

PLDI 2025를 다녀와서

황도연

June 30, 2025

1 소개

6월 18일부터 20일까지 웨스틴호텔 조선에서 열린 PLDI 2025에 다녀왔다. PLDI는 PL 분야 최고 학회 중 하나로, 향후 기술이 축이 될 수 있는 새로운 이론들과 기존 PL 이론 기반으로 real-world 문제를 해결한 기술들을 모두 살펴볼 수 있는 좋은 기회였다. 참석으로 인해, 높은 수준의 발표와 재미있는 연구자들의 교류로부터 많은 것을 배운만큼 내가 배우고 느낀 것들이 동료들에게 전달이 잘 되었으면 하는 바람과 함께 본 후기를 작성한다.



Figure 1: 연구실 단체사진

2 인상깊은 논문들

Partial Evaluation, Whole-Program Compilation 이 논문은 기존 인터프리터에서 복잡한 JIT 컴파일러를 구현할 필요 없이 부분 평가(partial evaluation)

를 통해 AOT 컴파일러를 자동 도출하는 기법인 weval을 제시한다. Just-In-Time 컴파일러는 큰 속도 향상을 이루어내지만 복잡한 semantics를 가진 동적 언어를 최적화하는 작업을 수행해야 하므로 어려운 작업이다. 구체적으로 JavaScript의 obj.prop 같은 property access에서 obj의 shape이나 prop의 타입이 어떤 값이 될지를 정적으로 추론하는 것은 어려운 문제다. weval은 해당 문제를 해결하기 위해 동적으로 결정되는 부분(inline cache의 indirect call 등)은 interpreter가 계산해둔 값을 사용하고, 나머지 정적으로 분석 가능한 부분들(constant propagation, branch folding 등)은 정적으로 optimization을 수행하여 구현의 부담을 크게 낮췄다. "추론이 어려운 부분만 인터프리터로"라는 관찰이 멋진 뿐만 아니라, 실제 산업현장에서도 바로 사용될 수 있는 기술이라는 점이 매력적이었다.

Relaxing Alias Analysis: Exploring the Unexplored Space 기존 연구가 alias analysis의 정밀도를 향상시키는데 집중해온 것과 달리, 오히려 alias 정보를 의도적으로 "완화(relaxing)"시키는 반대 방향의 접근을 통해 컴파일러 최적화에 대한 기존 가정을 도전하는 연구다. 전통적으로 "더 정확한 alias 정보가 더 나은 최적화로 이어진다"는 가정이 지배적이었지만, 최근 연구들은 이 가정이 항상 성립하지 않음을 보여준다. 저자들은 (1) alias analysis를 완전히 제거해도 최종 바이너리에 미치는 영향이 작으며, (2) 전체 alias query 중 단 3%만이 실제로 영향을 미치고, (3) 무작위 relaxation이 특정 경우에 실행 시간을 21%까지 줄일 수 있다는 것을 보였다. 중요한 연구라는 생각이 들었고, 좋은 연구가 꼭 방법론을 제안해야만 되는 것은 아니라는 것을 다시 한 번 깨닫게 해준 논문이었다. 여담으로, 해당 논문은 발표 전에 연구자에게 직접 요청하여 읽어볼 정도로 큰 관심이 있었으나 발표는 생각보다 별로였다.

Program Synthesis From Partial Traces side effect가 있는 함수 호출만 기록된 부분적 trace로부터 완전한 프로그램을 합성하는 방법을 제안한 연구이다. 접근법은 크게 어렵지 않고, 초기에 모든 trace를 단순 재현하는 trivial solution부터 시작하여 점진적으로 규칙들을 적용해 일반화된 프로그램을 도출하는 방식이다. 실제 벤치마크에서 평가한 결과, API 로그나 시스템 콜 trace만으로도 유용한 실세계 프로그램을 생성할 수 있음을 보여줬다. 사실 논문만 봤을 때는 "이게 정말 쓸모가 있는 기술인가?" 싶었는데, 실제 발표에서 어떻게 이 연구를 시작하게 됐는지, 이 기술이 어떤 측면에서 유용한지를 매우 "생생하게" 보여주어 해당 의문을 잠재웠던 게 매우 인상적이었다.

Solving Floating-Point Constraints with Continuous Optimization 부동소수점 제약 조건(Floating-Point constraint) 해결에서 기존 SMT solver들의 한계를 극복하기 위해 연속 최적화(continuous optimization) 기법을 도입한 연구다. 기존 mathematical optimization 기반 solver들이 제약 조건을 불연속 함수로 변환하여 gradient-based method를 사용할 수 없었던 것과 달리, 저자들은 부동소수점 제약 조건을 연속적으로 미분 가능한 수학 함수로 변환하는 새로운 번역 규칙을 제안한다. 예를 들어 $a \leq b$ 같은 제약 조건을 $T(a \leq b) = (a \leq b) ? 0 : a - b$ 형태의 연속 함수로 변환하여, 전체 제약 조건의 만족 여부를 global minimum 찾기 문제로 바꾼다. 이를 통해 강력한 최적화 기법인 gradient descent를 적용할 수 있게 되는 것이 핵심. 매우 어려운 Floating-point solving을 신선한 방법으로

잘 풀었다고 생각해서 큰 기대를 가지고 발표를 들었으나 스크립트를 읽는 수준의 발표였다.

