

PLDI 2025 Report

Seoul, Korea



Seongmin Ko, PLRG@Korea Univ.

16-20 June 2025

Introduction

PL 분야의 최고 수준의 학회 중 하나인 Programming Language Design and Implementation (이하 PLDI)가 올해 서울에서 열리게 되었다. 학회에 논문이 붙어 발표를 하게 된 것은 아니지만, 서울에서 열린 덕분에 첫 국제 학회 참석을 PLDI로 할 수 있었다. 사실 이번 PLDI 이전 다른 해외 학회에 참석 할 기회가 00이었던 것은 아니었지만 아직은 내가 학회를 갈 짬이 될까 싶어서 갈 생각을 하지 않았다. 물론 1년 전의 내가 이번 PLDI를 참석했더라면 '이게 무슨 말이지?'만 반복했을지도 모른다. 하지만 첫 학회 참석이 주는 경험은 아주 값지기 때문에 기회가 된다면 무슨 수를 써서라도 꼭 참석해보길 바란다는 글을 남긴다. 또한 학회에 참석할 수 있도록 지원해주신 박지혁 교수님에게 감사의 말씀을 드리며, 앞으로 이야기할 학회에 참여하며 느낀 점들이 이 글을 읽는 사람들에게 도움이 되길 바란다.

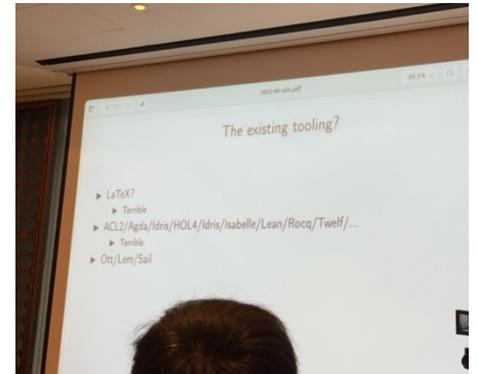
사실 지금의 나도 이번 학회에서 들은 논문 발표들 중, 발표만으로 온전히 이해한 경우는 거의 없는 수준이다. 그 이유를 돌이켜 보자면 첫째는 내가 원어인 수준의 영어 회화를 하지 못한다는 점과 둘째로는 해당 분야에 대해 잘 알지 못한다는 점이다. 예를 들어 발표자가 분위기를 풀어주려 농담을 해도 그 느낌을 제대로 이해하지 못해 혼자 웃지 못한 경험이 있다. 그래서 회화 연습을 좀 더 하기로 마음먹게 되었다. 그리고 해당 분야에 대한 이해도가 낮아 '무엇을 해결하고자 했는지'를 이해하기가 쉽지 않았다. 본 학회 첫날 abstract 부분만 읽고 관심있는 논문들을 선택해서 발표를 들었다가 머릿속에 물음표만 남는 참사가 발생했다. 그래서 다음 날부터는 Introduction을 보면서 해당 분야에 대해 찾아보고 발표를 들었는데, 확실히 왜 문제인지에 대한 이해가 잘 됐고 문제가 이해가 되니 해결방법도 훨씬 쉽게 이해가 되었다.

학회는 시청역 근처 소공동에 존재한 웨스틴 조선 호텔에서 16일 월요일부터 20일 금요일까지 5일에 걸쳐 진행되었다. 그 중 본 학회는 수요일부터 3일간 진행되고 월요일과 화요일은 워크샵과 튜토리얼들이 진행되었다. 대부분 오전 9시에 학회가 시작되며 오후 5시쯤 발표 세션들이 끝이 난다. 최대한 많이 얻어가려는 나의 몸부림 때문일까 학회가 끝이 나면 녹초가 되어서 집에 들어왔지만 정말 값진 시간이었다.

SOAP & RPLS

나는 워크샵으로 State Of the Art in Program Analysis (이하 SOAP)와 Real-World Programming Language Specification (이하 RPLS)를 신청하였다. 첫 날 진행한 SOAP는 Xavier 교수님의 키노트로 시작되었다. 전형적인 정적 분석에 대한 내용과 자료구조를 표현할 수 있는 abstract domain에 대한 강연이었다. 오후 세션의 시작을 알리는 Yulei Sui 교수님의 키노트는 deep neural networks verification에 대한 내용이었다. 가장 최근 연구 주제이신 incremental verification 부분은 제대로 이해하진 못했지만 robustness DNN에 대해 예전에 관심이 생겨 찾아본 적이 있어서 나름 흥미롭게 들을 수 있었다. 키노트들을 모두 흥미롭게 들었던 탓에 다른 발표들도 기대를 하고 들었지만 그렇게까지 흥미로운 발표는 없었다. 아무래도 첫 날이다 보니 처음 참석하는 거에 대한 긴장감이 남아 있어 이해력이 반토막이 나는 바람에 흥미가 돌지 않은 것일지도 모르겠다. 그래도 워크샵으로 학회의 분위기를 먼저 경험할 수 있어서 다행이었다.

