

OOPSLA 2025 Trip Report

2025.10.14 ~ 2025.10.19, Singapore

고려대학교 소프트웨어분석연구실 김동욱



1 들어가며

내가 진행했던 결함 위치 추정 기법 연구인 “**PAFL: Enhancing Fault Localizers by Leveraging Project-Specific Fault Patterns**”에 대해서 발표하기 위해 싱가포르에서 열린 OOPSLA 2025에 참가하였다. 이번 학회 참가로 내가 경험했던 것과 느낀 바를 이 글을 통해 공유하려고 한다.

2 발표 후기

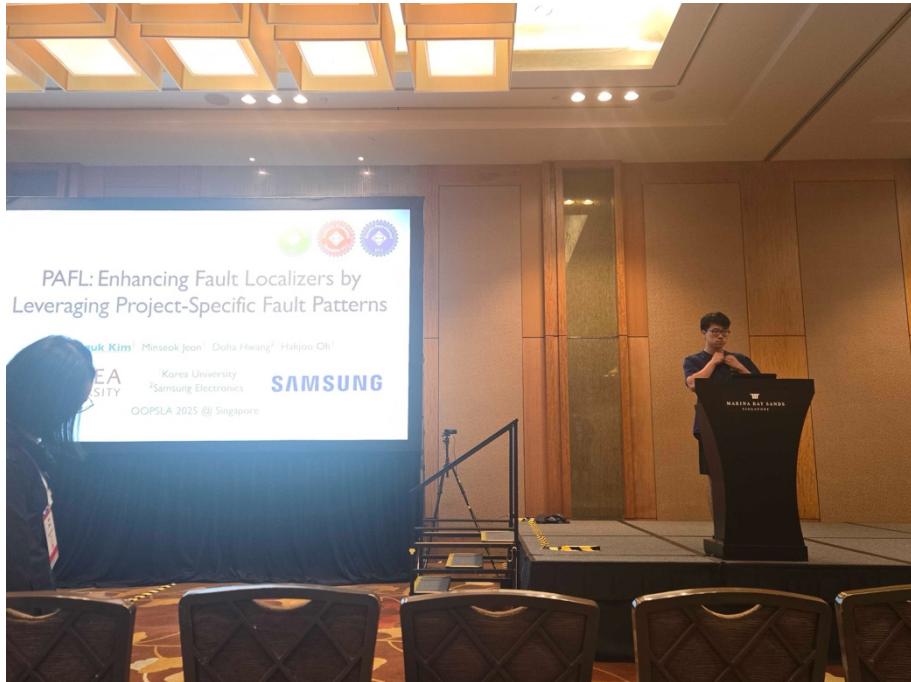


Figure 1. 발표 준비.

이번 학회 이전에 마지막으로 이 연구를 발표했던 시기는 2024년 8월 쯤이었다. 벌써 1년이 넘게 지나가면서, 발표에 사용할 자료, 풀어나가야 할 이야기 흐름까지 다시 구성할 필요가 있었다. 발표를 준비하면서 가장 중점에 둔 부분은 프로젝트에 반복되는 결함을 고려해야 한다는 핵심 관찰을 전달하는 것이었다. 연구의 주제가 PL 학회에서는 거의 다루지 않던 이질적인 내용이라 어떻게 설명할지 고민이 되는 지점들이 많았는데, 결국 제목을 보고 관심이 있는 사람들만 발표를 들으러 오기 때문에 학회의 성격이 다르다고 발표의 양상이 크게 달라질 필요는 없는 것 같다.

발표 현장에서 미리 발표 준비를 할 때 케이블 연결 이슈가 발생해서 많이 당황스러웠다 (거의 모든 발표자 분들에게 이슈가 생겼었다). 어떤 발표자 분은 발표 시간의 거의 1/3이 화면 끊김으로 날라갔는데 그런 와중에도 적절하게 발표 내용을 수정하면서 시간을 정확하게 지키는 모습이 인상깊었다.

3 학회 발표들

Fuzzing C++ Compilers via Type-Driven Mutation

사람마다 학회에 참여할 때 어떤 발표를 들을지 정하는 나름의 기준들이 있을 것이다. 나의 제1순위 기준은 해당 발표가 C++과 관련된 연구인지 여부이다. 그래서 이번 학회에서는 이 발표가 먼저 내 눈길을 끌었다.

이 연구는 C++ 컴파일러를 퍼징하기 위한 새로운 변이 방법을 제안한다. 저자들은 C++ 컴파일러의 복잡한 타입 시스템 때문에 올바른 C++ 컴파일러 만들기가 어렵다고 지적한다. 이에, C++ 컴파일러의 타입 시스템을 구석구석 퍼징하기 위한 DSL과 변이 연산자들을 디자인한다.

첫 번째로, C++ 프로그램 분석을 위해 MiniCpp라는 C++ 프로그램의 타입 시스템을 표현하는 DSL을 정의한다. 이 DSL은 상속관계, 호출 최적화(noexcept), 가상화, 명시성/암시성, 타입 캐스팅, 타입 앤리어스, 템플릿, 기타 등등의 타입 시스템을 포함한다. 두 번째로, MiniCpp DSL 위에서 5가지 카테고리의 30가지 변이 연산자들을 정의한다. 그 후, 이 연산자들을 기반으로 valid 프로그램과 invalid 프로그램을 구분하며 컴파일러를 테스팅한다.

