POPL 2015를 다녀와서

강지훈

2015년 1월

POPL 2015를 다녀와서

초록

2015년 1월에 소프트웨어 무결점 연구센터의 지원을 받아 인도 뭄바이에서 열린 POPL(Principles of Programming Languages) 학회에 다녀왔다. CPP (Certified Programs and Proofs), VMCAI (Verification, Model Checking, and Abstract Interpretation), PLMW (Programming Languages Mentoring Workshop), CoqPL (Coq for Programming Language) 등 다양한 위성학회도 함께 열린, 우리 분야에서 가장 중요한 행사였다. 최근 연구 동향을 살펴본다는 목적도 있었지만, 가장 중요하게는 최근에 우리가 검증한 CompCert에서 따로따로 컴파일하고 링킹하기에 관한 논의를 하기 위해 학회에 참석했다. 비슷한 논문을 제출한 그룹 사람들과 논의도 하고, 우리가 검증한 방법이 왜 좋은지 설명하기도 했다. 그외에도 비슷한 분야를 연구하는 사람들과 대화를 많이 나누는 계기가 되었다. 이전에 갔던 학회들보다 더 충실한 시간을 보낼 수 있었던 것 같다.

감사 인사

2015년 1월에 소프트웨어 무결점 연구센터의 지원을 받아 인도 뭄바이에서 열린 POPL(Principles of Programming Languages) 학회에 다녀왔습니다. 학회에 다녀오기를 허락해주신 센터장 이광근 교수님과 허충길 지도교수님께 정말 감사드립니다. 연구비를 지원해주신 분들께 정말 감사드립니다. 저 개인적으로는 이번 학회 참석이 더 나은 연구자가 되는데 중요한 계기가 되었던 것 같습니다. 앞으로도 계속 정진하여 열심히 하겠습니다. 감사합니다.

인도 뭄바이

POPL이 열린 인도 뭄바이는 활기찬 도시였다. 사실 뭄바이는 빈부격차가 매우 커서 길거리에서 사는 사람도 많고 어딘가 정돈되지 않은 도시라는 느낌이 많이 들어서 처음에는 오래 있고싶다는 생각이 들지 않았었다. 심지어 길에서 생활하는 어린 아이들도 많이 찾아볼 수 있었다. 영어를 할 줄 아는 인구가 무척 적어서 의사소통이 어렵기도 했다. 그러나 시간이 지나면서 길에서 만나는 사람들의 눈에서 생기를 찾아볼 수 있다는 것을 알게 되었다. 겉보기보다 훨씬 활기차고 강한 도시였다.
처음 보는 것들도 많았다. 사람들이 크리켓을 하는 장면을 난생 처음으로 구경하기도 했다. 호텔에서 학회장으로 이동하는 차 안에서 봤는데, 매일 같은 장소에서 크리켓을

하는 사람들을 만날 수 있었다. 역시 차로 이동하면서 학교, 철도공사, 관공서, 군부대 등 다양한 건물도 봤다. 긴 역사를 가진 국가답게 역사책에서 볼 수 있을만한 건물들을 많이 볼 수 있었다.
학회가 열린 TIFR(Tata Institute for Fundamental Research)는 인도 과학/공학의 요람과 같은 곳이라고 한다. 인도 최초의 컴퓨터를 설계하고 조립한 장소라고 한다. 그외에도 인도의 수많은 업적이 이곳에서 나왔다고 한다. TIFR은 인도 서쪽 바다와 맞닿아있는데, 멋지게 펼쳐져 있는 해경을 보노라면 연구를 더 잘할 수 있을 것 같다는 생각이 들 정도로 감동적인 전망이 보이기도 했다.
진짜 인도 카레도 많이 먹었다. 뭄바이에 체류하는 동안의 거의 매 끼니마다 인도 카레를 먹었다. 내가 알던 한국의 인도 카레는 인도 카레가 아니라 "한국식" 인도 카레였다는 것을 깨달았다. 진짜 인도 카레는 종류가 무척 다양했고, 향과 맛이 매우 강했다. 카레를 너무 많이 먹어서 배탈이 나기도 했었다. 그래서 다시 한국으로 돌아오기 직전에는 배탈이 많이 나서 (그나마 내가 많이 먹어본 음식인) Domino Pizza를 애용하기도 했다.
뭄바이가 좀더 안전한 도시였으면 좋겠다고 생각했다. 차가 보행자들을 배려해주지 않고, 운전자들이 거울 없이 운전하기도 했다. 전철은 문을 열어놓고 주행하기도 한다. 안전은 비용이지만, 투자할만한/해야만 하는 비용이라는 생각이 들었다.
학회 기간 도중 외교부에서 짐슈/카슈미르 지역에 있는 사람들은 대피하라는 문자가 왔었다. 뭄바이는 비교적 치안이 잘 유지되고 있는 도시지만, 국경 지역은 그렇지만도 않은 것 같았다. 외교부 문자를 보고 이 지역에 평화가 있기를 빌었다.

