

SAS 2009 , PCC 2009

UCLA, USA, August 8 ~ 17, 2009

순호형과 UCLA에서 열린 SAS 2009 학회와 PCC 워크샵에 참석하였다. 올해에는 SAS와 LICS 두개의 학회와 4개의 부속 워크샵이 함께 진행되었는데 행사가 SAS로 시작하여 PCC로 끝난 덕분에 우리는 행사기간 내내 UCLA에 머물렀다.

학회 참가자 기숙사 건물에 묵었다. 기숙사 건물은 □자로 생겨서 건물 밖에서는 일단 중앙의 공터로 들어온 다음에 중앙 공터에서 건물 내부로 들어가게 되어있었다. 공터에는 나무와 벤치가 있는데 외부의 시선에서 차단된 하늘과 트여있는 공간이란 느낌이 좋았다. 학회는 기숙사 지역 중앙의 학생회관 같은 건물(Covel Commons)에서 진행되었다. 학교 곳곳에 Commons란 이름이 붙은 건물이 있는데 대체로 1층에 카페테리아나 식당 있고 2,3층으로 세미나실이 있었다. 그런 건물들을 Commons라고 부르나보다.

SAS 2009

9일부터 11일까지 3일간은 SAS Conference가 진행되었다.

Rastisloav Bodik의 Invited Talk과 한양대팀, Songtao Xia의 발표가 흥미로웠고

Patrick Cousot 교수님이 ROSAEC 워크샵에서의 최광무교수님처럼 모든 발표에 코멘트를 하는 것이 인상적이었다.

Rastiv Bodik,

Algorithmic Program Synthesis with Partial Programs and Decision Procedures

프로그래밍을 하다보면 세부사항을 컴퓨터가 자동으로 채워주었으면 하는 경우가 있다. 이 세션에서는 그 중에서도 "올바르지만 느린 구현"과 "새로운 구현의 뼈대"를 바탕으로 컴퓨터가 "새로운 구현"을 도출하게 하는 연구에 대해 발표하였다. 뼈대에 채워질 코드가 돌아다니는 문제공간이 유한하다면, Model Checker로 원래의 프로그램과 동일한 결과를 보장하는 코드를 찾아낼 수 있을 것이다. 컴파일시간에 빈칸을 채우는 기법 (Programming by Sketch)와, 동적 시간에 빈칸을 채우는(Angelic Operator) 두 가지 기법을 소개하였다.

최초의 구현이 올바르다는 것을 검증하면 그 위에서 안전하면서도 효율적인 프로그램을 기계를 이용하여 유도할 수 있는듯 해 매력적이었다. Theorem Proving 커뮤니티에서는 증명에서부터 안전한 구현을 도출하는 연구를 하고 있다고 들었는데, 증명에서 비효율적이지만 안전한 구현을 도출하고 거기서부터 Programming by Sketch를 이용하여 점진적으로 개선해 나갈 수 있으면 어떨지.

Songtao Xia,

Inferring Dataflow Properties of User Defined Table Processor

DB 질의가 포함된 프로그램을 분석 할때, 프로그램의 실행 흐름과 질의문의 의미를 같이 고려하여 "사용되지 않는 Table Entry"등을 찾아내는 분석기법을 소개하였다. 이 작업은 C#과 연동되는 DB 질의실행기를 최적화하는데 사용된다고 하였고 그래서인지 C#으로 작성된 프로그램의 Control Flow를 구하는 분석을 소개하는데 절반에 가까운 시간을 할애하였다. DB 질의문이나 Script언어를 고려한 분석은 큰 규모의 프로그램을 검증하고 최적화하려면 넘어야하는 산이라고 생각하던 터라 더욱 관심있게 들었다. 이론적으로 새로운 것이 없고 노력은 많이 필요하여 학계에서 다루기에는 적합하지 않을지 몰라도 분석 커뮤니티가 더 커진다면 산업체를 중심으로 연구가 본격화되지 않을까 싶다.

