



PLDI 2014

Edinburgh, Scotland

서울대학교 오학주

2014.06.09 - 2014.06.15동안 영국 에든버러에서 열린 PLDI 2014에 논문발표차 다녀왔다. PLDI(ACM Conference on Programming Language Design and Implementation)는 프로그래밍 언어 분야에서 실제적인 연구결과들이 주로 발표되는 학회로, 여느때와 마찬가지로 올해에도 전세계에서 다양한 연구자들이 모여 활발한 교류가 이루어지는 곳이었다. PLDI와 함께 열린 워크샵, 튜토리얼에 참가했던 경험을 공유하고자 한다.

학회 분위기

PLDI는 월,화,수 3일에 걸쳐서 진행되었다. PLDI에는 해마다 논문접수 편수가 늘고 있는데 이번 학회에는 총 287 편의 논문이 접수되어 또 다시 기록을 세웠다고 한다. 이중 52편이 학회에서 발표되었고 두 세션으로 나뉘어 동시에 진행되었다. 2012년부터 도입된 광고발표(teaser talk)가 매일 아침에 있었고, 사람들은 이를 듣고 어느 발표를 들을지 선택하였다. 초청발표도 흥미로웠고 그 외 기억에 남는 논문발표가 많이 있었다. 목, 금에는 위성학회와 워크샵들이 진행되었다. 월요일부터 수요일까지는 PLDI만 열렸기 때문에 집중된 분위기였던 반면, 목, 금에는 굉장히 다양한 워크샵, 튜토리얼, 학회들이 동시에 이곳저곳에서 열렸기 때문에 좀 느슨한 분위기였다. 나는 SOAP (Workshop on State-of-the Art of Java Program Analysis), APPROX (Workshop on Probabilistic and Approximate Computing)등에 참석하였다.

학회는 에든버러 구시가지에 위치한 Assembly Rooms (AR)라는 곳에서 열렸다. AR은 18세기 말에 지어진 고풍스런 건물인데 200년 이상 사람들의 모임, 발표, 공연 장소로 활용되어 오는 곳이라고 한다. 우리로 따지면,

코엑스나 예술의 전당 같은 곳인 셈이다. AR은 역사가 오래되고 예전 외관을 그대로 간직하고 있고 대신 규모는 작았는데, 메인 장소였던 Music Hall과 Ballroom을 제외하고는 워크샵, 튜토리얼들이 열린 Artist Room등은 기껏해야 20명 정도 앉을수 있을 정도로 아담했다.

이번 학회는 여느때보다 사람들끼리 교류할 수 있는 기회를 많이 제공 하려는 주최측 노력이 엿보였다. 75분의 논문발표 세션이 끝나면 30분의 coffee break가 주어졌다. 커피나 홍차를 마시면서 자유롭게 돌아다니며 사람들과 교류하는 시간이다. 특히, 점심도 테이블에 앉아서 먹는 형식이 아니라 작은 그릇들에 다양한 음식들이 담겨있고 하나씩 가져다가 학회장을 돌아다니며 먹을 수 있게 했다. 그래서 1시간이 넘는 점심시간도 사람들끼리 교류하는데 쓰였다. 음식들도 괜찮았고 평소 옆 사람과만 이야기할 수 있었던 다른 학회점심시간에 비해서 더 나왔다.

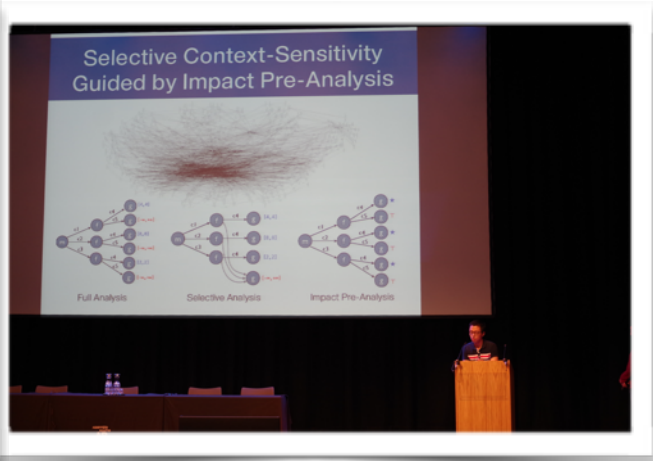


논문 발표

이번에 발표한 논문은 정적 분석기의 정확도를 필요한 경우에만 선택적으로 향상시킴으로써 추가비용을 최소화 시키는 방법에 대한 연구였다. 지난 POPL에 냈었는데 탈락했고 좀 더 보강을 해서 PLDI에 낸 것이었다. POPL 에서는 대부분의 리뷰어가 논문의 주된 내용이 너무 간단하다는 지적을 받았다. 하지만 핵심 아이디어가 전달이 잘 되지 않은 느낌을 받았고, 이 부분을 명료하게 하고 뒷받침 할 수 있는 응용(선별적 관계분석)을 하나 더 추가하여 PLDI에 냈더니 이번에는 대부분의 리뷰어로부터 아이디어가 참신하다는 평가를 받았다.