학회 발표 경청

How to give a good research talk (Stephanie Weirich, PLMW)
좋은 발표란 무엇인가에 대한 발표이다. 내용을 요약하면 포인트는: 논문과 발표는 다르다! 효과적인 논문을 쓰는 것과 효과적인 발표를 하는 것은 전혀 다른 일이며, 효과적인 발표를 하는 것은 광고에 가깝다는 이야기를 했다. 왜 내가 한 연구가 재밌는지, 중요한지, 그리고 당신이 논문을 읽어봐야 하는지를 잘 설명하는 것이 좋은 발표라고 한다.
사실 이런 내용들은 그동안 많이 들었지만, 이렇게 정리된 톡으로 들으니 핵심 정리가 된 느낌이고 무엇보다도 기분이 좋았다. 내가 어렴풋이 알던 내용을 깔끔하게 정리해 주었기 때문이다. 슬라이드(http://www.cis.upenn.edu/~sweirich/talks/plmw15-giving-a-talk.pdf)의 일독을 권한다.

A Scalable, Correct Time-Stamped Stack (Mike Dodds et al, POPL)
이 논문은 동시성을 지원하는 효율적인 스택을 새로 개발하고 이를 검증한 일이다. 동시성을 지원하는 효율적인 자료구조를 구현하기란 매우 어렵다고 알려져 있는데, 왜냐하면 효율적이기 위해서는 스택에 관한 각 함수가 굉장히 복잡한 제약조건(invariant)를

만족해야 하기 때문이다. 이 논문을 timestamp를 이용한 비교적 간단한 제약조건을 이용하여 매우 효과적인 스택을 구현하고 이를 검증했다. 벤치마크에서 기존에 구현된 동시성 스택보다 나은 성능을 보였다.
나는 동시성 프로그래밍에 굉장히 관심이 많아서, 이 논문 발표를 주의깊게 들었다. 이런 일도 POPL에 게재되는구나 싶은 생각도 들었다. Queue 등 다른 자료구조에도 확장한 결과를 어서 볼 수 있게되면 좋겠다.

Programming up to Congruence (Vilhelm Sjöberg et al. POPL)
Dependent type을 가볍게 적용하는 방법에 대해 다루었다. 논문에서 소개한 Zombie 언어는 Coq같이 본격적인 dependent type을 사용하지는 않지만 대신 효과적인 타입 추론이 가능한 지점에 대해 잘 포착했다고 생각한다. Dependent type을 써서 프로그래밍을 한다면 볼만한 논문이다.