Clement Hurlin,

Automatic Parallelization and Optimization of Programs by Proof Rewriting

이 세션에서는 주어진 프로그램에 대한 Separation Logic기반의 증명이 존재할 때 Framerule을 이용해 프로그램을 (Observationaly) 동일한 다른 프로그램으로 다시 쓸 수 있다는 것을 보이고, 이것을 이용한 최적화를 제안하였다. Separation Logic에 관심 있는 사람이라면 읽어볼 만한 (어렵지 않고 흥미로운) 논문이다. 하지만, 이론적으로 흥미로운데 반해 실험결과가 없어 아쉽기도 하였다. 아마도 Separation Logic으로 산업현장의 소스코드에 대한 만족할만한 증명을 제공할 수 있는 분석기가 아직 없기 때문이 아닐까 추측해본다.

David Schumit and KyungKu Doh,

Abstract Parsing : Static Analysis of Dynamically Generated String Output Using LR-Parsing Technology

프로그램 실행중에 만들어질 수 있는 문자열을 직접 모으는 대신 문자열이 파싱된 결과를 모음으로서 빠르게 문자열이 올바른 모양을 지니고 있는지 검증하는 기법을 소개하였다. 발표는 David Schumit 교수님이 내용을, 도경구 교수님이 실험부분을 담당하여 진행했다. 내용을 익히 알고있던 터라, 내용 자체보다는 어떻게 발표하는지 살펴보는 데 집중하였다. 어떤 논문이 되었던지 질문시간을 포함해서 30분 남짓 주어지는 발표시간동안 모든 디테일을 다루기는 불가능하다. 때문에 발표는 무엇을 포기하고 무엇을 남겨야 하는지 잘 선택하는 문제가 된다. Schumit 교수님은 세부사항을 최대한 감추면서 옛날이야기를 하듯 부드럽게 직관을 전달하였다. 세부사항을 언급해야 할 필요가 있으면 Formal한 정의를 살짝 보여준 뒤 곧바로 딱 하나의 예제를 통해 직관이 실제 어떻게 적용되는지를 보여주었다. SAS 셋째날 있었던 Schumit 교수님의 다른 발표도 비슷한 느낌으로 진행된 것으로 미루어, 고유한 발표방식인 것 같다. SAS와 PCC를 합쳐서 20명이 넘는 발표자의 발표를 보니 노련한 발표자일 수록 세부사항을 전달하는데 시간을 많이 할애하지 않았다. 직관을 전달하되 꼭 필요한 디테일을 선별해서 첨가하는데 학회 발표의 묘가 있다.

E.D.Clark,

27 Years of Model Checking, E.D.Clark

Model Checking이 지나온 길에 대해 개괄적으로 설명하였다. 아마도 2007년에 우리 연구실을 방문하여 발표했던 내용과 크게 다르지 않을 것이다. Model Checking에 대해 이미 알고있는 사람은 무용담을 듣는 가벼운 기분으로, 잘 모르는 사람은 "Sound하지만 Termination이 보장되지 않는 검증기법"이라는 정도의 이해를 얻어서 돌아갈 수 있지 않았나 한다. 개인적으로는 Predicate Abstraction에 대한 Model Checking 대부의 통찰을 듣고 싶었는데 들을 수 없었다. Talk의 주제가 포괄적이어서 Predicate Abstraction만 자세히 다루기에는 시간이 부족했을 것이다. Model Checking 전체를 아우르는 입장에서는 Software Model Checking이란 분과에서 사용하는 하나의 State Abstraction 기법에 불과할지도 모르겠다.

PCC 2009

15일에는 PCC Workshop이 있었다.

두개의 invited talk이 모두 영양가 있었다. Andrew Appel의 발표는 내용이 익숙하지 않아 절반정도 밖에 따라가지 못해 아쉬움이 남는다. 나머지 발표중에는 Juan Chen과 David Pichardie의 발표가 들을만 했다.