화요일에는 RPLS를 참여하였는데, 박지혁 교수님께서 talk를 맡고 계시기도 하고 우리 연구실의 분야이기도 하니 specification에 대한 이해를 더 하고싶어 참석했다. 실제로 ECMA-262를 작성하는 Michael Ficarra의 이야기를 들으니 ESMeta가 그들에게 얼마나 도움이 되는지, 그들이 무엇을 필요로 하는지 좀 더 알게 되었다. 그리고 류석영 교수님께서 WebAssembly의 spec을 생성해주는 도구인 SpecTec에 대해 발표해주셨는데, WebAssembly는 애초부터 spec 자체를 mechanized language로 작성한다는 사실을 알게 되었다. ECMA-262는 자연어로 작성되어 있어 ESMeta를 이용해 mechanized language로 변환하여 분석 및 fuzzing등에 사용한다. 그래서 '애초에 mechanized spec으로 작성 후 자연어로 변환하자' vs '자연어로 작성할테니 알아서 mechanized spec으로 번역해라'의 구도가 자연스럽게 잡혀 연사 분들이 서로를 우아하게 설득하는 장면이 많이 나와서 재미있었다. 그리고 마지막으로 Peter Sewell 교수님께서 language spec을 작성하기 위해 필요한 도구와 그런 도구들이 가져야할 성질을 이야기해주셨는데, 모든 도구에 대한 모두짜기를 시전하셔서 발표를 듣는 내내 웃으면서 들을 수 있었다. 개인적으로는 학회 활동 중 RPLS를 제일 재미있게 참여한 것 같다.



All tools are terrible

Interesting Talks

Neurosymbolic Program Synthesis

금요일 오프닝 키노트로 진행된 Işıl Dillig 교수님의 neurosymbolic program은 PL분야에서 AI를 어떻게 사용해야할지 느끼게 해주는 강연이었다. Neurosymbolic program이란 neural networks와 symbolic reasoning을 적절히 섞어 서로의 단점을 보완해 문제를 해결하는 프로그램이다. Neural networks는 모든 것이 확률에 의해 선택되기에 reasoning하기 쉽지 않고 100%라는 것을 보장하기 어렵다. 이러한 성질은 PL에 관심을 가지는 사람들이라면 AI를 달갑게 여기지 않는 이유 중 하나일 것이다. 나 또한 AI를 연구에 사용하는 것이 달갑지 않았는데 발표를 듣고나니 AI를 쓰지 않겠다고 떼 쓸 때가 아니라 어떻게 잘 쓸지를 생각해볼 필요가 있다고 느꼈다. Işıl Dillig 교수님께서 크게 세가지 분야(Image editing, Data extraction, Robot learning)에 대해 이야기하셨는데, 그 중 data extraction 부분은 나의 연구분야와 같이 집중해서 들었다.

우리는 흔히 data extraction을 위해 regular expression (이하 regex)를 자주 사용한다. 하지만 regex는 syntactic matching만 가능하기 때문에 regex에 semantic한 정보를 넣어주기 위해서는 사용자가 해당 의미를 내포하는 syntactic한 표현을 regex로 잘 적어서 찾아야만 했다. 이러한 한계를 해결하기 위해 semantic regex (이하 sem-re)라는 것을 정의하여 regex가 semantic matching을 할 수 있도록 확장했고, 무려 예제 기반으로 sem-re를 자동으로 합성해주는 것까지 하셨다. 합성을 할 때 세부적인 부분에 대한 합성은 고전적인 top-down 프로그램 합성 기법을 사용하였고, 전체적인 틀을 잡는 부분은 LLM을 이용하여 합성하였다. Top-down enumerative 합성 방식이 가장 큰 문제인 search space의 증폭을 LLM을 이용하여 확 줄인 것인데 단순히 'template 만들어줘'에서 그치는 것이 아니라 단계적으로 template을 생성하고 fix해가며 합성하는, 그런 전체적인 구조를 되게 잘 구성했다고 느꼈다. 그리고 사용자가 추출하고자 하는 semantic한 분류를 type으로 생성하는데 이 과정에서도 LLM을 사용하였다. 생각해보니 LLM의 근본은 단어 집합에서 각 단어들의 의미적 유사도를 거리로 표현하는 점에 있는데, 그러한 본질적인 부분을 생각하니 LLM을 정말 적재적소에 잘 사용했다는 생각이 들었다.

