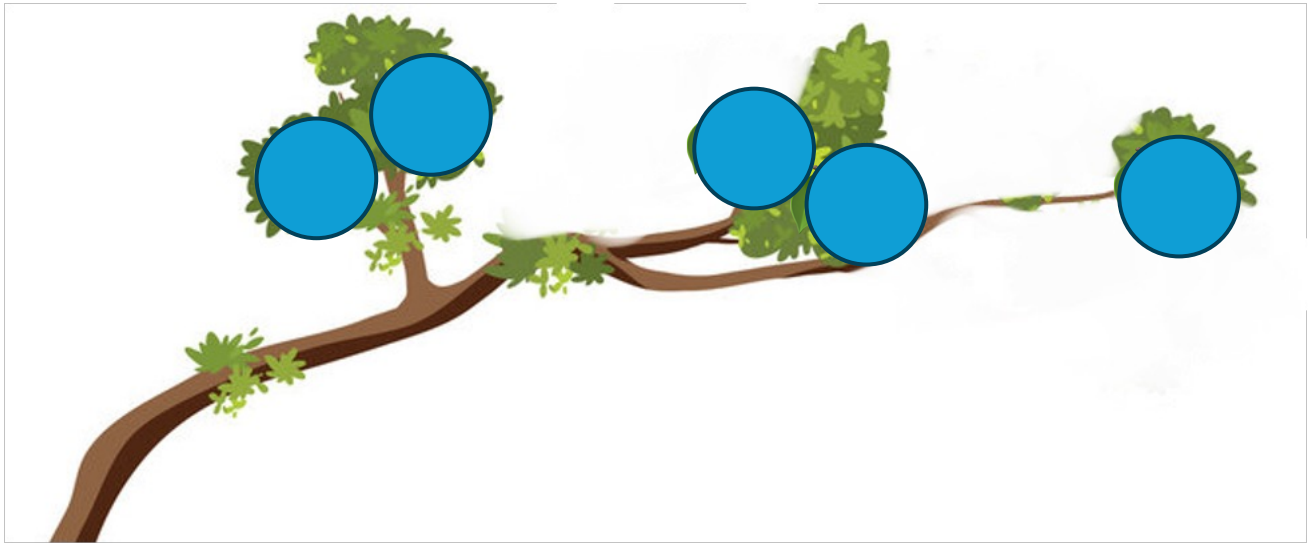
Ours

아이디어 직관

Blackbox Fuzzing
(FcFuzzer)



Ours



아이디어 직관

나무가지를 탐색하여 벌레를 잡자!

- 나무가지: 표준 라이브러리 CPython 소스코드의 모든 분기
- 벌레: 특정 분기로 실행되는 프로그램에서 발생하는 기능 오류
- 과일: 표준 라이브러리 API 인자의 Python 타입 제약

Observation 기능 오류는 Python 타입 제약에 관련된 분기에 존재!

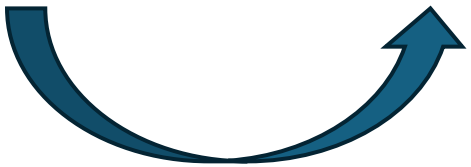
목차

연구 질문

1. **효과성**: 기존 기술 FcFuzzer에 비해 오류 많이 찾는가?
2. **???**

표 1. 실험 결과 레이아웃 (채워진 숫자는 현재 기준, 최저치)

Implement	FcFuzzer		Ours	
	known	unknown	known	unknown
PyPy	2	-	1	⭐
GraalPython	-	-	-	⭐
MicroPython	-	-	-	⭐
RustPython	1	-	-	⭐



(기대) unknown 개수 증가 !

마무리

세 줄 요약

- 문제: 대체 Python 구현체의 표준 라이브러리에서 기능 오류 찾는다.
- 접근: 답지인 CPython의 분기를 선별하여 전부 탐색한다.
- 실험: known 버그 하나(`ascii`) 찾았다.

향후 계획

- 목표: 얼른 unknown bug를 많이 찾자!

오늘 저녁 포스터 발표에서 만나요!

(고려대 / 오학주 교수님 연구실)