

ICSE 2023 TRIP REPORT

Melbourne, Australia
14 - 20 May 2023

황도연

ICSE 2023에 참석하기 위해 5월 13일부터 21일까지 호주 멜버른에 다녀왔다. ICSE(International Conference on Software Engineering)는 소프트웨어 엔지니어링 분야에서 최고 수준의 학회로 소프트웨어에서 발생하는 다양한 문제와 발생하는 이슈를 해결하기 위한 새로운 기술들을 확인해볼 수 있던 좋은 기회였다. 이 리포트를 통해 내가 학회 참석 기간 동안 느꼈던 경험과 생각들이 잘 전달될 수 있기를 바란다.

ICSE

ICSE는 소프트웨어 엔지니어링이라는 광범위한 주제를 다루는 학회답게 우리 연구실에서 친숙한 주제인 퍼징, 버그 자동 수정, 정적 분석과 같은 기술 뿐만 아니라 'Mining software repositories', 'Cognitive aspects of software development'와 같은 다소 생소한 주제에 대해서도 세션이 있었다. 이번 ICSE에서 얻은 첫 번째 교훈은, 잘 알지 못하는 세션의 발표를 듣는 게 도움이 된다는 것이다. 좋은 연구는 좋은 방법론보다는 좋은 문제로부터 나오고, 그렇기 위해서는 연구의 motivation을 직관적으로 설명할 수 있어야 한다. 배경지식이 없는 생소한 분야임에도 나에게 연구 motivation을 잘 설득시키는 발표들은 큰 참고자료가 되었다. 교수님이 항상 강조해주시는 것처럼 '왜 이 연구가 필요한지'에 대해서 잘 설명할 수 있어야겠다는 생각을 다시 한 번 하게 되었다.

이번 학회를 통해 얻은 두 번째 교훈은, 최대한 다른 사람과 많이 대화해보라는 것이다. 한국에서는 모르는 사람에게 말을 거는 것이 쉽지 않지만 해외 학회에서는 "잠깐 이야기할 수 있을까?"라는 말과 함께 인사를 건네면 상대방의 호의와 함께 대화를 할 수가 있었다. 다른 나라의 연구실에서는 어떤 연구를 하고 있는지, 연구에 있어서 어떤 것을 중요하게 생각하는지, 새롭고 다양한 관점을 들을 수 있는 귀중한 시간이었다. 또한 영어가 유창한 연구자와의 대화는 현재 내 영어 실력에 대한 좋은 가늠자가 되기도 했다. 학회에서 영어로 많은 것을 해보려고 노력한 덕에 내가 어느 부분이 약한지 잘 파악할 수 있었다. 다음에 참가하게 될 학회에서는 이를 잘 보완해서 더 생산적인 대화를 나누어보려고 한다.

논문

Concrat: An Automatic C-to-Rust Lock API Translator for Concurrent Programs

Sigpl에서도 본 적이 있던 카이스트 홍재민씨의 연구이다. C 코드에서 발생할 수 있는 잘못된 lock api 사용을 방지하기 위해 해당 코드를 올바른 lock 사용을 보장하는 Rust 코드로 자동 변환하는 것을 목표로 하는 연구이다. 현재 활발하게 사용되는 번역기인 *c2rust*는 문법적으로만 C 코드를 Rust 코드로 변환할 뿐, 안전한 lock api 사용을 위해 필요한 Rust lock api는 제대로 사용하지 못하는 모습을 보인다. 이러한 문제를 해결하기 위해 분석을 통해 *data-lock*과 *flow-lock*이 요약되어 있는 정보인 *lock summary*를 생성하고 해당 정보를 transformer에 넘겨주어 문법 뿐만 아니라 Rust의 lock api를 적절하게 사용한 Rust 코드를 생성하도록 한다. 이 연구에서 인상 깊었던 것은 *path-insensitive*한 분석을 사용했다는 점이었다. Rust 코드가 컴파일되기 위해서는 Rust 컴파일러의 타입체킹을 통과해야 하는데, 이 과정에서 사용되는 타입분석이 *path-insensitive*이기에 부정확한 분석을 수행해도 충분하다는 점을 이용한 분석이었다. 학부 컴파일러 수업 수준의 쉬운 분석으로 좋은 결과를 만들어낸 것이 인상 깊어 어떤 과정에서 이러한 해법이 나오게 되었는지 재민씨에게 질문드렸는데, 처음에는 비용이 비싼 기존 분석기를 사용하려고 시도하다가 문제가 요구하는 만큼(다시 말해, 컴파일 될 수 있을 만큼)의 분석이 그렇게 비싸지 않아도 된다는 것을 깨닫고 지금과 같은 형태로 분석을 디자인하셨다고 한다. 우리 연구실에서 강조되는 "자신만의 문제를 자신만의 스토리로" 해결한 연구라 굉장히 인상 깊었다.