Kelly Hayhurst,

Mdeing the Gap, An effort to aid the transfer of formal methods technology

산업 프로젝트에 "Formal Method"를 적용하는 어려움에 대해 이야기하였다. NASA의 비행기 개발 부문(우주 개발이 아닌)에서 "표준 개발 방법론"을 개발하고 문서화하는 일을 담당하는 분이 Speaker였다. 프로그램 분석에 대해 잘 모르는 분이었고, 그래서 더 가치가 있는 세션이었다고 생각된다. 산업체에서는 치밀하게 설계된 수천 페이지에 달하는 공정 매뉴얼을 따라 항공기를 설계하고 제작한다고 한다. 지금은 소프트웨어 검증 비용이 항공기 전체 개발 비용의 절반에 육박하기 때문에 "Formal Method"를 도입해야 할 필요성은 절감하지만 어느 산업체에서도 공정에 포함시키지 못하고 있다. (Airbus 360은 유일한 시범 케이스다) 도입하지 못하는 이유는 "Formal Method"가 무엇을 검증해주는지 "품질관리" 측면에서 설명할 수 없기 때문이라고 한다.

David Pichardie

Towards a Certified Lightweight Array Bound Checker for Java Byte Code

정적 분석을 통해 Java byte code에서 array bound check를 가능한 없애서 실행속도를 높이는 논문이 있는데, 그것에 PCC를 적용하여 실행 전에 정말 없애도 되는 검사들만 사라졌는지 실행전에 한번 확인하는 기술을 소개하였다. PCC의 한 갈래인 abstract carrying code에서는 trusted base의 크기를 줄이기가 매우 어렵다. abstraction이 올바른지 확인하기 위해서는 abstraction을 바탕으로 다시 한번 고정점 계산을 수행해야 하고, 이를 위해서는 분석기 전체가 필요하기 때문이다. 저자는 이런 문제를 fix-point 확인에 최적화된 Semantics를 디자인하여 해결하였다. Domain과 언어에 따라 최적화 Semantics는 천차만별일 것이므로 이 작업이 직접적인 도움이 되는 일은 없겠지만, abstract carrying code를 사용할 일이 있다면 한번쯤 참고할만할 것 같다.

Our Talk

우리 차례는 오전 11시 30분 부터 12시 까지 30분이었다. 순호형이 발표하였고 노트북에서 프로젝터를 잘 인식하지 못해 시작이 약간 지연되었지만, 발표 자체는 연습할때에 비해서 부드럽게 잘 진행되었다. 13,14일 이틀간 준비하면서 어떻게 하면 디테일을 배제하면서도 뜬구름잡는 느낌이 들지 않게 발표할지를 주로 고민하였다. 분석예제를 거의 애니메이션 수준으로 보여주면 많은 말을 하지 않아도 직관적으로 이해할 수 있겠다 싶어서 50컷에 달하는 분석 예제 애니메이션을 만들어 보여준 것이 주효한듯 하다.

질문시간에 Naoki Kobayashi 교수가 Syntax Check 이상의 것을 할 수 있냐는 질문을 하였고 점심시간에도 두 사람이 (Ewen Denney, Sagar Chaki) 같은 질문을 하였다. Syntax만 가지고 얼마나 대단한 성질을 체크할 수 있겠느냐는 뉘앙스였다. 오후세션에 Syntax Check를 넘어설 궁리를 하다가 의외로 쉽게 방법을 찾았다. 우리는 Syntax만으로도 훌륭하다고 생각하여 다른 시도를 해볼 생각을 못했는데, 잘 모르는 사람의 눈에는 이상하게 보였던 모양이다. 다양한 관점을 접하고 공개적으로 평가받는 것이 얼마나 중요한지 새삼 느꼈다.

학회를 알차게 보내려면

처음 가보는 학회라서 제대로 즐기지(?) 못하는 기분이 들 때가 여러번 있었다. 어떤 준비가 필요한지 적어둔 것이 있어서 옮겨본다.

대화!

