



VMCAI & POPL 2010

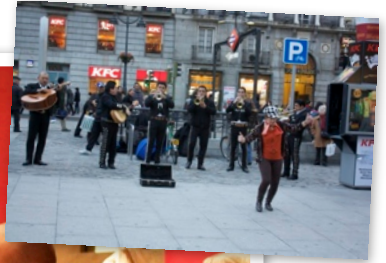
VMCAI (Verification, Model Checking, and Abstract Interpretation)

VMCAI와 POPL이 모두 같은 장소, 스페인의 마드리드에서 열렸다. VMCAI에는 대만에서 방문을 하셨던 왕 교수님과 그의 논문 (Deriving Invariants by Algorithmic Learning, Decision Procedures, and Predicate Abstraction)이 채택되었다. 아쉽게도 왕교수님은 참석을 못 하셨고, 발표는 같은 연구실의 공순호 학생이 맡게 되었다. 우리는 이 논문의 후속 논문을 CAV에 제출하기 위해 준비하고 있었고, 논문 마감에 학회와 겹치어 정신없이 보냈다. 급하게 제출한 논문에는 증명에 오류가 있었고, 결국 CAV에서 떨어지게 되었다. 이와는 별개로 순호의 발표는 순조롭고, 명료하여 많은 참석자들의 좋은 반응을 이끌어 냈다. 내 생각에는 Matthew Might의 발표와 함께 VMCAI의 최고의 발표였다.

Matthew Might는 Shape Analysis in the Absence of Pointers and Structure라는 논문을 발표했다. 제목부터가 흥미롭다. 순수한 함수형 프로그램(functional program)에서 모양 분석(shape analysis)이 무엇에 대응될지를 고민했다. 환경(binding environment)을 동적으로 할당되는 구조체로, 바인딩(binding)을 주소로, 값 환경(value environment)를 힙(heap)으로 보면 함수형 언어에서의 흐름분석(CFA)을 모양분석이라고 볼 수 있다. 이와 같은 연관성을 바탕으로 모양분석에서 쓰는 분석 테크닉을 CFA로 가져와서 정확한 분석을 하는 방법을 연구했다. 발표도 인상적으로 잘했지만 블로그에 가서 보고 참 배울점이 많은 사람이라는 생각을 했다. 장애가 있는 아들을 둔 아버지로서도 훌륭하였고, 연구나 생활태도도 배울점이 많았다. 특히 박사학생들이 읽기에 도움이 될 만한 글들이 많이 있으니 모두들 꼭 읽어봤으면 좋겠다.



다른 인상 깊었던 발표는 요약해석(abstract interpretation)에 조예가 깊은 이탈리아 Verona 대학의 Roberto Giacobazzi의 invited talk (Abstract Interpretation-Based Protection)이었다. AI를 통해 프로그램을 보호하는 방법을 일반화 하였다. 프로그램의 의미를 끄집어 내려는 공격자를 요약해석기로 생각하는 것이 재밌었다. 그렇게 생각하면 그 공격자가 얼마나 자세한 분석(프로그램의 이해)을 할 수 있는가의 따라 방어(프로그램을 변환)를 어떻게 할지를 결정할 수가 있게



된다. 자연스러운 생각인데 잘 일반화 하여 정리하였다. 요약해석으로 할 수 있는 좋은 연구 분야를 새로 개척한 사람이라는 인상을 받았습니다.