Validating SMT Solvers via Skeleton Enumeration Empowered by Historical Bug-Triggering Inputs

현재 내가 연구하고 있는 분야인 smt solver fuzzing 연구이다. 이 논문이 사용한 아이디어는 regression test를 fuzzing으로 확장하는 것으로부터 시작한다. 많은 소프트웨어에서 버그가 발견되고 해당 버그가 고쳐지면, 해당 버그가 확실하게 고쳐졌는지/향후에 다른 fix가 해당 버그를 다시 촉발하지는 않을지 등을 판단하기 위해 버그를 촉발한 입력을 regression suite에 포함한다. 이 논문은 버그를 촉발한 입력으로부터 동일한 구조를 가진 다른 입력을 생성하여 버그에 대한 fix가 제대로 수행되었는지를 테스트하고자 한다. 저런 방식으로 얼마나 많은 버그를 찾을 수 있을까 생각했었는데 생각보다 굉장히 많은 버그를 찾아서 놀라웠다. 연구 초기에 교수님께서 이 논문의 아이디어와 비슷한 접근을 제시해 주신 적이 있는데, (지금도 귀담아 듣고 있지만) 앞으로는 교수님의 말씀을 더욱 새겨들어야겠다는 생각이 들었다. 이외에도 논문에서는 해당 방법론이 좋은 이유를 몇 가지 더 제시했는데 타당한 이유로 보이지는 않았다. 전반적으로 소프트웨어 엔지니어링 분야의 논문들은 이론적으로 정당화되거나 논리적으로 아주 정합적이지 않아도 직관적인 motivational example과 방법론을 통해 좋은 성능 보일 수 있으면 괜찮은 연구가 되는 듯하다. 여담으로 이 논문의 발표자인 maolin sun은 내가 버그를 리포트할 때 z3/cvc5 이슈란에서 많이 보았던 사람이었는데, 발표를 끝나고 말을 걸었을 때 나에게 '나도 너 알고 있어'라고 말하던 게 기억에 남는다.

DLIInfer: Deep Learning with Static Slicing for Python Type Inference

최근 활발하게 연구되고 있는 주제인 인공지능을 이용한 파이썬 타입 추론 연구이다. 이 연구의 초점은 딥러닝 application 분야에 속하는 대다수 연구가 그렇듯이 '어떻게 딥러닝 모델이 잘 학습할 수 있도록 데이터를 정제할 것인가'에 맞추어져 있다. 이 논문에서는 slicing이라는 특정 scope 안에 있는 변수의 값에 영향을 끼칠 수 있는 statement들을 추출하는 방법을 사용했다. 구체적으로 논문의 저자들은 PySlicer라는 데이터 흐름 분석(data flow analysis)를 기반으로 전역변수, 지역변수, 함수 파라미터를 추론할 수 있도록 하는 slicing tool을 구현하였다고 한다. 이후로는 적절한 vector representation, training을 수행한 뒤 기존 도구들에 비해 좋은 성능을 보였음을 보여주었다. slicing을 통해 타입 추론에 필요한 정보들을 잘 추출해낸 것이 좋은 설명이 된 듯하다. 인공지능을 사용하여 추론하는 연구이지만 성능의 핵심은 데이터를 잘 가공해서 정보를 추출하는 것에 있는 만큼 정적분석을 잘 아는 사람이라면 성능 개선에 비교적 쉽게 기여할 수 있는 분야인 것 같다는 생각이 들었다. 우리 연구실이 지향하는 연구와는 다소 거리가 있어보이지만 정적분석과 파이썬에 관심이 있는 사람이 연구를 처음 시작하기에는 나쁘지 않은 주제인 것 같다.