발표를 듣지 않아도 논문을 읽으면 무슨 연구를 했는지는 알 수 있다. 보다 중요한 것은 대화를 통해 다른 사람들의 통찰과 조언을 얻는 일인 것 같다. 학회에 오는 사람들의 관심분야와 얼굴(!)을 알아두고 발표할때 질문거리들을 만들었다가 쉬는시간에 이것저것 대화를 해 보면 좋을것 같다. 영어를 못하면 정보라도 있어야 이야기를 풀어나가기 쉬울테니까.

식사시간

점심시간이 2시간이 나 되기 때문에 식사시간을 다른 참석자들과 같이 보내는 것이 중요하다. 쉬는시간에는 친한 사람끼리 2-3명씩 무리지어 이야기를 나누기 때문에 처음 참석하는 사람들은 대화상대를 찾기가 쉽지 않지만, 식사시간에는 작게는 4명, 많게는 10명까지 같이 앉을 수 있다. 일단 같이 앉으면 통성명도 하고 대화도 하는 것이 인지상정이다. 이것을 마지막날 깨달아 아쉽다.

영어!

말하는 능력보다는 듣는 능력이 훨씬 중요하게 느껴졌다. 상대방의 말을 따라가는 동안은 한단어 두단어로 응수해도 대화가 이어졌지만, 상대의 말을 제대로 듣지 못하고 "Pardon?"을 외치는 순간 대화의 흐름이 딱 끊기는 것을 느꼈다.

발표 직전 노트북 확인

노트북에서 프로젝터를 인식하지 못하는 일이 발표 시작전에 종종 벌어졌다. 프로젝터를 인식하는 동안 사라진 시간 자체는 유동적으로 조절할 수 있지만, 뜻하지 않은 사건을 만났을때의 긴장을 피하기 위해서라도 미리 작동하는지 확인해두는게 좋겠다.

UCLA

첫날 만찬이 끝난뒤 소화도 시키고 학교지리도 알아볼겸 UCLA를 돌아보았다. 넓직한 평지에 여유있게 자리잡은 건물과 학교 곳곳에 가득한 수목이 아름다웠다. 서울대도 산속에 위치해있어 자연과 가까운 것으로는 빠지지 않지만, 건물들이 있는 구역과 그렇지 않은 곳이 완전히 다른 분위기인데 비해 UCLA캠퍼스는 둘이 자연스럽게 섞여있었다. 배우고 싶은 부분이다.

둘째 날 점심시간에 학생회관을 찾아가 25\$에 자전거를 빌렸다. 기숙사 키카드와 신용카드를 제시하니 별로 어렵지 않게 빌릴 수 있었다. 알고보니 기숙사 키카드가 UCLA 전체에 통용되는 ID 카드였다. 구내 식당부터 시작해서 도서관과 스포츠센터까지 거의 모든 시설을 이용할 수 있다. 우리는 식당과 자전거대여에 밖에 사용하지 않았지만...

자전거를 빌린 덕에 월요일부터는 행동반경을 UCLA 학교 밖으로 넓혔다. 덕분에 인근 마트에서 먹거리를 사와 아침,점심을 저렴하게 해결하고 남은 돈으로 학교 밖에 있는 제법 그럴듯한 식당에서 저녁을 즐길 수 있었다. UCLA 인근이 제법 잘사는 동네라고 들었는데 거리도 식당도 깨끗했다. 그만큼 비쌌지만.

우리가 방문해있던 기간은 여름방학이라 미국학생들은 집에 가있고 계절학기를 방문한 외국인(중국,일본) 학생들과 연습때문에 학교를 떠날 수 없는 운동부 소속 학생들이 대부분이었다. 그래서 장소에 따라 캠퍼스가 아주 다른 느낌을 주었다. 계절학기가 진행중인 기숙사 인근은 중국인이 바글바글하고 기숙사만 벗어나면 운동부 학생들이 여기저기서 뛰고 체조하고 있었다.

[Wontae Choi](#)