



ICSE 2018

Gothenburg, Sweden

Junhee Lee
Software Analysis Laboratory
Korea University

18.05.27 - 18.06.03

1 개요

스웨덴 예테보리에서 열린 ICSE(International Conference on Software Engineering) 학회에 다녀왔다. ICSE 학회는 소프트웨어 엔지니어링 분야 학회중 가장 규모가 큰 학회로, 이번에는 1,650명 정도가 참석을 하였다. 이 글에서는 ICSE 학회를 참석하면서 경험한 것과 느낀 바를 공유하고자 한다.

2 ICSE 학회 참석

2.1 포스터 세션

학회 세션 중에서 포스터 세션이 아마 사람들과 가장 교류하기 좋은 장소가 아닐까 싶다. 포스터 발표 세션은 하루 종일 열려있지만, 주로 점심에 발표가 없는 때 사람들이 모이는데 정말 많은 사람들이 포스터를 구경하고 흥미로운 포스터에는 사람들과 열띤 토론을 하는 것 같았다. 발표자들은 너무 활발해서 포스터를 잠깐만 봐도 질문이 있냐고 물어왔는데, 한번은 내가 잠깐 포스터를 보고 있자 옆에서 밥을 먹고 있던 발표자가 뛰어와서 질문이 있냐고 물어왔었다. 그 당시에는 포스터를 보면서 이게 정말 필요한 연구일까 라고 생각을 하고 있어서 차마 그 질문을 하지 못하였다.

아쉬웠던 점은 포스터 발표를 미리 등록하지 못하여 구경만 하다가 마지막 발표날이 되어서 붙였는데, 사람들이 많이 있는 첫째날에 미리 구석에라도 붙였더라면 좋았을 것 같았다. 마지막날은 아무래도 사람들이 발표를 마치고 여행을 가거나 다시 귀국을 한 것 같아 사람이 많이 없었다.

2.2 발표 세션

ICSE는 발표도 매우 많아서 총 10개의 세션으로 나뉘었고, 그 중에 4 6개가 technical paper 세션이었다. 보통 비슷한 주제의 발표끼리 한 세션으로 묶어주었는데 그렇게 잘 묶어주는 것 같진 않았다. 그 때문에 세션을 이리저리 돌아다니면서 발표를 들었는데 발표간에는 쉬는 시간이 없기 때문에 발표를 앞 부분을 잘라 듣게 되었다. 정말 듣고 싶은 발표가 있는게 아니라면 그렇게 하지 않는게 나을 것 같다.

ICSE 발표들은 훌륭한 발표도 많았지만, 저게 도대체 무엇을 하고자 한 연구인지 알 수 없는 발표들도 많았다. 앞 부분에서 어떠한 문제를 풀고자 하는지를 명확히 전달하지 못하니깐 뒷 부분은 전혀 이해가 되지 않고 흥미도 가지 않았다. 또 반면에 문제를 매우 잘 설명해서 어떻게 풀었을까 기대를 했지만 어프로치에서 약간 김이 새는 발표들도 많았는데, 그래도 ICSE에 된 것을 보면 어찌됐든 문제를 잘 설정하고 어떻게든 문제를 푸는 것이 중요한 것 같았다.

수영이형 발표는 ICSE 발표들 중에서도 매우 좋았다. 핵심은 연구 동기와 아이디어를 듣는 사람들에게 확실히 공감을 시켰다는 점이고, 그 사이사이 유머 포인트가 사람들을 집중 시키는데 많은 도움이 되었다. 유머포인트 2개중 한개가 이상하게 먹히지 않았는데, 내 생각에는 수영이형 자신도 그 유머를 너무 많이하면서 식상하다고 느꼈기 때문에 그러지 않았을까 생각이 든다.

아쉬웠던 점은 역시 마지막날 발표에 마지막 세션이라는 점이었는데, 만약 첫째날이나 둘째날에 발표가 있었다면 듣는 사람도 많았을테고 반응도 좋았을 것 같다.

