

# SPLASH 2024 Trip Report

Pasadena, California, United States

고려대학교 소프트웨어 분석 연구실

신지호

2024.10.20 ~ 2024.10.25

## 소개

SPLASH (Systems, Programming, Languages, and Applications: Software for Humanity)는 프로그래밍 언어와 소프트웨어 공학의 교차점에서 프로그래밍 언어 응용 분야의 최고 학회로 자리매김하고 있다. SPLASH 2024는 미국 캘리포니아 파사데나에서 개최되었으며 OOPSLA, Onward!, SAS, GPCE, SLE 등 여러 학회와 워크샵이 함께 열렸다. 강찬구 선배님과 이중훈 선배님의 논문이 OOPSLA 2024에 게재되었고 감사하게도 교수님께서 나에게 기회를 주셔서 학부생 신분으로 이번 학회에 참석할 수 있었다. 처음으로 경험한 학회의 기억을 기록하고 공유하고자 트립 레포트를 작성한다.

## 인상적인 발표들

### SAS Tutorial - A New Language for Expressive, Fast, Safe, and Analyzable Authorization

Cedar라는 authorization language를 소개하는 발표였다. Authorization은 다중 사용자 시스템에서 누가 무엇에 접근할 수 있는지를 결정하는 것으로, 모든 클라우드 기반 애플리케이션에서 중요한 문제이다. Request, entity data, policies가 입력으로 주어지면 Cedar는 10 마이크로초안에 allow/deny diagnostic를 내놓을 수 있다. Cedar의 policy analysis는 authorization use-cases를 나타내는 policy들을 logical formula로 symbolically compile함으로써 이루어진다. Cedar의 expression들이 symbolic term들로 reduce된다는 부분이 가장 흥미로웠는데, symbolic term들이 decidable fragment of smt가 될 수 있다고 한다. Expression이 바로 smt formula로 reduce될 수 있다는 것과, 그렇게 만든 formula가 decidable하다는 게 신기했다.

발표가 끝난 후 발표자 Emina Torlak에게 내 관심사에 대해 얘기하고 해당 분야의 연구를 시작하려는 사람에게 추천할 만한 리소스가 있는지 여쭤봤더니 이분께서 senior researcher의 특징이라 하시면서 웃으며 본인이 대학교수 시절 진행했던 수업의 자료를 추천해주셨다. 그리고 이분이 Rosette를 만드신 분이라는 사실을 알게되었다. Rosette와 Dafny가 어떻게 다른지에 대해서도 설명해주셨다. Dafny는 그 자체로 끝단에서 쓰이는 언어라면 Rosette는 solver aided language를 만들 수 있게 해주는 framework이다. 그리고 Rosette는 Dafny와 다르게 bound된 값들을 쓴다고 한다.

며칠 후 OOPSLA에서 같은 제목의 paper 발표를 다른 분께서 하셨다. 이분은 성함을 모르겠는데, 커피 브레이크 때 이분을 발견해서 다가가 Cedar expression을 어떻게 decidable한 formula로 바꿀 수 있었는지 여쭤봤다. 그랬더니 이분은 잘 모른다고 하시며 Emina Torlak 같은 SMT 전문가들이 고안하

는 것이기 때문에 이런 전문가들이 decidable하다고 하면 decidable한 것이라는 답을 주셨다. 참고할 만한 reduction principle 같은 것을 기대했는데 좀 실망했다...