내 발표는 PLDI마지막날인 수요일 오후에 있었다. 학회발표는 첫날하는게 제일 좋은 것 같다. 마지막 날이라고 해서 준비를 더 잘 하게 되는것도 아닐뿐더러 학회에 맘편히 참가하기도 어렵기 때문이다. 그리고 일찍 발표하면 내 연구로 사람들과 이야기할 기회가 더 많아진다. 이번 논문은 발표준비하기가 좀 까다로웠다. 논문의 제목이나 주요 내용은 선택적 함수 문맥 구분 분석(selective context-sensitivity)인데 이를 일반화해서 얘기하려고 하다보니 자칫하면 뜬 구름잡는 이야기가 되기 쉬웠다. 거기다가 2012년 스파스 분석과 연관지어 연구동기와 과정을 소개 하려다 보니 발표의 포인트가 여러개가 되어 버리는 문제가 있었다. 이런 문제들로 슬라이드는 발표날 아침이 되어 서야 마음에 드는 버전이 완성되었고 마음에 썩 드는 발표를 하지 못했다. 좀 아쉬웠지만 그래도 끝나자 후련했고 내 연구로 PLDI에서 두번째 발표했다는 것도 뿌듯한 경험이었다.

주제 자체가 여러 사람들이 관심을 가지는 주제이다 보니 질문과 발표 후 토론은 어느때보다 많았다. 발표후에 Peter O'Hearn 교수님, Manu Sridharan, Francesco Logozzo, Eric Bodden 등과 이야기를 했다. 특히 Eric Bodden과 오래 이야기를 했는데 자신들이 만든 분석기(FlowDroid)가 context-sensitivity때문에 매우 느리다면서 선별적으로 어떻게 할 수 없을지에 대해서 이야기 했다. FlowDroid는 context-sensitivity를 구분 할 때 우리처럼 함수가 호출되는 경로를 이용해서 구분하는 방식이 아니라 함수의 입력값의 경우별로 구분한다. 이런 분석의 경우, 각 함수마다 미리 가능한 입력 값을 알아 낸 후 정확도 향상에 도움이 되는 입력값만 구분하고 싶다는 것이었다. Francesco는 선택적 관계 분석을 옥타곤외의 다른 도메인에도 적용가능할지 물어보았고, Peter O'Hearn 교수님은 상향식 분석(summary-based)에도 적용가능할지 질문하셨다. 대부분의 질문들이 이번 논문을 쓰면서 생각해본 주제였는데, 우리의 경우는 동기가 없어서 하지 않고 있던 문제들이었다.



인상깊게 들은 발표들

PLDI와 워크샵에서 들은 발표중에서 기억에 남는 몇 편을 소개한다.

FlashExtract: A Framework for Data Extraction by Examples

임의의 일정한 형태를 가지는 파일에서 필요한 정보를 쉽게 추출하기 위해서 프로그램 합성 기법을 이용한 연구이다. 자동으로 생성된 로그나 텍스트 파일이 있을때 내가 원하는 정보를 추출하기 위해서 가장 흔히 사용하는 방식은 스크립트를 짜는 것이다. 하지만 서로 다른 로그 파일 형식마다 서로 다른 스크립트 프로그램을 만들어야 하며, 또한 일반적인 사용자에게는 스크립트를 짜는 것 자체가 매우 어려울 수 있다. 이 연구의 목적은 사용자는 추출하고자 하는 정보를 example로 몇개 보여주면 시스템이 귀납적으로 이를 학습하여 원하는 정보를 추출해주는 스크립트 프로그램을 합성해주는 것이다. 참 재미있는 접근 방식이고 프로그램 합성의 매우 좋은 응용이라는 생각이 든다. 그리고 프로그램 합성 기술이 이렇게 실용적으로 쓰일 수 있다는 사실도 놀라웠다. 합성 알고리즘을 논문에서 자세히 읽어봐야 겠다는 생각이 들었다.