2.3 논문 소개

Context-Aware Patch Generation for Better Automated Program Repair 이 연구는 홍콩 과기대의 Ming Wen이라는 중국인의 연구인데, 발표가 매우 훌륭하여 인상에 남았다. 기술적으로 보자면 이 연구는 Automatic Program Repair의 합성 단계에서 그럴 듯한 패치를 합성할 부분의 주변(context)를 보고 우선적으로 하겠다는 것이다. 이렇게 하면 APR의 scalability를 높이는 것 뿐만 아니라 좀 더 그럴 듯한 패치를 우선적으로 생성함으로써 testcase에 overfitting된 패치를 덜 생성하게 된다는 장점이 있다. 이번에 이우석 박사님이 PLDI'18에서 발표한 Accelerating Search-Based Program Synthesis using Learned Probabilistic Models와도 프로그램 합성을 그럴듯한 프로그램을 먼저 합성한다는 점이 비슷한데, approach 측면에서 보자면 PLDI에 비하면 대충 한 느낌이 든다. 그렇지만 Automatic Program Repair의 한계를 잘 분석하고 이 한계를 극복하기 위해 search space를 어떻게 줄였는지, 그리고 이 space를 줄임으로써 scalability 뿐만 아니라 plausible한 패치 또한 많이 줄일 수 있음을 잘 주장하여서 완성도가 높은 연구임을 느낄 수 있었다. 또한 approach를 설명하기가 꽤 복잡했을텐데 이를 슬라이드를 매우 잘 쪼개고 말이 많음에도 불구하고 막히지 않고 정확히 전달 하는 것을 보고 감탄을 하지 않을 수 없었다.

UFO: Predictive Concurrency Use-After-Free Detection 이 연구는 concurrent 프로그램에서 발생하는 use-after-free 오류를 탐지하는 논문인데, 한 multi-thread execution의 trace들을 분석하여 concurrent use-after-free를 false positive 없이 예측하여 탐지한다는 것이 매우 놀라웠다. 우선 concurrent 프로그램에서 발생하는 use-after-free는 기존의 정적 분석으로는 탐지가 안되거나 많은 수의 false alarm 을 내게 되버리는 매우 어려운 종류의 오류이다. concurrent 하게 객체를 접근하여 해제를 하거나 사용을 하게 되면 한 스레드는 해제를 하고 다른 스레드가 사용을 하게 되면서 use-after-free를 일으킬 수 있는데 이러한 오류는 우선 사람이 코드를 보고 use-after-free가 발생할 시나리오를 생각하기가 무척 힘들고 이번 MemFix 연구를 하면서도 몇가지 케이스를 본 적이 있기 때문에 이러한 문제가 정말 중요하고 어려운 문제임이 체감이 되었다. 논문의 핵심 아이디어는 multi-thread 실행의 event(e.g., lock, free, use, etc.)를 모아서 use-after-free가 일어날 수 있는 다른 스케줄링의 실행이 있는지를 판단하는 것이다. concurrent 오류들은 대개 같은 인풋으로 실행하여도 다시 오류를 일으키기 쉽지 않은데, 이러한 방법을 통하면 한 인풋으로 일어날 수 있는 다른 스케줄링의 실행을 판단할 수 있기 때문에 효율적으로 use-after-free를 탐지할 수 있다.

DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars 이 연구는 Deep-Neural-Network로 작동하는 자동차를 테스트하는 연구인데, 학회장에 정말 덩러닝으로 움직이는 자동차가 있어서 깜짝 놀랐다. 덩러닝으로 움직이는 자동차는 테스트가 필수인

데, 왜냐하면 실제 상황에서 학습되지 않은 이미지를 인풋으로 받을 경우 어떻게 행동할 것인가가 매우 중요하기 때문이다. 그런데 이때 인풋이 이미지이기 때문에 최대한 다양한 상황에 대해서 테스트를 진행하려면 이미지를 합성해야 하고 이는 매우 어려운 일이다. 이러한 문제에 대하여 이 논문은 정말 간단한 해결책을 제시하였는데, 기존에 자동차를 테스트하던 이미지의 셋에 대해서 명암, 밝기 등만을 바꾸면 다양하면서도 실제 일어날만한 이미지를 만들 수 있다는 것이다. 게다가 명암, 밝기만을 바꾼 이미지에 대해서는 실제로 같은 동작을 수행해야하기 때문에 이를 통해서 해당 자동차를 더욱 학습시킬수도 있다.

3 마무리

많이 유익한 학회 참석이었다. 새롭게 알게 된 한국 연구자 분들도 많이 봤다. 또 다른 사람들이 어떤 연구를 하고 있는지, 어떠한 마인드로 연구를 하고 있는지를 많이 느낄 수 있는 시간이었다. 발표에 대한 생각도 새롭게 바뀌었는데, 발표는 발표자가 생각하는 그대로 나타나는 것 같다. 예를 들면, 발표자 자신도 잘 모르거나 애매하면 발표 자체를 알아 들을 수 없게 되는 것 같고, 발표자 자신이 재미있으면 듣는 사람도 재미있는 것 같다. 마지막으로 항상 좋은 연구를 하도록 지도해주시는 오학주 교수님께 감사드립니다.