발표를 듣고 가장 먼저 들었던 생각은, 위에서 길게 나열한 것처럼 C++ 타입 시스템이 정말 짜증날 정도로 다양하고 복잡한데 이 것들을 전부 커버하려고 저자들이 많이 고생했을 것 같다는 생각이었다. 또한, 평소에 내가 생각했던 C++ 언어 분석의 도전 과제를 저자들이 지적해주어 특히 더 기억에 남았다. C++은 산업 전반에 깊게 뿌리내린 기반 언어이지만 새로운 프로그래밍 패러다임을 계속 도입해서 (제너릭, RAII, 함수형, ...) 언어가 지나치게 비대해졌다. 이런 복잡한 C++ 언어를 어떻게 다룰 것인지는 계속해서 도전 과제로 남을 것 같다.

Interleaving Large Language Models for Compiler Testing

이 연구도 앞에서 소개한 연구와 비슷하게 컴파일러의 버그를 찾기 위한 새로운 퍼징 기법을 제안한다. 구체적으로는, LLM을 활용해 입력 프로그램을 생성하는 LLM-driven 기법에 기반한다.

이 연구는 기존의 비현실적인 LLM 기반 기술들을 한계점을 명료하게 지적하고 있고, 올바른 LLM 활용 방안에 대해 생각할 거리를 주기 때문에 공유하고자 한다. 많은 연구들이 LLM 기반 기술들을 일종의 on-demand 방식으로 동작시킨다. 예를 들어 퍼징의 경우, 시드 프로그램에서 새로운 입력을 만들어 낼 때마다 LLM 서버에 쿼리를 날리는 식이다. 사실 이러한 방식은 비용적 문제로 인해 비현실적인 경우가 많다. 우리가 더 나은 기능의 VSCode 확장자를 사용하기 위해서 데스크탑에 500만원 가량의 rtx 5090이 필요하다거나 매 실행마다 LLM API 비용을 내야한다고 생각해보면, 차라리 예전의 non-LLM 기술들을 사용하고 싶을 것이다.

이 연구에서는 LLM 활용과 퍼징 실행을 분리된 단계로 구성한다. 먼저, LLM을 이용하여 대상 언어의 시맨틱을 다양하게 사용하는 고품질의 코드 조각들의 데이터베이스를 구성 한다. 그후, 오프라인으로 만들어진 재활용 가능한 데이터베이스를 이용해 컴파일러를 효과적으로 테스팅하기 위한 입력 프로그램 합성 기법을 퍼징 단계에서 실행한다.

4 싱가포르



Figure 2. 학회 장소인 마리나 베이 샌즈.

싱가포르 창이 공항에 도착해보니 10월 중순임에도 열대야가 지속되는 찌는 더위였다. 더위를 많이 타는 체질은 방비책을 미리 마련해 가는게 좋을 것 같다. 그래도 국토 대부분이 (냉방이 잘 되어있는) 건물들로 꽉 들어차있기 때문에 실내 위주로 다닌다면 괜찮은 편이다. 냉방이 너무 잘 되어있어서 축구장 두 개 크기의 (2 헥타르) 초대형 온실 전체를 시원하게 해 놓을 정도이다. 이 온실들, 플라워 돔과 클라우드 포레스트도 재미있게 잘 꾸며놓았으니 한 번 가보는 것을 추천한다.

싱가포르에서 제일 눈에 띄는 풍경은 역시 인공 구조물, 특히 각양각색의 건물 양식들과 웅장한 마천루다. 어차피 학회 일정이 늦게 끝나기도 해서 며칠 동안 싱가포르의 인공미 넘치는 야경들을 즐길 수 있었다.

추천 여행지: 나이트 사파리 (만다이 야생동물 보호구역)



Figure 3. 트램. 기둥을 제외한 공간은 전부 뚫려있다.



Figure 4. 하이에나.

싱가포르 만다이 야생동물 보호구역에는 밤에만 입장할 수 있는 독특한 형태의 사파리가 있다. 트램을 타고 둘러보다가 운이 좋으면 야생동물을 매우 가까운 거리(3m 이내)에서 가림막 없이 직접 볼 수 있는 신기한 경험을 할 수 있다. 보호구역 내에는 나이트 사파리 이외에도 여러 어트랙션들이 있기 때문에 새를 좋아한다면 낮에는 버드 파라다이스, 밤에는 나이트 사파리를 구경하는 코스를 추천한다.



Figure 5. 즐고 있는 유황앵무 in 버드 파라다이스.

5 마치며

내 처음 논문이자 처음 국제학회 발표를 무사히 끝맞쳤다는 생각에 매우 기쁘다. 이번 OOPSLA 2025를 시작으로 재밌고 의미 있는 연구들을 해내고 싶다. 끝으로 2년 동안 이 연구를 지도해주시고 새내기 대학원생에게 많은 가르침을 주신 전민석 교수님과 오학주 교수님에게 깊은 감사를 드립니다.