Deep Specifications and Certified Abstraction Layers (Gu et al. POPL)
OS를 검증한 논문이다. OS를 검증하기 위해 (1) OS를 modular하게 나누고, (2) 각 부분이 만족해야 할 "deep" specification을 정하고 검증했다고 한다. Deep specification이란 (종료 여부를 포함하여) 프로그램의 행동을 완벽하게 기술하는 스펙을 말한다. 보통 사람들이 생각했던 스펙은 종료 여부, 타입 안전성 등 프로그램의 행동을 일부만 기술했었다. 하지만 이런 약한 스펙을 사용하는 경우 증명해야 할 성질이 늘어날 경우 증명 전체를 바꿔야 하는 단점이 있다고 한다. 강한 deep specification을 이용하여 검증하였을 때 1년 내에 Linux를 guest로 돌릴 수 있는 하이퍼바이저를 검증할 수 있었다고 한다.
우리가 최근에 해결한 CompCert에 링커를 다는 문제와도 연관이 깊고, 우리도 궁극적으로 컴파일러를 사용하는 시나리오를 포함한 소프트웨어 검증 일반에 관심이 있기 때문에 이 논문을 주의깊게 들었다. 나는 개인적으로 이 논문에서 스펙을 표현하는 방식에 불만이 있는데, 이 불만을 보다 체계적으로 발전시켜 정리하고 싶다. 이런 고민들이 추후에 우리가 항공기와 같은 소프트웨어를 검증하는데 도움이 되리라 생각한다.

Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning (Jung et al. POPL)
동시성 프로그램을 검증하는 방법에 대한 논문이다. 논문도 한 번 읽었지만 내용을 완전히 이해할 수는 없었다. 이 문제 자체가 매우 관심있는 문제이기도 하고, 내가 관심을 가지고 있는 다른 문제: 동시성 프로그램을 컴파일하는 컴파일러를 검증하는 방법과도 깊은 관련이 있으리라 생각한다. 꼭 주의깊게 다시 읽고 싶은 논문이다.

A Calculus for Relaxed Memory (Crary et al. POPL)
현대의 C/C++ 표준은 메모리가 보장해야 할 조건을 간접적으로 기술하는 반면, 이 논문은 조건을 직접적으로 기술하는 모델을 제시한다. C/C++ 프로그램이 동시성 환경에서 어떻게 동작할지 논증하는 것은 매우 어렵다고 알려져 있다. 이 논문에서 제안한

방법을 사용하면 논증이 상당히 쉬워질 수 있으리라 기대한다. 다만 표준이 이 논문과 같이 변경될 수 있을지는 의문이 있다. 좀 더 추세를 지켜보고 싶다.

Common compiler optimisations are invalid in the C11 memory model and what we can do about it (Vafeiadis et al. POPL)
사실 다른 사람들과 이야기하느라 이 톡은 듣지 못했는데, 페이퍼를 조금 읽었다. 동시성 메모리 모델을 생각하면 사람들이 다 옳다고 생각했던 최적화가 틀렸다는 내용이다. 생각하면 생각할수록 정말 동시성 메모리 모델은 요물이라는 생각이 든다.

Summary-Based Context-Sensitive Data-Dependence Analysis in Presence of Callbacks (Tang et al. POPL)
예전에 JavaScript 분석하면서 만났던 문제가 바로 이 문제이다: 함수별로 bottom-up 방식으로 분석하면서도, callback 함수를 잘 분석하는 방법이 무엇일까? 내 잠정적인 결론은 없다는 것이었는데, 이 논문은 이 방향으로 일보 전진한 결과를 선보인다. 일반적인 값분석은 아니지만 데이타 의존성 분석(taint 분석과 같은)에 대해서는 앞서 말한 문제가 어느정도 해결 가능함을 보였다. 하지만 일반적인 값분석으로 확장하기에는 큰 허들이 있는 것으로 보인다. 앞으로 이 그룹에서 어떤식으로 이 주제를 탐구해나갈지 궁금하다.

A Coalgebraic Decision Procedure for NetKAT (Foster et al. POPL)
KAT(Kleene Algebra with Tests)이란 network를 기술하고 검증하는데 널리 사용되고 있는 논리 기반인데, 이 논문은 이 기반에 대한 decision procedure를 만들었다. 이 decision procedure를 사용하면, 예를들어 주어진 network가 loop을 가지고 있지 않음을 판단할 수 있다.
작년부터 KAT을 이용해서 network를 기술하고 검증하는 일에 대해서 점점 더 많이 듣곤 했는데, 참 재미있는 주제인 것 같다. 많은 학생들이 교과서에서나 볼 오토마타이론이 이렇게 현실적으로 중요한 문제에 사용될 수 있다는 것이 무척 고무적이라고 생각한다. 좋은 theory가 가장 practical하다는 교훈을 다시 한 번 느낄 수 있는 톡이었다.

