



FSE 2018

Florida, United States of America

Junhee Lee
Software Analysis Laboratory
Korea University

18.11.05 - 18.11.09

1 개요

미국 플로리다에서 열린 FSE(Foundation of Software Engineering)학회에서 발표를 하고 왔다. FSE는 탑 티어의 소프트웨어 엔지니어링 학회로, 주로 어떻게 하면 소프트웨어를 쉽게, 문제가 없고, 유용한 소프트웨어를 만들고 관리할지를 연구하는 분야이다. 이번 발표는 어떻게 하면 소프트웨어의 메모리 관리 오류를 자동으로 고칠 수 있을지에 대한 연구였다. 이번에 학회에서는 크게 두가지를 경험했다. 처음으로 학회 발표를 준비하고 발표한 것, 그리고 SE 학회 사람들과의 대화이다. 이 경험들을 공유하고자 한다.

2 학회 참석

FSE 학회는 ICSE와는 달리 규모가 많이 작았다. 총 300명정도가 참석하였고, 모든 행사는 학회 호텔안에서 이루어졌다. 발표는 총 4개의 세션으로 4개의 방에서 동시에 진행되었다. 발표를 안 듣고 돌아다닐 사람이 100명 정도라고 하면 평균 50명정도가 각 발표를 들었고, 그 중 인기있는 발표에 몰린다는 점을 고려하면 약 70, 60, 30, 20명정도의 구성으로 진행되었던 것 같다. 내 발표는 아쉽게도 인기 있는 뉴럴 네트워크와 정적 분석 두 세션과 동시에 진행되었고, performance라는 이상한 세션에 배치되었기 때문에 30명 정도 되었던 것 같다. 또, research paper 발표가 많게는 동시에 3개씩 진행되어 놓친 발표도 많아 아쉬웠다.

점심 시간은 테이블이 여러개 배치되어있고, 아무데나 들어가서 다른 사람들과 같이 이야기하게 되어있었다. 다른 사람들과 자연스럽게 이야기를 나누게 되는 점이 활발한 학회 활동에는 매우 바람직한 구조였지만, 밥을 먹는데 밥에 집중을 못하고 계속 귀를 쫑긋 세우고 무슨 말을 해야할지 생각을 하니 소화에는 좋지 않았다. 이번 학회는 나랑 성준이 2명이서 참가했기 때문에, 3일 내내 다른 누군가와 같이 대화를 하면서 밥을 먹었다. 이때 1:1 이라면 계속 끈질기게 물으면 뭐든 이해하면서 이야기가 가능하니까 괜찮을텐데 대개 4,8명이서 같이 테이블에 앉기 때문에 흐름을 따라가기 쉽지 않았다. 날이 갈수록 괜찮아지긴 했지만, 다른 주제를 미리 알아두고 이것저것 거리를 준비해두면, 더 재밌는 점심시간이 되었을 것 같다.

2.1 발표

내 발표가 있는 세션에 참가한 건 처음이었는데, 발표가 있으니까 새로운 것들이 많이 보였다. 발표장 사람 수가 처음으로 신경이 쓰였다. 내 발표장은 주제가 명확하지도 않은 performance 세션이라 사람수가 생각보다 많이 적었는데, 첫 발표 시작할때는 15명도 채 되지 않았던 것 같다. 그리고 세션 도중에 나가고 들어오는 사람도 처음으로 신경이 쓰였다. 내 발표는 4번째였기 때문에, 3번째 발표 중후반부터 들어오는 사람들이 정말 고마웠고 신기했다. 그 사람들은 정말 내 발표만을 듣기 위해서 세션에 들어온 것이기 때문이다. 다행히 그렇게 들어온 사람들이 조금 되서 발표때는 사람이 적당히 찼던 것 같다. 또, 발표 도중에는 나가는 사람이 거의 없었던 것 같고 핵심 부분 끝나고 정적 분석 이야기를 시작하니까 나가는 사람은 있었다.