Abstraction Refinement for Program Analyses in Datalog

정적 분석 관련 세션이 두 개 있었는데 이 논문이 첫 번째 발표였다. Georgia Tech의 Mayur Naik과 양홍석 교수님이 함께 쓰신 논문으로 네 리뷰에서 모두 A를 받았다고 한다(best paper로도 뽑혔다). Datalog는 Prolog와 유사한 로직 프로그래밍 언어로써 포인터 분석을 비롯한 다양한 정적 분석을 간결하게 표현하는데 많이 쓰인다. 이 논문은 Datalog로 표현된 프로그램 분석에서 효율적으로 요약의 정도를 다듬어가는(abstraction refinement) 효율적인 방식을 제안하였다. 즉, 처음에는 부정확한 요약을 이용해서 프로그램을 분석한 후 주어진 query가 증명이 안 된경우에, 현재 요약정도를 다듬어서 더 정확한 요약상태로 다시 분석을 시도한다. 이 때 핵심은 어떻게 현재 요약상태를 다듬는것인가인데, 이 논문에서는 Datalog 프로그램의 특징을 절묘하게 이용하여 이 부분을 해결하였다. 현재 요약으로 query를 증명하는데 실패했다면 동일한 원인으로 실패할 요약들을 모두 파악하고 이를 다음 후보군에서 제외시킴으로써 탐색공간을 효율적으로 좁혀 나가게 된다. 결과적으로 주어진 query를 증명하는데 필요한 최소의 요약을 찾아내거나 주어진 요약 도메인으로는 증명할 수 없다는 결론을 얻게 된다. 비교적 간단한 분석이어서 이러한 문제 설정과 해결이 가능한 것인데, 우리처럼 무한한 도메인을 쓰는 분석의 경우는 어떤 이슈가 있을지 궁금해졌다.

Hybrid Top-down and Bottom-up Interprocedural Analysis

이 논문도 역시 Mayur Naik 그룹과 양교수님이 함께 쓰신 논문이다. 이로써 양교수님은 이번 PLDI에 우리논문을 포함해서 총 3편을 내셨다. 이 논문은 사실 작년 닥스틀 세미나에서 Mayur를 처음 만났을때 이야기를 들었던 것인데 몇번 떨어진 후에 결국 PLDI에서 발표하게 된 것이라고 한다. 이 논문의 내용은 고전적인 정적 분석 문제에 대한 것인데 함수 호출을 분석하는데 두 가지 방법(상향식 분석과 하향식 분석)의 장점만을 취하는 방식에 대한 것이다. 말단 함수부터 main함수까지 써머리를 만들어 가는 상향식 분석은 함수 내부의 분기로 인해서 경우가 다양해지면 써머리가 복잡해지고 때문에 분석이 느려진다. 따라서 실제로 유용한 입력에 대해서만 케이스를 구분하면 좋을텐데 이를 위한 힌트를 하향식 분석에서 얻는 아이디어이다. 상향식 분석의 경우는 입력을 모르기 때문에 모든 케이스를 구분하지만 하향식 분석은 입력을 알기 때문에 경우를 좁힐 수 있다. 어떻게 보면 간단한 아이디어인데 상향식/하향식 분석의 특징을 명료하게 나열한 점, 이로부터 두 스타일의 분석을 어떻게 합칠수 있을지에 대한 아이디어를 끌어낸 점, 좋은 실험결과 등등이 좋았던 논문이다.



Introspective Analysis: Context-Sensitivity, Across the Board

내 논문 발표 바로 다음 논문으로 내 논문과 매우 비슷한 주제를 다루고 있다. 때문에 발표하면서 우리 논문을 여러 번 인용하기도 했다. 함수 문맥을 구분하는 분석이 보통 매우 큰 비용이 들기 때문에 비용이 폭발적으로 증가하지 않는 컨텍스트만 골라서 선택적으로 적용하는 것이 이 논문의 목적이다. 우리 논문과 같은 목표를 가지고 있는데 해결방법이 많이 다르다. 이 논문은 먼저 문맥을 구분하지 않는 분석(context-insensitive)을 전분석으로 수행한 후 분석과정에서 정보를 얻어서 어느 부분에서 문맥을 구분하면 비용증가가 크지 않을지를 휴리스틱하게 판단해낸다. 이 휴리스틱들은 잘 이해하지 못했는데, 결국 기술적으로는 별다른 새로울것이 없는 내용이지만 포인터 분석에 한정해서만큼은 그 의미가 큰 것 같았다. 포인터 분석이라는 문제 자체가 매우 구체적이어서 이렇게 특화된 아이디어를 제시해도 실험을 통해서 그 효과를 입증하면 논문을 받아주는 것 같다. 다양한 포인터 분석에 대해서도 여러가지 클라이언트를 붙였을때 성능향상이 어떻게 되는지 잘 정리함으로써, 컴파일러나 기타 포인터 분석을 사용하는 툴에서 실제 성능향상을 기대할 수 있다는 점이 인상적이다.