## SAS Keynote - What's Still Missing in Static Analysis? A Decade-Long Journey.

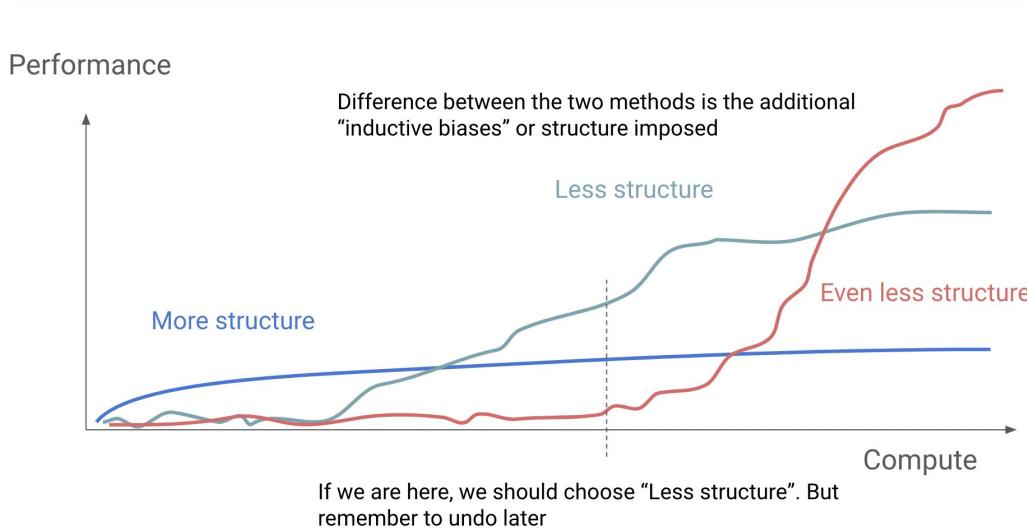


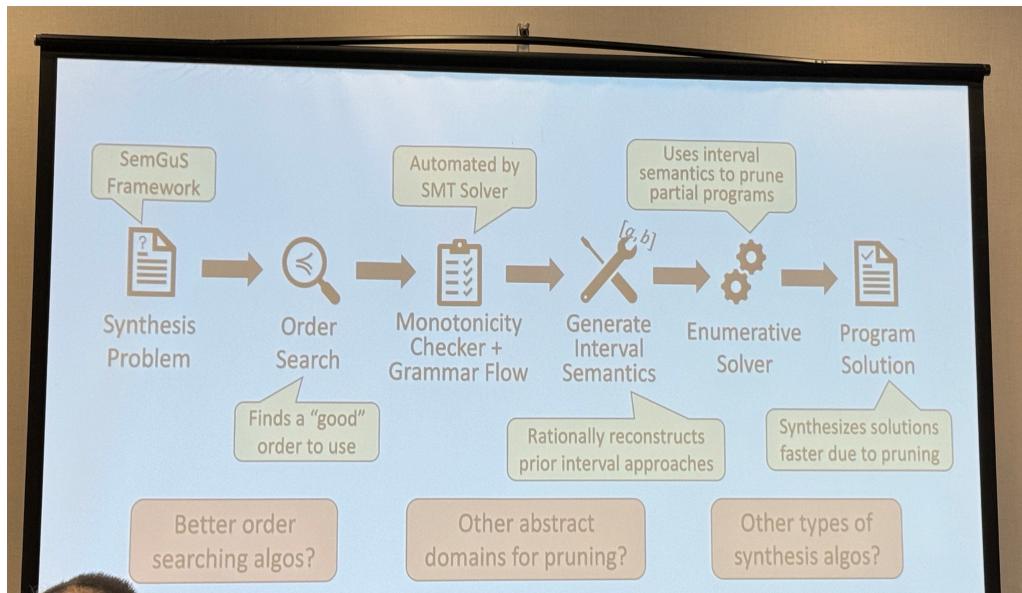
Image by Hyung won Chung, openai

Structure complexity에 따른 performance를 나타낸 그래프가 기억에 남는다. 간단한 structure가 대체로 더 scalable하지만, 더욱 간단한 structure의 경우에는 scale에 따라 performance가 달라지는 양상이 특이했다. 어떤 scale인지에 따라 다른 complexity의 structure를 사용해야 한다는 사실을 알아두면 좋을 것 같다.

발표의 주 내용은 Neurosymbolic approach에 대한 소개였다. 기존의 symbolic한 방식의 한계를 뛰어 넘기 위해 머신러닝을 활용하자는 것이다. 발표자가 static analysis에 lilm을 leverage한 IRIS를 만들며 얻은 insights를 공유했다. lilm은 자연어 정보를 활용하여 code context를 이해할 수 있으며 lilm을 이용하면 false negatives와 false positives를 filter out할 수 있다는 내용이었다. Neurosymbolic approach는 정적 분석 외에도 relational knowledge extraction과 combining foundation models 같은 분야들에 쓰일 수 있다고 한다. 이번 학회에서 무작위로 만난 사람들의 입에서 IRIS에 대한 말이 여러 번 나왔었다. 나는 neurosymbolic이라는 표현을 서칭하다 몇 번 보기만 했지 잘 모르는 분야였는데 IRIS가 핫하다는 건 분명하게 알 수 있었다.