발표는 성공적이었던 것 같다. 세션이 끝나서도 질문 하는 사람이 많았고, 좋은 기술, 그리고

좋은 발표라는 코멘트도 받았다. 기억에 남는 코멘트 중 하나는, ”MemFix가 scalable하지 않은 이유가 무엇이냐”는 질문에 대해 내가 방어적으로 대답을 하고 나서, 다른 사람이 질문자에게 ”저렇게 보장되는 패치면 trade-off가 당연히 있지 않겠느냐”라고 반박한 코멘트였다. SE학회로서 실용적이지 않은 기술에 대해 각박할 줄 알았는데, scalability에 대해 언급을 하는 사람이나 지적을 하는 사람이 session chair 한분밖에 없었다. 또, 다른 질문은 조금은 신기한 질문이었는데 ”패치를 개발자가 어떻게 믿을 수 있느냐”라는 질문이었다. 왜냐하면, 내가 계속 강조한 포인트가 MemFix가 만들어낸 패치는 sound and safe하기 때문에 새로운 오류도 발생시키지 않고 오류를 고치는 것이 보장되기 때문에 개발자가 다시 리뷰할 필요가 없다는 점이었다. 나는 패치의 readability에 대해서 이야기 하는 줄 알고 그에 대해 언급했는데, 질문자의 의도는 ”의심이 많은 개발자가 어떻게 툴이 만들어낸 패치를 믿을 수 있느냐”였다. 즉, 정말 sound하고 safe함을 개발자에게 어떻게 설득을 시킬 수 있느냐는 질문이었다. 나는 검증된 정적분석을 사용하면 당연히 믿을 수 있다고 생각하였는데, 질문자는 패치를 믿을 수 있는 이유를 그림으로 보여주라는 것이었다. 사실 도대체 왜 믿을 수 없는건지 이해가 되지 않아서 답을 잘 못하였는데, 주변 사람들이 저 사람 질문이 이러한 것을 말하는 것인데, 아까 보여준 그림과 같이 개발자한테 보여주면 되지 않겠느냐라고 친절히 다시 설명을 해줘서 다행히 잘 넘어갔다. 그 뒤로도 MemFix 분석이 만들어낸 constraint problem을 항상 풀 수 있는지에 대해 물어보기도 하고 이것저것 많이 물어봤는데, 걱정한 것과 달리 내 연구에 대해서는 신기하게도 잘 들리고 잘 말할 수 있었다.