Expressing and Verifying Probabilistic Assertions

가장 인상적이었던 발표 가운데 하나였다. 주제도 흥미로웠지만 핵심개념만을 쉽고 간단히 표현한 슬라이드와 물 흐르듯 끊임없이 이어지는 발표가 매우 훌륭했다. 특성상 확률적으로 동작하는 프로그램들(sensors, 머신러닝, approximate computation)의 경우에는 검증하고자 하는 성질도 확률적일텐데 이를 위한 probabilistic assertion을 표현하고 검증하는 기법을 제시하였다. 즉, 보통의 assertion의 사용이 프로그램의 모든 실행에서 항상 성립하는 성질을 표현하고 증명하는 것이 목적이었다면, 이 논문에서 probabilistic assertion(passert)은 세가지 인자 e , p , cf 를 받는다. e 는 보통의 assertion에 들어가는 검증하고자 하는 성질을 표현하고 p 는 이것이 만족될 최소 확률, cf 는 신뢰구간이다. 이렇게 표현된 passertion을 검증하기 위해서 세 단계를 거치는데 symbolic execution, static analysis, dynamic analysis를 모두 이용한다. 첫째로 symbolic execution을 이용해서 확률모델(Baysian network)을 만든 후 이를 최적화 하는 단계를 거치는데, 이 부분이 핵심인 것 같았다. 자세히 이해하지는 못했지만 최적화 단계를 지나면 passert를 검증하기가 매우 쉬워지는 것 같았다. 그래서 직접 검증을 하거나 아니면 샘플링을 하는 단계를 거친다.

Automatic Runtime Error Repair and Containment via Recovery Shepherding

Martin Rinard 교수님의 발표로 많은 사람들이 몰렸고 가장 많은 질문과 호응을 이끌어냈던 발표였다. 이 논문에서는 RCV라는, null-dereference와 divide-by-zero 오류가 프로그램에서 발생하더라도 런타임에 이를 잡아내서 프로그램이 죽지 않고 계속 실행되게끔 하는 시스템을 소개하고 있다. 사실 접근 방식은 별것 없는것 같고 그냥 에러가 발생하면 임의의 값을 리턴하거나 무시하는 식이다. 예를 들어 divide-by-zero가 발생해서 시그널이 발생하는데 시그널 핸들러를 바꿔치기 해서 0을 리턴한 후 프로그램 실행을 이어가게 한다. 또 null-dereference 시그널이 발생하면 0에 값을 쓸때와 함수호출을 하는 경우는 무시하고 0으로 읽는 경우에는 0을 리턴한다. 이 논문은 이런 방식으로 프로그램을 계속 실행해도 별 문제없고 실제 프로그램들에 대해서 어떤 일이 일어나는지에 대해서 매우 상세한 실험결과를 가지고 보인다. 실험 부분이 인상적인데 CVE 데이터베이스에서 특정기간에 발생한 모든 null-dereference와 divide-by-zero오류를 모으고 자신들의 실험환경에서 재현 가능한 것들을 추려내는 작업은 정말 많은 인내심과 노력이 필요할 것 같았다. 어찌보면 황당한 아이디어로 보일수도 있을텐데, 이를 추진해내는 자신감과 완성도 있는 결과 등등 배울점이 많은 연구였다. 또한 Martin Rinard 교수님의 엄청난 쇼맨십은 정말 인상적이었다.