Sibyl: Improving Software Engineering Tools with SMT Selection

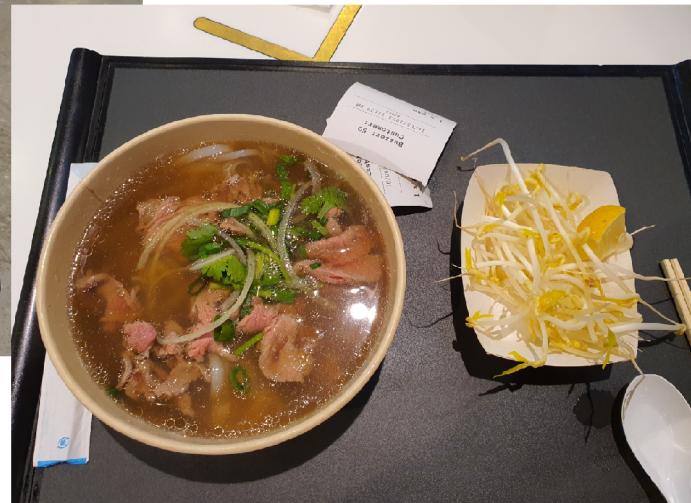
문제 정의가 재미있어 기억에 남는 연구이다. 많은 종류의 소프트웨어 검증 기술들(e.g. 기호실행, 정적분석 등)은 논리식을 생성하고 이를 smt solver를 통해 풀도록 설계되어 있다. 따라서 만들어진 smt query를 가장 빠르게 풀 수 있는 smt solver를 선택하는 것은 중요한 일이지만, 실제로 특정 논리식에 대해 optimal한 SMT solver를 찾는 것은 어렵다고 한다. 이러한 이유로 대다수의 기술들은 일반적으로 빠르다고 알려진 solver를 통해 풀고 있는데, 이러한 문제를 해결하기 위해 이 논문은 특정 smt query가 주어질 때 해당 query를 가장 빨리 풀 수 있는 smt solver를 찾는 것을 목적으로 한다. 기존 방법론들은 smt query의 의미(semantic)를 고려하지 않고 문법적 특징만을 고려해 잘못된 추론을 하는 경우가 있는데 논문에서는 해당 예제를 직접 보여주어 실제로 의미를 고려하는 것이 중요하다는 것을 잘 보여주었다. 의미를 고려한 추론을 수행하기 위해 이 논문은 graph neural network를 사용하여 representation learning을 수행하는 방법을 사용했다. 방법론 자체는 크게 흥미롭지 않았지만 재미있는 문제와 기존 방법론으로는 해결하기 힘들었던 motivating example이 인상적이었던 논문이었다.

멜버른

멜버른은 아름다운 자연과 웅장한 건축물을 동시에 가진 굉장히 매력적인 도시였다. 멜버른에 도착하자마자 멜버른 도심을 구경했는데 한국에서는 볼 수 없던 다양한 장면들이 새롭게 다가왔다. 고풍스러운 건물 양식, 도로 위로 다니는 트램(지하철은 이동하면서 바깥을 볼 수 없으니까!), 아름다운 야경과 미세먼지 하나 없는 맑은 공기들이 아직도 생생히 기억날 정도이다. 해외를 가본 경험이 거의 없어서 한 곳이라도 더 가보려고 열심히 길거리를 쏘아다녔다. 여행 경험이 많은 원석이가 사전에 멜버른에 대해 많이 조사해주어서, 짧은 시간 안에도 멜버른의 매력적인 명소들을 많이 볼 수 있었다.



- 빅토리아 도서관: '도서관이 구경할 게 있나?'라고 생각하는 사람도 이곳을 간다면 생각이 바뀔 거라고 장담한다. 이렇게 아름다운 도서관을 두 눈으로 볼 수 있다는 게 참 행복했다. 초록색 불빛이 은은하게 빛나면서 해리포터에나 나올 법한 분위기를 보여주고 있었다. 찍은 사진이 눈으로 보았던 그 분위기를 모두 담지 못해 아쉽다.
- 피츠로이 파크: 그냥 공원이지만 멜버른에서 가장 인상 깊게 남았던 장소다. 책에서나 볼 수 있었던 풍경이 눈 앞에 있어서 신기했다. 큰 나무들 사이로 맑은 햇빛이 들어오는데 모습이 장관이었다. 이 공원에서 웨딩 촬영을 하시는 분들도 계셨는데, 그럴만한 장소라는 생각이 들었다. 멜버른 사람들은 이 공원을 일상에서 누릴 거라는 생각에 굉장히 부러운 마음이 들기도 했다.
- 음식들: 학회 음식을 포함해서 멜버른의 음식들은 굉장히 맛있었다. 모든 식사 시간이 굉장히 즐거웠다. 호주에서 살았던 사촌동생이 추천해준 그릴드 버거, 유명한 카페인 둑스, 그리고 마이어 백화점의 일반 푸드코트에서 먹었던 짜국수까지 너무 맛있게 먹었다.



마무리

최고 수준의 학회에서 발표를 듣고 뛰어난 사람들과 연구에 관해 대화해볼 수 있어서 행복했다. 동시에 연구에 있어서도 좋은 동기부여가 되었는데, 나도 나만의 주제를 나만의 스토리로 풀어낸 연구를 사람들 앞에서 발표해보고 싶다는 생각이 들었다. 학회 뱅켓에서 중국인 친구가 "나는 이러한 내용으로 발표했는데 네 발표는 뭐였어?"라고 물어봤는데, 차마 답변할 수 없어 너무 아쉬웠다. 다음에는 꼭 내 발표가 무엇인지 알려줄 수 있도록 더 열심히 할 것이다. 마지막으로, 좋은 기회를 통해 값진 경험을 할 수 있도록 해주신 오학주 교수님께 감사드립니다