## OOPSLA Paper - Automating Pruning in Top-Down Enumeration for Program Synthesis

### Problems with Monotonic Semantics

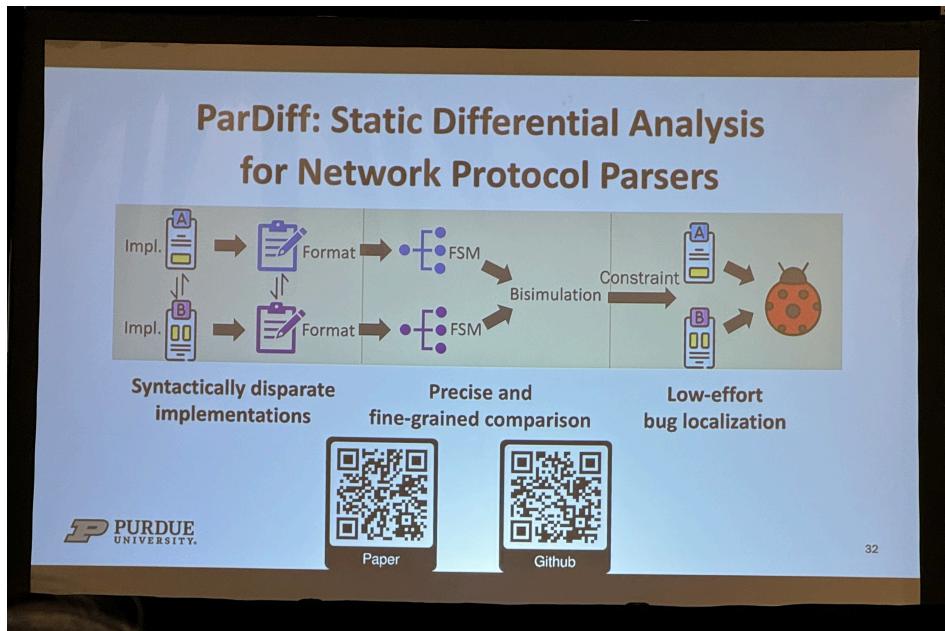


프로그램 합성을 위해 top-down enumeration을 할 때 abstraction-based pruning을 해왔던 기존의 연구들은 domain-specific한 abstract domain과 semantics를 일일히 만들어줘야 했다. 이에 대한 예시로 우리 연구실에서 publish되었던 imperative program 합성과 regular expression 합성 연구들이 제시되었다. 이 발표에서는 만약 합성하고자 하는 언어의 semantics가 monotonic한 성질이 있다면 concrete semantics로부터 바로 interval-based semantics를 얻어서 도메인에 대한 선행지식 없이 interval-based pruning이 가능하다는 것을 보였다. 임의의 합성 문제에 대해 abstraction-based pruning을 할 수 있는 조건을 선보였다는 점에서 의미가 있는 것 같다.

여담으로 포스터 세션 때 이 발표를 하신 분을 찾아가서 더 설명을 듣고 싶었지만 박지혁 교수님께서 이분을 계속 차지하고 계셔서 나는 하염없이 박지혁 교수님의 뒷모습을 바라보다가 자리를 떴다. 학회 포스터 세션에서 관심있는 연구에 대한 얘기를 듣고 싶다면 세션이 시작할 때 혹은 그보다 이른 시점에 발 빠르게 가야 한다는 교훈을 얻었다.

## OOPSLA Paper - ParDiff: Practical Static Differential Analysis of Network Protocol

### Parsers



네트워크 프로토콜이 여러 구현 스타일을 가질 때가 많고 이들 간의 semantic discrepancy가 버그를 나타낼 수 있다는 관찰에서 시작된 연구이다. 다양한 스타일로 작성된 구현들을 비교하는 어려운 문제를 다른 차원으로 가지고 가서 풀어내는 듯한 해결 방법이 인상적이었다. 두 개의 syntactically disparate programs의 차이를 구하기 위해 먼저 각 프로그램에 대해서 protocol format을 구한다. Smt solver를 사용해서 protocol format들이 구체적으로 어느 지점들에서 차이가 나는지를 알아내기는 쉽지 않다. 또 기본적으로 한 번에 한 개의 차이만 구할 수 있기 때문에 모든 차이를 한 번에 구하려면 smt solver의 구현 자체를 고쳐야 한다. 이를 피하기 위해 저자들은 format constraint를 finite state machine으로 변환하는 방법을 택한다. 그리고 세밀한 비교를 위해 fsm bisimulation 알고리즘을 이용한다. 큰 공식의 비교 문제를 여러 개의 작은 공식 비교 문제로 축소시키는 것이다. 네트워크 프로토콜에 관심이 없어서 처음에는 기대감 없이 들었는데 문제를 해결해나가는 흐름이 가장 재밌게 느껴졌던 발표였다.