Verification Modulo Versions: Towards Usable Verification

두 버전 사이의 차이에서 발생하는 알람만 보여주는 기법에 대한 논문이다. 프로그램 P를 분석해서 나오는 알람을 사용자가 모두 분류하고 분류했다고 하자. 이 알람들 중에는 false alarm도 있고 사용자가 의도적으로 무시한 알람도 있을 것이다. 그 다음 P를 P'으로 수정했을때 다시 분석을 하면 사용자가 P에서 이미 분류했던 알람들이 또 다시 나올 것이다. 이 논문의 목적은 이 때 P'에서 새로 발생한, 사용자가 아직 보지 않은 알람들만 새로 보여주는 것으로써 이를 baselining이라 부르고 있다. 기존의 상용 정적 분석들에서도 baselining을 지원하는데 모두 프로그램의 위치와 모양을 기반으로 하고 있어서 프로그램 변화에 취약하다고 한다. 그래서 이 논문은 두 프로그램의 의미가 달라질 경우에만 알람을 보여주는 의미기반 기법을 제시하는 것이 목적이다. 의미기반 이라고는 하지만 어느정도 프로그램의 위치정보를 이용하는 점, 그리고 동작 방식(P에서 알람이 발생할 조건을 찾아서 P'에 이를 넣고 분석)이 그다지 새로울게 없다는 점 등에서 약하긴 한데 이 과정을 요약해석의 틀에서 이론적으로 정립하고 이 방식의 안전성으로 보였다는 점을 강조하였다. 그리고 마이크로소프트에서 개발하는 C# 프로젝트를 대상으로 실험한 결과가 눈에 띄는 논문.

Improving JavaScript Performance by Deconstructing the Type System

연구내용과 결과가 매우 인상적인 논문. 사실 논문 발표를 듣지는 못했지만, best paper를 받은 논문이고 한국인이 쓴 논문이어서 나중에 읽어봤다. 브라우저에 들어가는 자바스크립트 엔진들이 특정 벤치마크에서는 좋은 성능을 보이는데 실제 웹페이지를 기반으로 하는 벤치마크에서는 훨씬 안 좋은 성능을 보인다고 한다. 이 논문은 이 현상의 주요 원인을 규명하고 해결하여 자바스크립트 실행 성능을 평균 36% 향상시켰다. 크롬 V8에서 쓰이는 최적화 기법들이 특정 벤치마크 프로그램에서는 수십배 이상의 성능 향상을 가져오는데 실제 웹 사이트 기반의 벤치마크인 JSBench에 대해서는 효과가 없거나 무시할 만한 수준밖에 되지 않는다고 한다. 또 이 문제의 원인이 웹 사이트 코드에서는 컴파일러가 오브젝트의 타입을 유추하는데 빈번히 실패하고 따라서 효과적인 코드를 생성하지 못하기 때문이며 이를 해결하는 휴리스틱을 제시하였다. 관련 분야에 있는 사람이라면 누구나 겪고 있을법한 문제이지만 아무도 시도해볼 엄두를 내지 않았던 일을 해결 한점이 대단해 보였다. 이 논문의 일저자인 안원선 박사님과 저널을 같이 먹었는데 산더미같은 V8코드를 뒤져가며 원인을 찾아내느라 매우 고생하셨다고 한다. 이 얘기를 듣고 논문을 읽어보니 PLDI에서 best paper를 받을만한 연구라는 생각이 들었다.

Large-Scale configurable Static Analysis (Mayur Naik, Invited talk at SOAP)

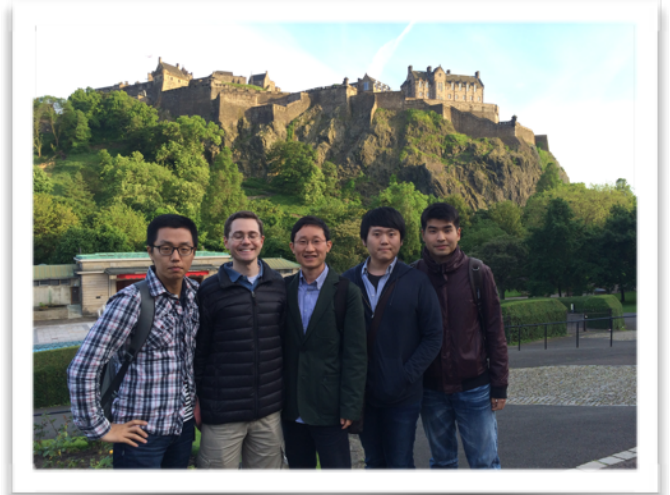
SOAP(State Of the Art in Java Program Analysis) 워크샵에서 Mayur Naik의 초청강연이었다. Mayur의 논문들 중 최근의 정적 분석 관련 논문들은 대부분 읽어보았지만 도대체 이런 연구를 하는 이유가 무엇인지, 왜 중요한 문제인지에 대해서는 잘 와닿지 않았었는데, 이런 점을 명확히 해준 발표였다. 연구의 기본적인 주제는 정적 분석기의 성능을 결정짓는 파라미터들이 있는데 최적의 파라미터 세팅이 무엇일지에 대해서 탐구해 보는 것이다. 처음에는 효율을 따지지 않고 최적의 파라미터값이 무엇인지에 대해서부터 알아보는 연구부터 시작해서 요즘에는 효율적으로 파라미터 세팅을 찾는 알고리즘들을 개발해 오고 있다. 문제가 잘 정의되어 있다는 느낌을 받았고, 그 문제내에서 다양한 하위문제를 설정하여 하나씩 풀어가고 있는 것이다. 이런 스타일의 연구방법도 배울 점