Program Synthesis from Partial Traces

본 연구는 먼저 API 호출 로그로부터 API를 조합하는 스크립트를 자동 합성하기 위한 연구로 시작되었으며, 그러한 traces로부터 프로그램을 합성할 때 발생하는 partial traces 문제를 잘 해결한 연구였다. 예를 들어 어떤 웹 페이지를 개발한다고 생각해보자. 만약 닫기 버튼을 누른다면 사용자는 그저 화면이 닫아지는 행동만 경험하겠지만 실제로는 여러 단계의 작업들로 이루어져있다. 개발자는 한 행동을 하기 위해 매번 각각의 API를 실행해줘야 하는데, 이것 자동으로 묶어서 만들어주면 좋을텐데라는 생각이 자연스럽게 들게 된다. 단순히 이런 거 해주면 좋지 않을까?가 아니라 되게 직관적이면서도 세부적인 문제를 동기로 잘 선택했다고 생각이 들었다. 또한 연구를 하면서 마주하는 문제인 traces로부터 프로그램을 합성하는 경우 생기는 비가시적 함수에 대한 문제도 잘 정립하여 해결하였다. 최근 논문의 review로 motivation이 부족(왜 PEG를 써야하는가? 왜 자동 합성을 해야하는가?)하다는 이야기를 들었다. PLDI에 붙은 프로그램 합성 논문들의 motivation을 보고 다시 내 논문을 돌아보니 그냥 하면 좋지만 계속해서 주장하고 있었다. 문제가 없는게 문제였던 것이다. 문제를 해결하기 위해 사용할 방법론들과 도구들은 노력하면 얼마든지 개선할 수 있다고 생각한다. 하지만 문제는 내가 노력한다고 해서 좋은 문제가 되지는 않는다. 좋은 문제설정이 곧 좋은 연구결과를 낳는다는 것을 다시 한 번 느끼게 되었다.

예제 기반 합성을 연구하다 보니 사용자가 원하는 프로그램과 예제를 모두 만족하는 프로그램이 다르다는 것을 어떻게 해결해야 할지 고민이었다. 사용자가 원하는 것은 자연어기도 하고 정작 원하는 동작 들이 표현되지 않을 때가 있기도 한다. Trace 기반 API 자동 합성에서도 그런 문제가 발생하는데, 예제를 모두 만족하는 목표 프로그램을 합성한다고 해도 그게 사용자를 만족시켜주는 프로그램이 아닐 수 있다. 예제가 사용자의 의도를 모두 반영하지 못하고 예제에 overfitting된 경우가 생길 수 있는 것이다. 이러한 부분을 본 논문에서는 예제를 만족하는 프로그램들 중 하한을 정의하고 프로그램을 계속해서 rewrite하면서 사용자가 원하는 목표 프로그램에 최대한 가깝게 합성해준다. 이 부분은 연구의 주요 부분은 아니지만, 확실히 해결하고자 하는 문제가 뚜렷하다 보니 실제로 쓸만한 합성 기술을 만들고자 한 고민이 보이는 부분이었다.

Food



목요일 저녁은 뱅킷이 제공되고, 그 외에는 저녁이 제공되지 않았다. 첫날 점심을 제외하곤 뷔페식으로 점심이 제공됐고, 꽤나 맛있었다. 첫날엔 웨스틴 조선 호텔에서 판매하는 도시락을 점심으로 제공해 주셨는데 한국인인 나에겐 남기는 음식이 없을 정도로 괜찮은 식사였다. 하지만 음식에 대한 선택지가 달리 없으니 외국에서 오신 분들에게는 쉽지 않은 식사가 될 수도 있어 보였다. 그리고 가격을 듣고 나니.. 가격에 비해 너무 아쉬운 도시락이라는 생각이 들긴 했다. 그 외에 식사는 매우 맛있어서 양질의 단백질을 거의 무한으로 섭취할 수 있었는데 (점시의 절반은 육회, 절반은 닭다리살로 구성되었다.) 특히 디저트가 너무 맛있어서 정말 많이 먹었다. 덕분에 살크업이 제대로 되었지만 만족스러운 1주일이었다.