CompCert로 따로따로 컴파일하고 링킹하기

현재 CompCert는 전체 프로그램을 한꺼번에 컴파일한 경우만 실행의미가 보존된다는 것이 증명되어 있다. 하지만 거의 대부분의 개발환경에서 컴파일러는 프로그램의 일부(파일 단위로)만 컴파일하고, 나중에 컴파일된 결과(목적파일)을 합치는(링킹) 과정을 거치게 된다.
우리는 이 갭을 메우기 위해 CompCert링커를 개발하고 검증했다.
우리와 비슷한 일을 한 그룹이 둘 있다. Princeton의 Andrew Appel 그룹에서 한 일은 이번에 POPL에 게재되었고, Yale의 Zhong Shao 그룹에서 한 일은 이번에 CPP에 게재되었다. 우리는 학회에 가기 직전에 막 증명을 끝낸 참이었다.

각 그룹이 한 일의 장점을 써보면 다음과 같다.
(1) Yale이 한 일이 Princeton이 한 일보다 더 강한 성질을 증명했다. 구체적으로 두 개의 프로그램을 링킹한 것이 무엇이냐는 질문에 대해 Princeton보다 현실적으로 답변했다. Yale의 경우 두 프로그램의 링킹은 프로그램 syntax를 합한 결과이다. 반면 Princeton은 두 개의 프로그램을 syntax 수준에서 링킹하지 못하고 semantics 수준에서만 링킹할 수 있다. 즉 두 개의 프로그램을 링킹한 것의 semantics는 각 프로그램의 semantics를 합한 결과이다.
(2) Princeton이 Yale보다는 더 많은 패스에 대해서 증명했다: CompCert 2.1 패스중 70% 가까이 증명했다.
(3) 우리는 Yale, Princeton에 비해서 더 적은 노력을 들이고도 CompCert 2.4의 전체 패스에 대해 증명할 수 있었고, 무엇보다도 CompCert의 원래 semantics를 바꾸지 않았다. Princeton과 Yale에서는 증명을 위해 CompCert가 사용한 언어들의 semantics를 바꿨다.

우리가 증명에 들이는 노력을 획기적으로 줄일 수 있던 까닭은 우리가 링킹이라는 문제의 핵심을 파악했기 때문이라고 본다. 기존의 CompCert 증명이 다른 모듈이 없을 때 함수가 올바르게 컴파일된다는 것이라면, 우리가 해야 할 증명은 다른 모듈이 있을 때에도 함수가 올바르게 컴파일된다는 것을 보이는 것이다. 비유하자면 이는 “strong induction”을 이용한 증명 과정과 닮았다. 다른 그룹은 이게 문제의 핵심이라는 것을 잘 이해하고 증명한 것 같지 않다.
참고로 링킹을 정의하려면 필연적으로 다른 모듈에 있는(그래서 현재 내가 정의를 가지고 있지 않은) 함수를 호출했을 때 어떤 일이 일어나는지를 기술해야 하는데, 다른 두 그룹에서는 이 경우 외부 함수를 호출한다는 특별한 이벤트가 발생하도록 semantics를 정의했다. 하지만 우리는 이를 semantics를 바꾸지 않고도 처리했다.

우리가 한 작업도 단점은 있다. 아직 CompCert로 컴파일한 결과만 링킹할 수 있다는 점이다. 하지만 우리는 일을 좀 더 해서 쉽게 이 단점을 극복할 수 있으리라 기대한다. 이렇게 일을 더 해서 기존에 Princeton과 Yale이 한 일보다 강한 결과를 적은 비용으로 증명하려 한다. 이를 다음 POPL 2016에 제출할 예정이다.

학회에서 Princeton, Yale 그룹 사람들 그리고 CompCert를 만든 Xavier Leroy와 많은 이야기를 나눌 기회가 있었다. 학회에 가기 전에 Coq Club이란 메일링리스트에 우리가 이런 일을 했다는 것을 공개했었는데, Princeton 그룹에서 POPL의 게재한 논문의 1저자가 우리와 보고싶다고 먼저 연락을 해왔다. Xavier로 이메일로 큰 관심을 보였었고, POPL에서 보기로 약속했었다.