## OOPSLA Paper - Monotone Procedure Summarization via Vector Addition Systems and Inductive Potentials

더 예측 가능한 프로그램 분석기를 만들고 싶다는 동기에서 비롯된 연구이다. 발표에서 기존 분석기의 한 계를 보여주는 예시를 통해 문제 의식을 잘 전달해서 이 연구의 필요성에 공감하며 들을 수 있었다. 이 연구에서는 예측 가능성을 높이기 위해 procedure의 요약본을 검증하는 방법을 제안한다. Vector

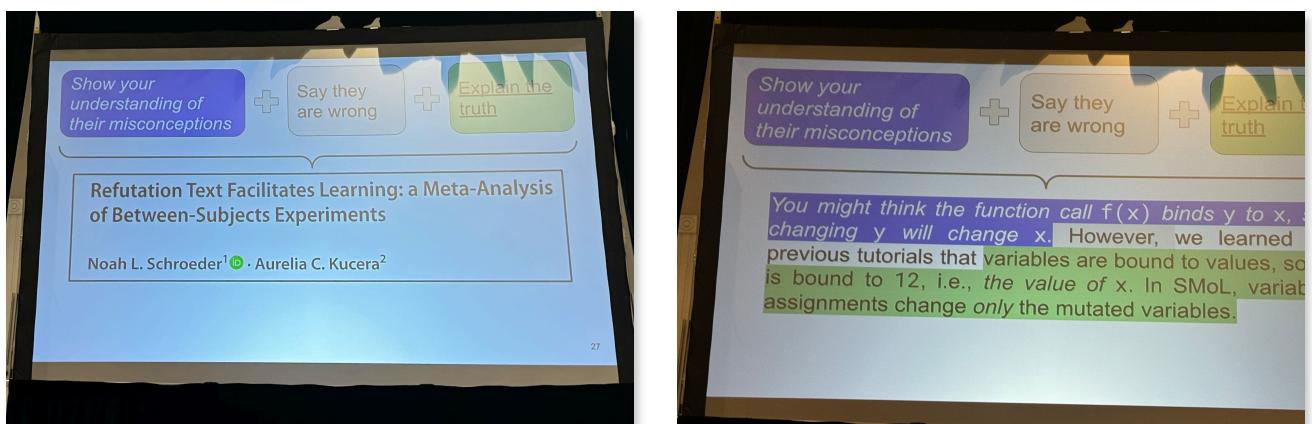
addition system with resets (VASR)을 이용해 best abstraction을 계산하고 이 VASR의 실행을 syntactic path의 context-free language에 따라 요약한다. 사실 요약 방법은 잘 이해되지 않았는데 발표자가 발표를 정말 잘했고 특히 초반부의 흡입력과 설득력이 기억에 남는다.

## OOPSLA Paper - Hyper: A Deductive Program Verifier for Hyper Hoare Logic

Hyperproperties는 sets of properties를 뜻하는 표현인데 프로그램을 여러번 실행했을 때 이 실행들을 연관지을 수 있는 성질을 나타낸다. Hyperproperty는 일반적인 correctness properties와 security properties를 표현하는 데 유용하다. 이 연구에서는 Hyper Hoare Logic에 대한 verification condition 자동 생성법을 보임으로써 동일 프로그램의 여러 실행에 대한 임의의 hyperproperties를 검증할 수 있는 기법을 제시한다. Hyperproperty라는 개념을 이 발표를 통해 처음 알게 되었는데 Hoare Logic을 확장시킨 후 proof rule를 만들고 이를 viper encoding으로 적용하는 과정을 보는 것이 흥미로웠다.

## OOPSLA Paper - Identifying and Correcting Programming Language Behavior

### Misconceptions



개인적으로 교육에 관심이 있어서 OOPSLA 마지막 날 마지막 시간대에 있었던 Education and Language Usability 세션을 가장 기대하고 있었다. 역시나 흥미로운 발표들을 여럿 들을 수 있었는데 그 중에서도 이 발표의 내용이 가장 재밌었다.