VMCAI 학회장에서 작년에 제가 MIT에 있을때 식사를 같이 했었던 Kuat Yessenov라는 친구를 다시 만났다. Victor Kuncak이 있는 EPFL에 방문했을때 한 연구를 VMCAI에 발표 했다. 논문 제목은 Collections, Cardinalities, and Relations인데 아쉽게도 발표시간이 CAV 논문 마감이라 겹쳐서 발표를 듣지는 못했다. 공동 저자인 Victor Kuncak의 제자 Ruzica Piskac는 공순호 학생과 악연이 있다고 한다. 순호가 미국에서 MSR 인턴으로 있을때 새벽에 빨래를 돌렸더니 이 여학생이 문을 벌컥 열고 들어와 새벽에 빨래 하지 말라고 소리를 질렀다고 한다. 내가 이제 시간이 많이 지났으니 한번 인사해 보는게 어떻겠냐고 해도 순호는 아직도 그 목소리가 잊혀지지 않는다면서 거부 했다. Vicktor Kuncak은 invited tutorial을 발표했는데 프로그램을 검증하기 위해 사용했던 모든 자동 검증기 및 프로그램 논리를 어떻게 확장하는지를 장시간에 걸쳐 설명했다. 자신들의 연구에 특화된 내용을 주저리 주저리 설명을 해서 듣기가 무척이나 힘들었다. MIT에 머무는 동안 Vicktor Kuncak이 쓰던 책상을 썼기에 또 그가 Rinard 교수님 밑에서 연구를 정말 잘했다는 사실을 알고 있었기에 관심을 갖고 들으려고 노력했으나 실패했다. Tutorial이라도 배경 지식이 없으니 이해하기가 어려웠다. VMCAI의 발표들은 신경이 온통 CAV 논문을 쓰는 것과 우리 논문의 내용을 제대로 발표하는데에 집중이 되어서 기억이 나는 논문이 별로 없었다. 다만, 학회 발표의 목적은 논문의 내용을 알리는 것이 아니라 사람들이 그 논문을 읽도록 유도해야 한다는 사실을 다시 한번 확인했습니다.

POPL(Principle of Programming Language)

POPL 학회가 시작되자 사람들이 다섯 배 정도 많아졌다. 지나가면서 스치면 대부분 형님(?)들이라 몸에 잔뜩 힘이 들어갔다. 한국에서 하던 식으로 그런 분들이 지나칠때마다 고개 숙여 인사하고 싶었는데, 서양 사람들이라 그게 안되니까 오히려 불편했다. 그렇다고 이름을 말하면서 아는척 할 용기는 나지 않았다. Mooly Sagiv 교수님은 실제로 보는 것이 처음이라 인사를 드리려고 시도만 하다 눈 인사만 어색하게 세 번 했다. 그리고, 대부분의 형님들께서는 항상 다른 사람들과 이야기를 나누고 계셔서 대화에 껴들 엄두가 나지 않았다. POPL의 발표는 시간도 짧고, 내용도 이론적인 논문들이 많아서 따라가기 힘들었지만 몇 가지 기억나는 논문들에 대해서 적어 본다.

Pure Subtype Systems - DeLesley S. Hutchins

개인적으로는 이 논문이 POPL 논문들 중에 가장 멋있는 논문이었다. 타입을 모두 배제하고 부분타입(subtype)규칙만으로 구축한 부분타입 이론을 제안하는 논문이다. 기존의 타입시스템은 객체(object)와 타입을 명시적으로 구분한다. 예를 들어, 객체 3은 정수 타입을 갖는다. 정수 타입 더하기 3은 정의가 되지 않는다. 하지만, 순수 부분타입 시스템(pure subtype system)에서는 이런 구분을 시원하게 깨버린다. 우리가 생각하는 모든 객체가 타입처럼 쓰이고, 반대로 타입이 객체처럼도 쓰일 수 있다. 이 시스템에서 객체 3 더하기 정수타입은 정수타입이 된다. 요약 해석에서의 요약 의미를 구하는 것과 비슷한 방법으로 타입과 타입의 연산도 정의가 된다. 또, 객체 3의 타입은 객체 3 자신이고, 정수 타입의 부분타입이 된다. 정해진 타입이 있는 것이 아니라 서로간의 부분타입 관계만 존재한다.