이 있다는 생각이 들었다. 큰 문제를 말이되도록 잘 설정한 후 그 범위내에서 할 수 있는게 어디까지인지 탐구해보는 방식. 자신만의 분야를 만들어가고 있다는 느낌이 들었다.

사람들

이번 학회에서도 다양한 사람들을 만날 수 있었다. 특히 작년 닥스톨 세미나에서 친해졌던 사람들을 다시 만나게 되어 좋았다. Mayur Naik과 그의 학생인 Xin과는 관심 분야가 잘 맞아서 서로의 논문의 아이디어에 대해서 깊이 있게 토론할 수 있었다. 같은 PL분야라도 구체적인 얘기를 나눌 수 있는 경우는 많이 없기에 즐거운 경험이었다. Ondrej Lhotak, Eric Bodden 등과도 근황과 요즘 연구이야기를 나눴다. 하지만 무엇보다 몇년 전에 우리 연구실을 방문했던 Will Klieber를 다시 만나게 되어 반가웠다. Will을 보게될 거라고는 생각도 못했는데 SOAP에 논문발표를 한다고 했다. 예전 그대로였고, 연구 이야기뿐 아니라 점심 저녁을 같이 먹으며 즐거운 시간을 보냈다. Context-sensitivity에 대해서 이야기 했던 Yannis, 덕환이형 연구실에서 온 사람들 등 이야기를 나눠보고 싶은 사람이 더 있었는데, 기회가 마땅치 않았다.



에든버러

에든버러는 도시 중간의 성이 인상적인 깨끗한 도시였다. 도시 한가운데에 있는 에든버러 성은 시내 어디에서도 잘 보일만큼 높은 절벽위에 위치해 있다. 마침 PLDI기간동안의 에든버러 날씨는 매우 좋았고 호텔과 학회장을 오가며 파란 하늘을 배경으로 에든버러 성을 실컷 감상할 수 있었다. 시내 주변에 언덕들이 있는데 칼튼 힐의 전망대에서 바라본 에든버러의 모습은 그야말로 장관이었다. 북쪽과 동쪽으로는 멀리 바다가 보이고, 남쪽 멀리는 깎아지른 듯한 절벽이 있고 남에서 서쪽에 걸쳐서 에든버러 구 시가지가 내려다 보인다. 전망대 바로 아래에는 잔디밭에서 사람들이 일광욕을 즐기고 있었다.

토요일에는 스코틀랜드의 하이라이트라는 하이랜드 투어를 다녀왔다. 하이랜드는 스코틀랜드 북쪽에 걸쳐있는 주로 언덕과 호수로 이루어진 지대이다. 하이랜드 투어는 네스호를 목적으로 하면서 가고 오며 스코틀랜드의 다양한 자연환경을 구경하는 방식으로 진행되었다. 어느 한 여행을 통해서 예약을 했고, 아침 8시에 투어가 시작되었다. 네스호까지는 대략 5시간 걸리고 중간중간 사진찍는 시간을 준다. 사실 더 인상적인 투어를 기대했는데 너무 긴 이동시간(12시간)과 수동적인 투어방식 때문에 기대에는 못 미쳤다. 네스호는 괴물 네스로 유명한 우리에게 친숙한 호수이다. 배를 타고 호수를 한바퀴 둘러보는데 호수 물이 흡사 석유처럼 까맣고 독특했다. 고요한 주변 경치를 보며 호수 위에 있으니 마음이 평온해졌다.



맺음말

개인적으로 PLDI에서 다시 발표할 수 있어서 감사하고 뿌듯한 여행이었다. 앞으로 더 열심히 해서 좋은 결과를 꾸준히 내보고 싶다는 자신감과 용기도 더 커졌다. 좋은 연구를 하도록 늘 장려해 주시고 지원해 주시는 이광근 교수님께 감사드립니다.