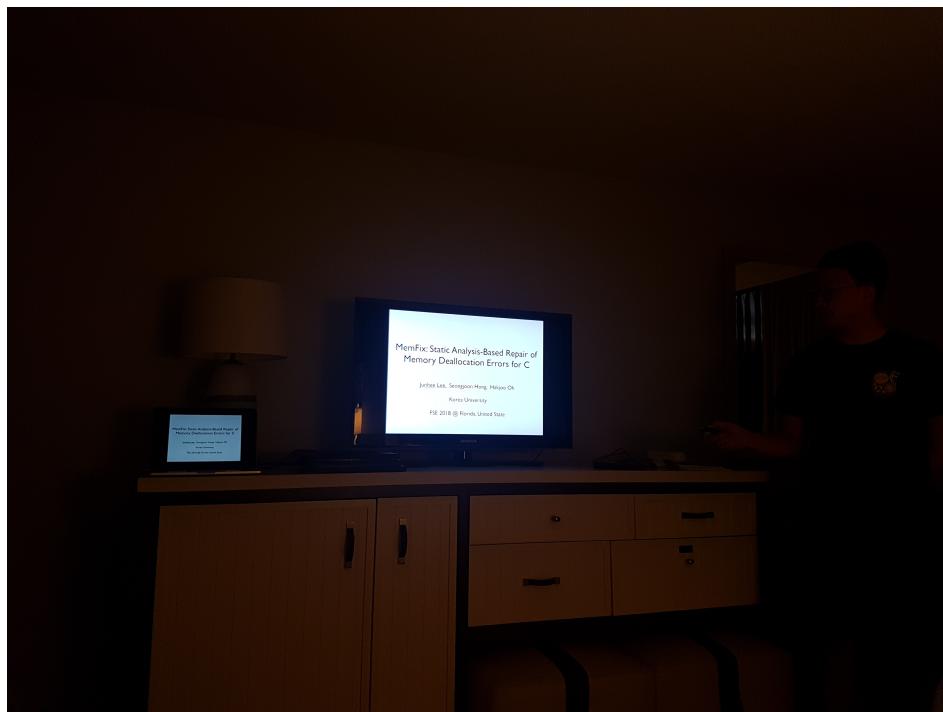


Figure 1: 숙소에서 발표 연습

발표는 정말 첫 날에 해서 다행이었다. 계속 중간에 머릿속에 텅 비어서 멈추지 않을까, 목소리가 떨려서 발음을 제대로 못하지 않을까 걱정을 많이하여서 연습을 계속 했다. 특히 학회에

도착해서는 긴장하고 머리가 맹한 상태에서도 발표를 잘 할 수 있도록 자기 전에도 하고, 아침에 막 일어나자마자 성준이가 시켜서 하기도 했다. 신기했던 점은 외국인들도 발표 연습을 꽤 많이 한다는 건데, 주변 외국인들과 이야기를 해보면 발표가 끝나서 편하겠다, 이따가 발표 연습하러 들어가야한다 와 같은 이야기가 많이 나왔다.

발표 준비 과정은 정말 많이 힘들었다. 시행 착오도 있었다. 초반에는 핵심이 문제를 푸는 직관과 알고리즘이라고 생각하였다. 왜냐하면, 논문의 핵심이 실험결과보다는 알고리즘에 초점이 있었기 때문이다. 그렇지만 20분 이내의 짧은 발표에서는 해당 알고리즘을 온전히 이해시키는 것도 힘들고, 무엇보다도 왜 이렇게 풀어야 하는지가 잘 설명되지 않았다. 따라서 발표는 딱 2 가지 포인트만 전달하려고 했다: 오류를 제대로 고치는 (sound and safe) 패치가 정말 필요하지만 어렵다는 것과, 메모리 해제 오류를 고치는 문제가 exact cover problem이라는 직관. 사실 정적 분석 디자인도 정말 재미있는 포인트가 많이 있는 곳인데, 질문을 듣고 사람들 이야기를 들어보니 정적 분석을 했다 정도만 이해한 것 같았다. 발표가 끝나고 신기하다는 표정을 지은 사람들이 몇몇 있었는데, 정적 분석을 듣고 그랬는지, exact cover problem을 보고 그런건지는 잘 모르겠다.

그래도 발표 준비를 하면서 내 연구의 동기나 핵심을 좀 더 잘 이해하게 되었다. 사실 논문을 내가 정말 동기부터 많이 고민해서 썼으면 이번 발표 준비가 좀 더 수월했을 것 같다. 이 연구는 단순히 오류가 많고 중요하기 때문에 하는 것이 아니다. 개발자가 오류를 고칠때는 오류가 정말 고쳐졌을지, 새로운 오류가 생기지 않을지를 검토하느라 상당한 시간이 소요되는데, 현재까지 이를 자동으로 하는 기술은 전무했기 때문에 한 것이고 그래서 더 의미가 있는 것이다. 다음 연구는 이러지 말자.

2.2 재미있는 논문들

Syntax-Guided Synthesis of Datalog Programs 이 논문은 이우석 교수님과 Xujie Si (공동저자)의 연구로 datalog program을 input relation로부터 합성하는 연구이다. 예를 들면, pointer의 alias 분석 같은 경우 pointer 간 Load, Store 등의 relation이 주어졌을때 어떤 alias 관계가 생성되는지를 명시해놓고 이를 만들어낼 수 있는 프로그램 분석 룰을 합성하는 것이 이 연구가 하고자 하는 일이다. 이때 search space가 매우 큰 것이 문제라고 지적하고, 이를 해결하기 위한 여러가지 technical idea를 제시하였는데 idea는 재밌는 pruning 기법들을 여러가지 제안하여 좋아보였다.