정말 특이한 점은 이 이론적인 논문을 연구소나 대학의 연구자가 아닌 무기만드는 회사(MZA Associates Corporation)의 직원이 썼다는 점이다. 질문 시간에 Philip Wadler께서 이 논문을 혼자서 생각한거냐고 질문했더니 저자가 박사과정일때 지도교수가 이 아이디어를 싫어해서 혼자 연구한 것이라고 답했는데 알고 보니 그 지도교수가 질문하는 Wadler였다. Wadler는 Benjamin C. Piercer의 책을 썼더니 혼자서 연구한 것이라고 자신은 지도한 것이 없다고 하면서 지금은 발표자를 너무나 자랑스럽게 생각한다고 말해주었다. 조사해보니 저 저자는 박사과정 때 쓴 논문들 모두에 Wadler의 이름이 빠져있다. 혼자서도 좋은 연구라 믿고 끝까지 해내어서 POPL에서 지도교수의 인정을 받다니 무척 힘들었텐데 대단하다는 생각을 했다.

Continuity Analysis of Programs - Swarat Chaudhuri, Sumit Gulwani, and Roberto Lubliner

논문 제출하기(submit)의 괴물 Sumit Gulwani는 POPL에 두 편의 논문을 냈는데 이 논문이 그 중에 하나이다. 프로그램의 입력값이 미세하게 변했을때 출력값도 미세하게 변하는지를 검증하는 방법을 제시했다. 같은 연구실의 기흥이가 로봇 소프트웨어에서 입력값에 따른 출력값의 변화를 연구하려고 했기 때문에 더욱 흥미를 갖게 되었다. 논문에서 연구의 동기라고 제시한 GPS나 센서 시스템은 그다지 상관이 있는 것 같지 않고, 함수의 연속성(continuity)이 프로그램에는 어떻게 대응이 될지 고민하다가 연구를 시작한 것 같다는 느낌을 받았다. 루프가 없는 프로그램에 대해 검증하는 방법은 직관적으로 쉽게 잘 써서 읽을 때 재밌었다. 핵심은 프로그램이 어떤 입력 변수와 출력 변수에 대해서 연속하는지를 검사하는 것과 분기문일때 연속하려면 어떤 성질을 만족해야하는지 체크하는 것이다. 분기문이 있을 경우에는 조건문의 값이 바뀌는 그 경계의 순간에 참과 거짓 두 가지에서 실행결과의 출력 변수의 값이 같아야 한다는 조건을 가지고 연속성을 판단한다. 루프가 있을때는 연속성이 부분적으로 깨지더라도 큰 틀에서 보면 같아지는 구간을 정의하고, 그 구간 단위로 연속성이 유지되는 것을 가지고 연속성을 검증해간다. 발표 마지막에 이들이 앞으로 하고 싶은 연구들을 제시했는데 잘 될까라는 의문은 들었지만 멋있다고 생각했다. 프로그램의 미분이 무엇일까? 프로그램의 푸리에 변환은? 프로그램의 적분 등등... 함수의 연속성 말고도 프로그램에 수학의 개념을 적용시키는 연구를 할 모양이다. 성공한다면 읽어보고 싶은 연구 주제들이다. 프로그램의 미분은 기흥이가 하고 싶은 요동 분석(perturbation analysis)과 연관지을 수 있지 않을까? 프로그램의 입력값에 따른 출력값의 변화량을 측정하고 싶은 거니까...



From Program Verification to Program Synthesis - Saurabh Srivastava, Sumit Gulwani, and Jeffrey S. Foster