Guided Tensor Lifting 이 논문은 기존 레거시 코드를 고성능 텐서 DSL로 자동 변환하기 위해 LLM과 enumerative synthesis를 결합한 방법을 제시한다. 예를 들어 C로 작성된 `for(i) for(j) for(k) C[i][j] += A[i][k] * B[k][j]` 같은 행렬 곱셈 코드를 TACO DSL의 $C(i,j) = A(i,k) * B(k,j)$ 형태로 자동 변환하는 것이 목표다. 해당 기법은 LLM을 활용해 도메인 특화 휴리스틱을 probabilistic grammar 형태로 자동 학습한다. LLM이 제공한 10개의 가능한 해법을 바탕으로 확률적 문법을 구축하고, dimensional analysis와 bounded model checking을 통해 후보를 검증하는 체계적인 파이프라인을 구성한다. 이를 통해 수동 휴리스틱 설계 부담을 크게 줄이면서도 기존 방법들보다 우수한 lifting 성능을 달성했음을 보였다. 내가 좋아하는 도메인은 아니지만 발표는 연구에 관심을 가지게 만들 정도로 훌륭했다. 개인적으로 연구 발표를 어려워하는 후배들이 있다면 이 발표를 꼭 참고했으면 좋겠다.

Pointer Analysis for Database-Backed Applications 이 기술은 기존 자바 정적 분석 도구들이 데이터베이스 프레임워크와 Java/SQL 간의 복잡한 상호작용을 포괄적으로 처리하지 못했던 한계를 해결한다. 예를 들어 동적 SQL 생성에서 `userId` 값이 데이터베이스로 어떻게 흘러가는지 정적으로 추적하는 것이 핵심 과제며, 방법은 더 정교한 “sql 쿼리까지도 포섭하는” 포인터 분석이었다. 이 그룹은 자바 포인터 분석기인 tai-e를 기반으로 지속적으로 논문을 탐타어 학회에 발표하고 있는데, 요약 해석기의 real world 적용에 큰 관심이 있는 나로서는 고무적이었다. 우리 연구실에서도 포인터 분석의 대가와 파이썬 의미구조 전문가가 있는만큼 도전해볼 수 있는 분야라고 생각한다.

3 연구자들과의 교류

최고수준의 학회이므로 대단한 연구자들을 볼 기회가 많았다. z3 개발자인 Leonardo de Moura, 분석의 대가 Isil dillig 등 책에서나 보던 훌륭한 학자분들을 실제로 만나 뵈 수 있어 좋았다. 특히 이번엔 Partial Evaluation, Whole-Program Compilation을 발표한 Chris Fallin과는 실제로 대화를 나눌 수도 있었는데, “이번에 발표하신 논문을 봤고, 매우 인상 깊었다.”라고 이야기드리니 고마워하시며 사진까지 같이 찍어주셨다. 컴파일러에서 좋은 연구들을 하시는 걸로 아는데, 언젠가 다른 학회에서 내 발표를 보실 수 있었으면 한다.

dreal의 개발자인 공순호 박사님과도 대화할 기회가 있었는데, 현업에서 SMT solver가지는 난관들에 대해 구체적으로 질문드릴 수 있어 좋았다. 나는 SMT solver 연구자지만, 우리 연구실에서는 절대다수의 연구원들이 SMT solver를 black box로 쓰는만큼 SMT solver 개발자들이 실제로 필요하는 기술이 내가 개발하고 있는 기술과 괴리가 존재할 수 있다는 막연한 두려움이 있었는데, 공박사님과의 대화를 통해 현재 개발자들이 어떤 생각을 가지고 있고, 구체적으로 어떤 기술을 원하는지 파악할 수 있는 기회가 됐다.

굳이 대가들이 아니더라도, 중간의 휴식시간마다 발표자들과 대화시간들이 매우 유익했다. 예컨대, Guided Tensor Lifting을 발표한 Yixuan Li과 대화할 때 LLM-based approach가 꽤나 과학적으로 연구되고 있음을 알게 됐다. 현재까지

알고 있던 LLM-based approach 대다수는 프롬프팅 관련 기술이거나, 간단한 수준의 엔지니어링만을 수행했기에, LLM에게 일종의 읍소를 하는 기술이라고 평가절하하고 있었는데 잘못 생각하고 있었다는 걸 깨달았다. 합성 분야에서는 LLM 위에서의 이론을 제안하고 해당 이론 위에서 과학적으로 기술이 개발되고 있다는 게 신기했다. 우리 연구실은 전통적인 PL / SE 기술들에 정통한 사람들이 많으므로 우리 또한 특정 이론 위에서 보다 견고한 LLM-based 기술들을 시도해볼 수 있을 듯하다.

4 마무리

PLDI 2025 참석은 나에게 깊은 영감을 안겨준 여정이었다. 학회장에서 만난 연구들은 단순한 학술적 호기심을 넘어 실제 세계의 문제를 해결하고 있었다. 특히 정적분석기가 실제 애플리케이션에서 보안 취약점을 발견하는 모습을 보며, 연구가 가진 진정한 힘을 체감할 수 있었다.

이번 경험을 통해 나 역시 toy problem이 아닌 real-world impact를 가진 연구를 수행하고 싶다는 열망이 더욱 강해졌다. 단순히 논문을 쓰는 것을 넘어, 실제로 사용되고 다른 연구자들의 기반이 되는 의미 있는 기여를 하고 싶다.

마지막으로, 이러한 귀중한 기회를 제공해 주신 오학주 교수님께 진심으로 감사드립니다. 교수님의 배려가 없었다면 이런 소중한 경험과 깨달음을 얻을 수 없었을 것이다. 교수님, 감사합니다.