사실 잘 설득이 되지 않은 부분은 datalog로 작성하기 어려운 프로그램들은 input output 자체도 정확히 명시하기가 어려운 경우가 많아서 input output relation이 정확히 주어져있다는 것이 현실적인 세팅인가다. 예를 들어, 프로그램 분석의 경우 우리가 명시해야 하는 것은 특정 프로그램 포인트의 invariant이다. 특정 분석을 염두해두고 invariant를 손으로 직접 써보면 무언가를 빠뜨리게 되는 경우가 굉장히 많다. 분석기를 디자인할때도 기대한 결과랑 분석의 결과가 다른 경우가 매우 빈번하다.

Symbolic Execution with Existential Second-Order Constraints 이 연구의 저자인 Sergey Mechtaev는 아주용 교수님과 함께 symbolic execution constraint를 바탕으로 프로그램을 고치는 기술을 계속 연구해왔고 최근에 semantic program repair라는 박사 학위논문으로 졸업을 했다. 이 연구는 Sergey Mechtaev의 마지막 학회 논문이 아닐까 싶다. 이 기술은 테스팅을 기반으로 작동하기 때문에 테스팅 기술이 매우 중요할 수 밖에 없는데, 그래서인지 최근 논문의 동향을 보면 테스팅쪽으로 관심을 많이 가지는 것 같다. 이번 연구는 symbolic execution 이라고는 했지만 second-order constraint를 염두해두기 때문에 사실 이번 논문도 symbolic execution 기반의 program repair이다. 다만 문제를 symbolic execution의 관점으로 바라보려고 한 것 같다. 하나의 기술을 가지고 ICSE 3편에 FSE 1편을 쓰고 박사학위 논문을 마무리한 점이 인상깊었다.

이번 논문에서 다루고자 하는 문제는 다음과 같다. 프로그램을 특정 input-output 을 만들 어내도록 수정할때 고려해야하는 실행 경로가 만약 loop을 포함하면 무한한데, 이를 일반적인 기호 실행으로 하면 여전히 무한하여 path explosion 문제가 발생한다. 이때, 일반적인 symbolic execution을 사용하면 constraint가 input variable 기반으로 만들어져서 input variable의 갯수에 exponential 한 path condition이 발생하기 때문에, 이를 ESOC(existential second order constraints)기반의 symbolic execution을 통해 path condition을 함수로 요약하여 분석하여 그 갯수를 줄이고자 하는 것이다. 이를 위해 함수는 프로그램 합성을 통해 만들어내고, 이를 효과적으로 하기 위한 encoding도 제시하였다. Path condition을 함수로 요약하는 포인트가 재밌는 것 같은데, 나중에 제대로 읽어봐야겠다.

3 플로리다

이번 학회 참석이 첫 미국 여행이었다. 같은 서양이지만 스웨덴과는 많이 달랐다. 스웨덴 사람들은 너무하다 싶을 정도로 여유로워보이고 운동하는 사람도 많아서 비현실적인 느낌이 강했다면, 미국 사람들은 바빠 보이는 사람들도 많이 보이고, 비만인 사람도 많고 유쾌한 사람도 많아서 좀 더 친근한 느낌이 들었다.

플로리다 기온은 28~30도 정도로 시원하진 않지만, 셔츠 하나정도면 나름 쾌적하게 다닐 수 있는 좋은 날씨였다. 그렇지만, 학회 호텔 주변은 디즈니 공원중 하나인 디즈니 스프링스를 제외하고는 걸어서 갈 수 있는 곳이 하나도 없어서 여행은 거의 하지 못하였다. 근처에는 디즈니 테마 파크가 4개, 유니버설 스튜디오, 씨월드 등 놀이 공원이 많았는데 입장료가 최소 100에서 160달러정도 하고 놀이기구 하나를 타는데 1시간 정도 걸린다고 하여 포기하였다.

4 마무리

이번 학회 참석은 저번보다 의미있었다. 하나는 발표가 있어서 그랬고, 또 하나는 다른 사람들과 많이 이야기를 할 수 있어서 의미가 있었던 것 같다. 그렇지만, 주제랑 관점이 많이 달라서 공감하기 어려운 이야기가 많아서 아쉬웠다. 다음에는 PL 학회도 어떤지 참석해보고 싶다. 마지막으로 재밌는 좋은 연구를 할 수 있도록 지도해주시는 교수님께 감사드립니다.