Sumit Gulwani의 또 다른 논문이다. 프로그램 검증을 이용해서 특정 성질을 만족하는 프로그램을 합성하는 연구이다. 물론 완전 자동으로는 할 수 없고 입력 상태와 출력 상태, 프로그램의 뼈대와 쓸 수 있는 연산(operation)들, 사용하는 변수들의 최대 개수등을 알려줘야만 합성이 가능하다. 크기가 큰 프로그램을 합성하는 것은 아니지만 여러가지 다양한 프로그램들을 합성할 수 있음을 실험을 통해 보여줬다. Gulwani는 이전에 거푸집(template)을 이용하여 프로그램의 불변성(invariant)을 유추하는 연구를 했다. 그 논문에서 프로그램 합성을 앞으로의 연구로 제시했는데 실제로 해냈다는 사실에 역시 대단한 사람이라는 인상을 받았다. Gulwani와 직접 이야기할 기회가 있었는데 우리의 VMCAI 논문이 좋았다고 이야기를 해주며 프로그램 합성에 대한 연구를 우리에게도 권했다. 그리고, 다른 방법으로 하는 프로그램 합성은 잘 되지 않으니 프로그램 분석을 이용해서 합성을 해야한다고 이야기 했다. 그래서, 왕교수님께 이 논문과 함께 Gulwani와의 대화 내용을 알려드렸더니 모델 체킹(model checking) 분야에서 프로그램 합성을 이미 성공적으로 했는데 잘 모르고 하는 이야기라는 답을 받았다. 관련이 있는 논리 프로그램이나 모델체킹 쪽의 논문들을 인용하지 않은 것으로 봐서 많이 알아보고 연구한 것 같지는 않다는 이야기를 들었다. 또, 자동으로 프로그램을 만드는 문제는 프로그램을 검증하는 문제보다도 더 어려운 문제라 잘 안 될 것 같다는 생각하셨고, 나도 동의하여 일단은 접어 두기로 했다.

Low-Level Liquid Types - Patrick Rondon, Ranjit Jhala, and Ming Kawaguchi

이 논문은 내용보다 다른 의미에서 인상이 깊다. 내가 2008년도 PLDI에 참석했을때 동일한 저자들이 Liquid Types라는 논문을 발표했었다. 일년 후 그 타입 시스템을 이용해서 자료 구조를 검증하는 논문을 PLDI에 다시 냈고, 이번에 POPL에는 하위 레벨의 프로그램을 검증하기 위해 타입 시스템을 고쳤다. 최근에 보니 올해 CAV에도 이 타입 시스템을 이용한 Dsolve: Verification Via Liquid Type Inference라는 논문이 채택되었다. 이렇게까지 한 우물을 깊이 파

서 좋은 학회들에만 논문을 내는 모습에 인상을 받았다. 앞으로 박사과정을 할 후배들도 깊이 있게 끝까지 연구하는 자세를 가졌으면 좋겠다.

처음에 나온 Liquid type은 정교한 의존 타입(dependent type)이라는 생각뿐이 없었는데, 이 타입시스템은 서술 추상화(predicate abstraction)와 자동 결정기(SMT solver)를 이용하여 경로를 고려하는 분석을 추가해서 정교한 타입(값의 범위)을 유추하는 것이 가능하다. 전반적으로는 프로그램 분석을 타입 시스템을 정교하게 만들어 감으로써 해내고, 잘 포장했다는 느낌을 받았다. 사실 요약해석 입장에서 보면 타입 시스템이나 프로그램 분석기나 다 같은 것인데 말이다.

POPL accepts more papers

POPL이 열리기 전부터 타입 메일링 리스트에서는 한 가지 주제가 핫이슈로 떠올랐다. POPL에서 논문을 더 많이 받아들여야 하냐는 것이다. 반대의 입장에 선 사람들은 채택률(acceptance ratio)이 올라가면 이 분야를 모르는 사람들의 관점에서 보면 학회의 명성에 누가 갈 수 있다고 말한다. 현재는 20% 이내 정도인데 이를 30%정도까지 올리게 되면 다른 분야의 최고 학회들과 비교했을때 권위에 흠이 간다는 주장이다. 그리고, 더 많은 논문을 받아들이면 발표하는 기간도 더 늘려야하는 문제도 있다고 말한다.