해당 발표는 학생이 작성한 프로그램이 어떤 misconception으로 인해서 잘못됐는지를 알려주는 interpreter에 대한 것이었다. 틀린 프로그램을 단순히 정답 코드로 repair해주는 것을 목표로 하는 것이 아니라, 왜 틀렸는지에 대해 설명해주고 그에 따라 어떻게 고치는 게 좋을지 방향성을 제안해주는 튜터링

시스템을 만들었다. 나는 틀린 코드를 자동으로 고쳐주는 방식이 숙련된 개발자에게는 좋을 지라도 프로그래밍 언어를 학습하는 학생의 입장에서는 그리 훌륭한 방식이 아니라고 생각해왔고 이 논문에서의 시도와 같이 교육학적인 고민이 담겨 있는 연구가 많이 이루어졌으면 좋겠다. 아쉬운 점이 있다면 아직은 각 코드마다 오직 하나의 misconception만을 리턴하는 제한이 있어서 복합적인 원인을 파악할 수 있는 시스템을 구축할 수 있다면 더 좋을 것 같다.

OOPSLA 첫 날에 Newcomer's coffee라는 이벤트가 있었는데, SPLASH에 처음 온 사람들끼리 만나 소통하는 자리였다. 상대적으로 자그마한 방에 20명 정도 되는 사람들이 예닐곱 명씩 한 테이블에 모여 오손도손 얘기를 나눴다. 이때 이 발표의 발표자를 만났었다. 혼자 내적 친밀감을 잔뜩 느끼며 발표를 들었다. 이분이 박사과정 5년차라고 하셨던 걸로 기억하는데 학회장에서 다른 사람들과 연구에 대한 얘기를 주고받는 태도를 보면서 무척 이상적이라고 생각했다.

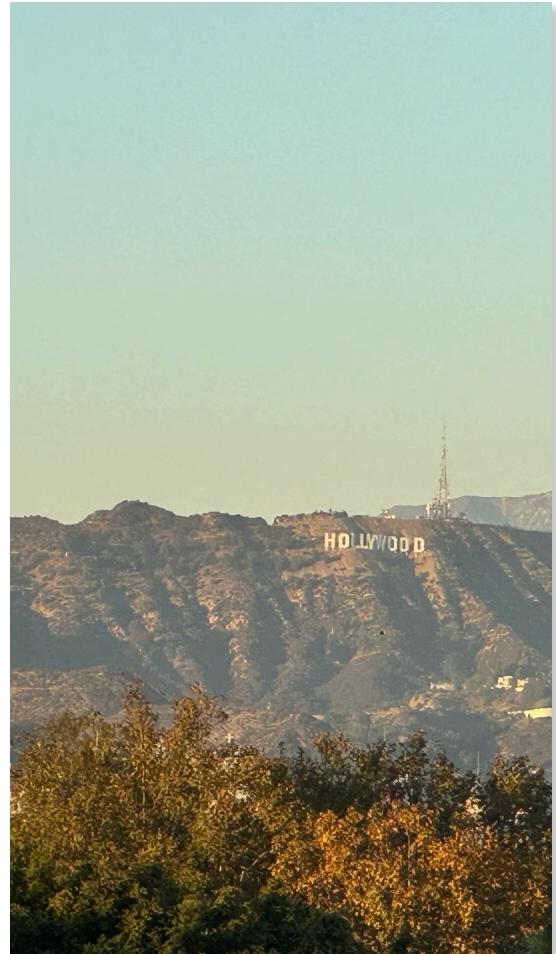
## 느낀 점

- 모니터 속 이름으로만 존재했던 이들이 내 앞에서 숨 쉬고 있는 게 신기했다.
- 발표가 잘 이해되지 않아서 어떤 발표는 외계어를 듣는 것 마냥 하나도 못 알아듣겠고 집중하기 힘들었다. 그럴 때 다른 곳에 가서 다른 유익한 경험을 할 수 있어서 좋았다. SAS 때 중간중간 옆 방에서 진행되는 PLMW 세션으로 피신해서 PhD Life Panel, Post-PhD Careers & Life Panel 등의 토크를 재밌게 들었고 Speed Mentoring Sessions 에도 참여했다. 학생들을 위한 자리가 준비되어 있다는 점이 좋았다.
- 학회장에 있는 많은 사람들이 들뜨고 신나하는 게 느껴졌다. 맨날 연구만 하다가 비슷한 관심사를 가진 사람들을 만나니까 좋아하는 것 같았다.
- 학술적 지식이 얇은 관계로 발표 내용을 이해하지 못하고 큰 흐름만 따라갈 수 있는 경우가 대부분이었다. 그래서 이번 보고서에는 학회에서 있었던 여담을 많이 늘어놓은 것 같은데 연차가 쌓이면서 연구 이야기의 비중이 커지길 바란다.