만나서는 (1) 왜 우리가 쉽게 증명을 할 수 있었는지 그 노하우를 전하는 시간을 가졌다. 그리고 (2) Princeton과 Yale 그룹이 증명한 성질이 어떤 점에서 약한지 논했고, (3) 우리가 어떻게 그 단점을 보완할 수 있는지 얘기했다. 주요한 논쟁이 벌어질 수 있다기보다

는 거의 허충길 교수님이 우리 그룹이 했던 일을 소개하는 시간이었던 것으로 기억한다. 두 그룹의 사람들 모두 우리가 한 일에 큰 관심을 보였고, 논문으로 정리되면 꼭 읽고싶다고 했다.

내가 한 일에 다른 사람이 관심을 갖고 얘기를 듣기를 원한다는건 참 기분좋은 경험인 것 같다. 여태까지는 이런 일이 없었고 학회에서 아무도 나에 대해 관심을 가졌던 사람이 없었는데, 이번 학회에서는 나를 알아봐주고 찾아서 말을 걸어주는 사람이 (적었지만) 있었다. 이게 내 자신감에 상당히 긍정적인 영향을 주는 것 같다. 그와 동시에 좋은 연구를 많이 해서 학회에 가야겠다는 생각을 하게 되었다. 이 일을 잘 마무리해서 또 좋은 학회에 게재할 수 있게되면 좋겠다고 생각했다.

사람들과 만나기

비슷한 관심을 가진 사람들을 많이 만날 수 있었다. USTC(University of Science and Technology in China)의 포닥인 밍 푸라는 사람과 많은 이야기를 나웠는데, 푸도 소프트웨어 검증을 하고 있고 대상으로 하는 소프트웨어도 컴파일러와 비슷한 면이 있어서 매우 비슷한 검증 방법(refinement)을 사용하는 것을 알 수 있었다. 그래서 우리는 저녁을 먹으면서 소프트웨어 검증의 어떤 측면이 뭐가 어렵고 귀찮고 힘든지에 대해 깊은 이야기를 나눌 수 있었다.
PLMW에서 인상깊은 톡(How to give a good research talk, http://www.cis.upenn.edu/~sweirich/talks/plmw15-giving-a-talk.pdf)을 들려주었던 Stephanie Weirich에게 많은 대학원생들이 모여들어서 이런 저런 질문을 했다. 그와중에 다른 저년차 대학원생들과도 많은 이야기를 할 수 있었다. 다들 연구가 어렵고 힘들었던 경험을 나누며 위로하며 더 나은 연구를 하자는 얘기와 함께 이메일 주소를 나누었다. 사실 지금까지 별로 더 교류한 바는 없지만 다시 학회에서 만나면 반가울 것 같다.

결론

그동안 참석했던 학회와는 다르게 내가 직접 한 일과 관련된 이야기하는 시간을 많이 가졌던 것이 무척 좋은 경험이었다. 학회에 참석한다는게 단순히 톡을 듣는게 아니라 사람들을 만나고 내가 한 일을 소개하는 것이라는 생각이 점점 강하게 든다. 앞으로도 좋은 연구를 많이 해서 학회에 참석해서 의미있는 이야기를 할 수 있도록 잘 준비해야겠다는 생각이 들었다. 그래서 더 많은 사람들과 친구가 되고, 더 많은 사람들과 더 자극적인 대화를 많이 나눌 수 있게 되면 좋겠다. 진짜 제대로 학회에 다녀왔다는 이 충실감을 다음에 갈 때에도 느낄 수 있도록 노력해야겠다.

다만 학회장에서 계속 톡 듣고 이야기하느라 사진도 많이 못찍고 인도 구경도 전혀 하지 못해서 조금 아쉬웠다. 그래도 학회에 적극적으로 참석하는게 더 의미도 있고 재미있는 것 같다 :-)

다시 한 번 학회에 참석할 수 있도록 도와주신 이광근 교수님과 허충길 교수님, 그리고 연구비를 지원해주신 분들께 정말 감사드립니다. 계속 열심히 하겠습니다.