그러나, 대부분의 사람들은 그럼에도 불구하고 더 많은 논문들을 받아들여야 한다고 생각한다. 나도 같은 입장이다. 특히 올해의 POPL은 좋은 논문들이 대거 탈락했다고 한다. POPL이 명실공히 최고의 학회이기에 아무리 좋은 리뷰를 많이 받아도 단 한사람이라도 높은 점수를 주지 않으면 그 논문이 될 가능성이 굉장히 희박하다. 그래서, 이전에 자신이 했던 연구로부터 약간의 변화가 있는 안전한 논문들이 많이 채택되는 경향이 있다. 이는 다양한 혹은 획기적인 (그래서 사람에 따라 호불호가 갈리는) 연구가 발전하는데 방해가 된다. 그 분야를 연구하는 사람들이 지속적으로 연구를 할 수 있도록 장려하는 방향으로 학회가 발전을 해야하는데 내부적으로 소모적인 경쟁을 하고 있다고 생각한다. Martin Rinard 교수님께서 메일링 리스트에 썼듯이 만약 우리 분야의 유망한 박사과정 학생이 좋은 논문을 쓰고도 POPL에서 내부 경쟁에 의해 좌절을 맛보게 되면 다른 분야로 떠나는 일도 생길 수 있다. 교수님은 이 와중에도 자신이 박사과정 중에 쓴 논문 (POPL 두 개 포함)들이 모두 채택되어 이 분야의 연구를 계속할 수 있었다는 일화 (자기자랑)를 썼다. 정말 자부심이 대단한 분이다.

학회장에서 이런 저런 이야기가 나왔지만 결국 더 많은 논문을 채택해야 한다는 생각에는 대부분 공감을 하였다. 그 방법이나 정도에 대해서는 거수를 통해 서로의 의견을 조율하였다. POPL같이 저명한 학회에 미래에 대해서 고민하는 자리에 나도 조금이나마 영향을 주었다는 생각에 뿌듯했었다.

Trip to Toledo

학회 마지막 날에 여행을 할 시간이 주어졌다. 스페인의 옛 수도였던 톨레도(Toledo)를 가기로 했다. 날씨가 무척 맑아 파아란 하늘을 볼 수 있었다. 열차를 타고 1시간 정도를 달려가면 고풍스러운 옛 도시에 도착할 수 있다. 마드리드는 전쟁 후에 도시를 싹 다 밀어버리고, 다시 지어서 길들이 짝짝 뻗어 있지만 톨레도는 길들이 구불구불하



고 좁아서 포근한 느낌을 받았다. 겹겹이 쌓인 세월의 흔적과 구석구석 묻어 있는 사람들의 추억을 맡을 수 있었다. 마치 어릴적 뛰놀던 동네의 골목에서 동무들이 앉곤 했던 그루터기에 다시 앉아 보는...

톨레도의 대성당을 본 순간 그 규모에 압도 되었다. 자세히 보니 그 세밀함에 또 한번 감탄하게 된다. 조각 하나 하나 주의 깊게 보지 않으면 그냥 지나칠 부분들까지 세세하게 꾸며 놓았다. 아, 진정 이 엄청난 건물을 중세시대에 지었던 말인가? 내부로 들어가니 더욱 입이 벌어진다. 그 세세한 장식을 누가 다 했단 말인가? 절대 권력을 가진 집단이 얼마나 엄청난 일을 할 수 있는지에 다시 한번 놀라게 되었다. 천장을 장식하는 조각들과 자연스럽게 이어지는 벽화들, 예배당 벽면에 장식된 수천개의 조각상들... 한번에 들어오는 정보의 양이 엄청나서 잠시 보는 것만으로는 다 이해하기 어려울 정도의 세밀함에 감탄하고 있는데 학주가 이야기 해줬다. 이탈리아에 있는 토리노 성당은 이것보다 세배는 규모가 크고, 세밀함도 세배라고... 정보의 양이 아홉 배이다. 진정 놀라운 일이다. 이집트의 피라미드를 직접 본 것은 아니지만 그것도 사람이 했다고 이제는 믿을 수 있다.