## 구경



LAX 도착 후 찾아간 산타모니카 해변



멀리서 본 할리우드 사인



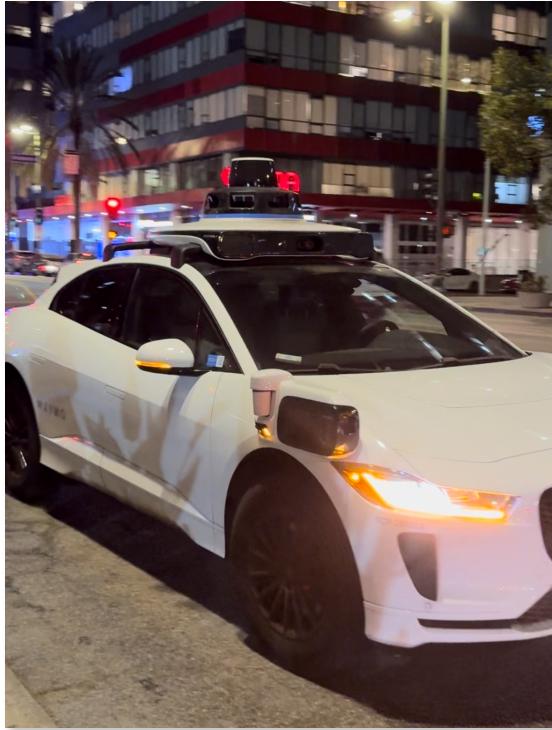
학회장에서 가까운 거리에 있는  
캘리포니아 공과대학교



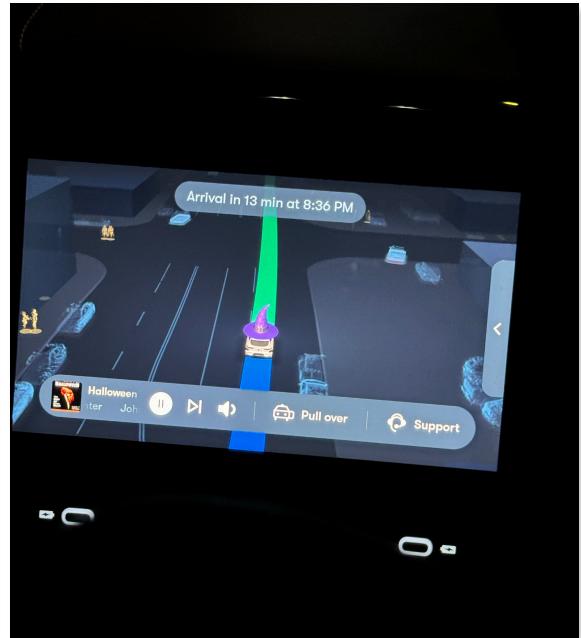
풍경이 멋졌다



팔자 좋아보이는 다람쥐도 봤다!



처음으로 자율주행자동차를 타봤다. 무서우면서도 재밌었다. 일단 나보다는 운전을 잘하므로 더 할 말이 없다.



안 그대로 잔뜩 움츠리고 있는 상태였는데 할로윈 기간이라고 음산한 분위기의 노래가 나왔다. 시승 경험을 한층 더 극적으로 느껴지게 해줬다.



마지막 날에 간 그리피스 천문대



천체 망원경으로 토성을 봤다! 고리가 있었다. 내년에는 고리를 볼 수 없다고 한다.

## 마치며

예상, 기대, 계획 등을 머릿속에 크게 그리지 않고 무작정 갔던 것 같은데 새로운 경험이 주는 자극을 제대로 받을 수 있는 시간이었다. 학회에 가서 뭘 알아들을 수는 있으려나 했는데 역시나 어려웠다. 그러나 사람들은 생각보다 친절했고 나는 생각보다 적극적이었다. 지금은 이름이 기억나지 않는 사람들과의 대화도 내게 생각할 거리를 많이 던져주었다. 내가 대학원생이 된다면 계속 연을 이어갈 수 있을 것 같은 사람들도 만났다. 열정적인 사람들이 잔뜩 있는 곳에 가서 그 속에 잠깐 있어보는 경험을 해볼 수 있어서 정말 좋았다. 좋은 연구로 학회에 갈 수 있는 길을 열어주신 연구실 선배님들과 내게도 갈 수 있는 기회를 주신 교수님께 깊은 감사의 마음을 전하며 글을 마친다.